

# **UnifiedPOS Retail Peripheral Architecture**

**Version 1.16.1 RCSD**

**International Standard**

**For Implementation of Point Of Service Peripherals**

**OMG Document Number: dtc/22-03-22**

**Original submission date: February 18<sup>th</sup>, 2019**

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

**This proposal adds to and extends the UPOS 1.16 standard.**

## 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.

## 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™, IIOP™, 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](http://www.omg.org/report_issue).)

**Document Submitter**

VINX Corp.

**Document Publishing Supporters**

OPOS-J

SorimachiGiken 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.

TABLE OF CONTENTS

TABLE OF CONTENTS.....	5
<b>PREFACE .....</b>	<b>15</b>
UPDATED ITEMS IN RELEASE 1.16.1 .....	17
<b>C H A P T E R 2 1 .....</b>	<b>37</b>
<b>LIGHTS .....</b>	<b>37</b>
<b>SUMMARY .....</b>	<b>37</b>
<b>GENERAL INFORMATION .....</b>	<b>40</b>
CAPABILITIES .....	40
DEVICE SHARING .....	40
LIGHTS CLASS DIAGRAM .....	41
LIGHTS SEQUENCE DIAGRAM.....	42
<b>PROPERTIES(UML ATTRIBUTES) .....</b>	<b>44</b>
CAPALARM PROPERTY.....	44
CAPBLINK PROPERTY.....	44
CAPCOLOR PROPERTY.....	44
CAPPATTERN PROPERTY <i>ADDED IN RELEASE 1.16</i> .....	45
MAXLIGHTS PROPERTY .....	45
<b>METHODS (UML OPERATIONS) .....</b>	<b>46</b>
SWITCHOFF METHOD .....	46
SWITCHOFFPATTERN METHOD .....	46
SWITCHON METHOD <i>UPDATED IN RELEASE1.12</i> .....	47
SWITCHONMULTIPLE METHOD <i>ADDED IN RELEASE 1.16</i> .....	48
SWITCHONPATTERN METHOD <i>ADDED IN RELEASE 1.16</i> .....	49
<b>EVENTS (UML INTERFACES).....</b>	<b>50</b>
DIRECTIOEVENT.....	50
STATUSUPDATEEVENT .....	50
<b>C H A P T E R 2 9 .....</b>	<b>51</b>
<b>POS POWER.....</b>	<b>51</b>
<b>SUMMARY .....</b>	<b>51</b>
<b>GENERAL INFORMATION .....</b>	<b>55</b>
CAPABILITIES .....	55
DEVICE SHARING .....	55
<b>MODEL .....</b>	<b>56</b>
POSPower SEQUENCE DIAGRAM .....	58
POSPower STANDBY SEQUENCE DIAGRAM .....	59
POSPower STATE DIAGRAM.....	60
POSPower POWERSTATE DIAGRAM - PART 1 .....	61
POSPower POWERSTATE DIAGRAM - PART 2 .....	62
POSPower POWERSTATE DIAGRAM - PART 3 .....	63
POSPower STATE CHART DIAGRAM FOR FAN AND TEMPERATURE .....	64
POSPower BATTERY STATE DIAGRAM.....	65
POSPower POWER TRANSITIONS STATE DIAGRAM.....	66
<b>PROPERTIES (UML ATTRIBUTES) .....</b>	<b>67</b>
BATTERYCAPACITYREMAINING PROPERTY .....	67

<b>UPOS Ver1.16 RCSD Specification</b>	
BATTERYCAPACITYREMAININGINSECONDS PROPERTY	<i>ADDED IN RELEASE 1.16</i> ..... 67
BATTERYCRITICALLYLOWTHRESHOLD PROPERTY	..... 67
BATTERYCRITICALLYLOWTHRESHOLDINSECONDS PROPERTY	<i>ADDED IN RELEASE 1.16</i> ..... 68
BATTERYLOWTHRESHOLD PROPERTY	..... 68
BATTERYLOWTHRESHOLDINSECONDS PROPERTY	<i>ADDED IN RELEASE 1.16</i> ..... 69
CAPBATTERYCAPACITYREMAINING PROPERTY	..... 70
CAPBATTERYCAPACITYREMAININGINSECONDS PROPERTY	<i>ADDED IN RELEASE 1.16</i> 70
CAPCHARGE TIME PROPERTY	<i>ADDED IN RELEASE 1.16</i> ..... 70
CAPFANALARM PROPERTY	..... 70
CAPHEATALARM PROPERTY	..... 71
CAPQUICKCHARGE PROPERTY	..... 71
CAPRESTARTPOS PROPERTY	..... 71
CAPSHUTDOWNPOS PROPERTY	..... 71
CAPSTANDBYPOS PROPERTY	..... 72
CAPSUSPENDPOS PROPERTY	..... 72
CAPUPSCHARGE STATE PROPERTY	..... 72
CAPVARIABLEBATTERYCRITICALLYLOWTHRESHOLD PROPERTY	..... 73
CAPVARIABLEBATTERYCRITICALLYLOWTHRESHOLDINSECONDS PROPERTY	<i>ADDED IN RELEASE 1.16</i> 73
CAPVARIABLEBATTERYLOWTHRESHOLD PROPERTY	..... 73
CAPVARIABLEBATTERYLOWTHRESHOLDINSECONDS PROPERTY	<i>ADDED IN RELEASE 1.16</i> 73
CHARGE TIME PROPERTY	<i>ADDED IN RELEASE 1.16</i> ..... 74
ENFORCEDSHUTDOWNDELAYTIME PROPERTY	..... 74
POWERFAILDELAYTIME PROPERTY	..... 74
POWERSOURCE PROPERTY	..... 75
QUICKCHARGE MODE PROPERTY	..... 75
QUICKCHARGE TIME PROPERTY	..... 75
UPSCHARGE STATE PROPERTY	..... 76
<b>METHODS (UML OPERATIONS)</b>	<b>77</b>
RESTARTPOS METHOD	..... 77
SHUTDOWNPOS METHOD	..... 77
STANDBYPOS METHOD	..... 78
SUSPENDPOS METHOD	..... 78
<b>EVENTS (UML INTERFACES)</b>	<b>79</b>
DIRECTIOEVENT	..... 79
STATUSUPDATEEVENT	..... 79
<b>CHAPTER 39</b>	<b>82</b>
<b>VIDEO CAPTURE</b>	<b>82</b>
SUMMARY	..... 82
<b>GENERAL INFORMATION</b>	<b>87</b>
CAPABILITIES	..... 87
VIDEO CAPTURE CLASS DIAGRAM	..... 88
<b>MODEL</b>	<b>89</b>
MODES	..... 89
DEVICE BEHAVIORS	..... 89
Photo Mode	..... 89
Video Mode	..... 90
DEVICE SHARING	..... 91
<b>PROPERTIES (UML ATTRIBUTES)</b>	<b>92</b>
AUTOEXPOSURE PROPERTY	..... 92

<b>UPOS Ver1.16 RCSD Specification</b>	
<b>AUTOFOCUS PROPERTY</b> .....	92
<b>AUTOGAIN PROPERTY</b> .....	92
<b>AUTOWHITEBALANCE PROPERTY</b> .....	93
<b>BRIGHTNESS PROPERTY</b> .....	93
<b>CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY</b> .....	93
<b>CAPAUTOEXPOSURE PROPERTY</b> .....	93
<b>CAPAUTOFOCUS PROPERTY</b> .....	94
<b>CAPAUTOGAIN PROPERTY</b> .....	94
<b>CAPAUTOWHITEBALANCE PROPERTY</b> .....	94
<b>CAPBRIGHTNESS PROPERTY</b> .....	94
<b>CAPCONTRAST PROPERTY</b> .....	94
<b>CAPEXPOSURE PROPERTY</b> .....	95
<b>CAPGAIN PROPERTY</b> .....	95
<b>CAPHORIZONTALFLIP PROPERTY</b> .....	95
<b>CAPHUE PROPERTY</b> .....	95
<b>CAPPHOTO PROPERTY</b> .....	95
<b>CAPPHOTOCOLORSPACE PROPERTY</b> .....	96
<b>CAPPHOTOFRAMERATE PROPERTY</b> .....	96
<b>CAPPHOTORESOLUTION PROPERTY</b> .....	96
<b>CAPPHOTOTYPE PROPERTY</b> .....	96
<b>CAPSATURATION PROPERTY</b> .....	96
<b>CAPSTORAGE PROPERTY</b> .....	97
<b>CAPVERTICALFLIP PROPERTY</b> .....	97
<b>CAPVIDEO PROPERTY</b> .....	97
<b>CAPVIDEOCOLORSPACE PROPERTY</b> .....	98
<b>CAPVIDEOFRAMERATE PROPERTY</b> .....	98
<b>CAPVIDEORESOLUTION PROPERTY</b> .....	98
<b>CAPVIDEOTYPE PROPERTY</b> .....	98
<b>CONTRAST PROPERTY</b> .....	99
<b>EXPOSURE PROPERTY</b> .....	99
<b>GAIN PROPERTY</b> .....	99
<b>HORIZONTALFLIP PROPERTY</b> .....	100
<b>HUE PROPERTY</b> .....	100
<b>PHOTOCOLORSPACE PROPERTY</b> .....	100
<b>PHOTOCOLORSPACELIST PROPERTY</b> .....	101
<b>PHOTOFRAMERATE PROPERTY</b> .....	101
<b>PHOTOMAXFRAMERATE PROPERTY</b> .....	101
<b>PHOTORESOLUTION PROPERTY</b> .....	102
<b>PHOTORESOLUTIONLIST PROPERTY</b> .....	102
<b>PHOTOTYPE PROPERTY</b> .....	102
<b>PHOTOTYPELIST PROPERTY</b> .....	103
<b>REMAININGRECORDINGTIMEINSEC PROPERTY</b> .....	103
<b>SATURATION PROPERTY</b> .....	103
<b>STORAGE PROPERTY</b> .....	104
<b>VERTICALFLIP PROPERTY</b> .....	104
<b>VIDEOCAPTUREMODE PROPERTY</b> .....	105
<b>VIDEOCOLORSPACE PROPERTY</b> .....	106
<b>VIDEOCOLORSPACELIST PROPERTY</b> .....	106
<b>VIDEOFRAMERATE PROPERTY</b> .....	106
<b>VIDEOMAXFRAMERATE PROPERTY</b> .....	107
<b>VIDEORESOLUTION PROPERTY</b> .....	107
<b>VIDEORESOLUTIONLIST PROPERTY</b> .....	107
<b>VIDEOTYPE PROPERTY</b> .....	107
<b>VIDEOTYPELIST PROPERTY</b> .....	108
<b>NOTE: VIDEO CAPTURE DEVICE PROPERTY VALUE RELATIONSHIP</b> .....	108
<b>METHODS (UML OPERATIONS)</b> .....	109

<b>UPOS Ver1.16 RCSD Specification</b>	
<b>STARTVIDEO METHOD</b> .....	109
<b>STOPVIDEO METHOD</b> .....	110
<b>TAKEPHOTO METHOD</b> .....	111
<b>EVENTS (UML INTERFACES)</b> .....	<b>112</b>
<b>DIRECTIOEVENT</b> .....	112
<b>ERROR EVENT</b> .....	113
<b>STATUSUPDATEEVENT</b> .....	114
<b>C H A P T E R 4 0</b> .....	<b>115</b>
<b>INDIVIDUAL RECOGNITION</b> .....	<b>115</b>
<b>SUMMARY</b> .....	<b>115</b>
<b>GENERAL INFORMATION</b> .....	<b>118</b>
<b>CAPABILITIES</b> .....	118
<b>INDIVIDUAL RECOGNITION CLASS DIAGRAM</b> .....	118
<b>MODEL</b> .....	<b>119</b>
<b>INPUT MODEL</b> .....	119
<b>DEVICE SHARING</b> .....	119
IndividualRecognitionFilter.....	120
IndividualRecognitionFilter Property Example Format.....	121
■ Basic Items.....	121
■ Face Recognition device example.....	122
IndividualRecognition Information Property Example Format.....	124
■ Basic Items.....	124
■ Face Recognition Device Example.....	125
<b>PROPERTIES (UML ATTRIBUTES)</b> .....	<b>126</b>
<b>CAPINDIVIDUALLIST PROPERTY</b> .....	126
<b>INDIVIDUALIDS PROPERTY</b> .....	126
<b>INDIVIDUALRECOGNITIONFILTER PROPERTY</b> .....	126
<b>INDIVIDUALRECOGNITIONINFORMATION PROPERTY</b> .....	127
<b>EVENTS (UML INTERFACES)</b> .....	<b>128</b>
<b>DATAEVENT</b> .....	128
<b>DIRECTIOEVENT</b> .....	128
<b>ERROR EVENT</b> .....	129
<b>STATUSUPDATEEVENT</b> .....	130
<b>C H A P T E R 4 1</b> .....	<b>131</b>
<b>SOUND RECORDER</b> .....	<b>131</b>
<b>SUMMARY</b> .....	<b>131</b>
<b>GENERAL INFORMATION</b> .....	<b>135</b>
<b>CAPABILITIES</b> .....	135
<b>SOUND RECORDER CLASS DIAGRAM</b> .....	135
<b>MODEL</b> .....	<b>136</b>
<b>DEVICE SHARING</b> .....	137
<b>PROPERTIES(UML ATTRIBUTES)</b> .....	<b>138</b>
<b>CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY</b> .....	138
<b>CAPCHANNEL PROPERTY</b> .....	138
<b>CAPSAMPLINGRATE PROPERTY</b> .....	138
<b>CAPSOUNDTYPE PROPERTY</b> .....	138



UPOS Ver1.16 RCSD Specification	
CAPSTORAGE PROPERTY .....	139
CAPRECORDINGLEVEL PROPERTY.....	139
CHANNEL PROPERTY .....	139
CHANNELLIST PROPERTY.....	140
RECORDINGLEVEL PROPERTY .....	140
REMAININGRECORDINGTIMEINSEC PROPERTY.....	140
SAMPLINGRATE PROPERTY .....	140
SAMPLINGRATELIST PROPERTY.....	141
SOUNDATA PROPERTY.....	141
SOUNDTYPE PROPERTY .....	141
SOUNDTYPELIST PROPERTY.....	141
STORAGE PROPERTY .....	142
<b>METHODS(UML OPERATIONS) .....</b>	<b>143</b>
STARTRECORDING METHOD.....	143
STOPRECORDING METHOD.....	144
<b>EVENTS(UML INTERFACES).....</b>	<b>145</b>
DATAEVENT .....	145
DIRECTIOEVENT.....	145
ERROREVENT .....	146
STATUSUPDATEEVENT .....	147
<b>C H A P T E R 4 2 .....</b>	<b>148</b>
<b>VOICE RECOGNITION .....</b>	<b>148</b>
<b>SUMMARY.....</b>	<b>148</b>
<b>GENERAL INFORMATION .....</b>	<b>152</b>
CAPABILITIES .....	152
VOICE RECOGNITION CLASS DIAGRAM.....	152
<b>MODEL .....</b>	<b>153</b>
DEVICE SHARING .....	154
<b>PROPERTIES (UML ATTRIBUTES) .....</b>	<b>155</b>
CAPLANGUAGE PROPERTY .....	155
HEARINGDATAPATTERN PROPERTY .....	155
HEARINGDATAWORD PROPERTY .....	155
HEARINGDATAWORDLIST PROPERTY.....	156
HEARINGRESULT PROPERTY .....	157
HEARINGSTATUS PROPERTY .....	158
LANGUAGELIST PROPERTY .....	158
<b>METHODS (UML OPERATIONS) .....</b>	<b>159</b>
STARTHEARINGFREE METHOD.....	159
STARTHEARINGSENTENCE METHOD .....	160
STARTHEARINGWORD METHOD.....	161
STARTHEARINGYESNO METHOD .....	162
STOPHEARING METHOD.....	162
<b>EVENTS (UML INTERFACES).....</b>	<b>163</b>
DATAEVENT .....	163
DIRECTIOEVENT.....	163
ERROREVENT .....	164
STATUSUPDATEEVENT .....	166
<b>C H A P T E R 4 3 .....</b>	<b>167</b>

<b>UPOS Ver1.16 RCSD Specification</b>	
<b>SOUND PLAYER</b> .....	<b>167</b>
<b>SUMMARY</b> .....	<b>167</b>
<b>GENERAL INFORMATION</b> .....	<b>170</b>
<b>CAPABILITIES</b> .....	170
<b>SOUND PLAYER CLASS DIAGRAM</b> .....	170
<b>MODEL</b> .....	<b>171</b>
<b>DEVICE SHARING</b> .....	172
<b>PROPERTIES(UML ATTRIBUTES)</b> .....	<b>173</b>
<b>CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY</b> .....	173
<b>CAPMULTIPLAY PROPERTY</b> .....	173
<b>CAPSOUNDTYPELIST PROPERTY</b> .....	173
<b>CAPSTORAGE PROPERTY</b> .....	174
<b>CAPVOLUME PROPERTY</b> .....	174
<b>DEVICESOUNDLIST PROPERTY</b> .....	174
<b>OUTPUTIDLIST PROPERTY</b> .....	174
<b>STORAGE PROPERTY</b> .....	175
<b>VOLUME PROPERTY</b> .....	175
<b>METHODS (UML OPERATIONS)</b> .....	<b>176</b>
<b>PLAYSOUND METHOD</b> .....	176
<b>STOPSOUND METHOD</b> .....	176
<b>EVENTS (UML INTERFACES)</b> .....	<b>177</b>
<b>DIRECTIOEVENT</b> .....	177
<b>ERROR EVENT</b> .....	178
<b>OUTPUTCOMPLETEEVENT</b> .....	179
<b>STATUSUPDATEEVENT</b> .....	179
<b>CHAPTER 44</b> .....	<b>180</b>
<b>SPEECH SYNTHESIS</b> .....	<b>180</b>
<b>SUMMARY</b> .....	<b>180</b>
<b>GENERAL INFORMATION</b> .....	<b>184</b>
<b>CAPABILITIES</b> .....	184
<b>SPEECH SYNTHESIS CLASS DIAGRAM</b> .....	184
<b>MODEL</b> .....	<b>185</b>
<b>DEVICE SHARING</b> .....	186
<b>PROPERTIES (UML ATTRIBUTES)</b> .....	<b>187</b>
<b>CAPLANGUAGE PROPERTY</b> .....	187
<b>CAPPITCH PROPERTY</b> .....	187
<b>CAPSPEED PROPERTY</b> .....	187
<b>CAPVOICE PROPERTY</b> .....	187
<b>CAPVOLUME PROPERTY</b> .....	188
<b>LANGUAGE PROPERTY</b> .....	188
<b>LANGUAGELIST PROPERTY</b> .....	188
<b>OUTPUTIDLIST PROPERTY</b> .....	188
<b>PITCH PROPERTY</b> .....	189
<b>SPEED PROPERTY</b> .....	189
<b>VOICE PROPERTY</b> .....	189
<b>VOICELIST PROPERTY</b> .....	190
<b>VOLUME PROPERTY</b> .....	190

<b>UPOS Ver1.16 RCSD Specification</b>	
<b>METHODS (UML OPERATIONS)</b>	<b>191</b>
<b>SPEAK METHOD</b>	191
<b>SPEAKIMMEDIATE METHOD</b>	193
<b>STOPCURRENTSPEAKING METHOD</b>	193
<b>STOPSPEAKING METHOD</b>	194
<b>EVENTS (UML INTERFACES)</b>	<b>195</b>
<b>DIRECTIOEVENT</b>	195
<b>ERROR EVENT</b>	196
<b>OUTPUTCOMPLETEEVENT</b>	197
<b>STATUSUPDATEEVENT</b>	197
<b>C H A P T E R 4 5</b>	<b>198</b>
<b>GESTURE CONTROL</b>	<b>198</b>
<b>SUMMARY</b>	<b>198</b>
<b>GENERAL INFORMATION</b>	<b>202</b>
<b>CAPABILITIES</b>	202
<b>GESTURE CONTROL CLASS DIAGRAM</b>	202
<b>MODEL</b>	<b>203</b>
<b>AUTOMATIC CONTROL</b>	203
<b>POSE / MOTION</b>	204
<b>DEVICE SHARING</b>	204
<b>PROPERTIES (UML ATTRIBUTES)</b>	<b>205</b>
<b>AUTOMODE PROPERTY</b>	205
<b>AUTOMODELIST PROPERTY</b>	205
<b>CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY</b>	205
<b>CAPMOTION PROPERTY</b>	206
<b>CAPMOTIONCREATION PROPERTY</b>	206
<b>CAPPOSE PROPERTY</b>	206
<b>CAPPOSECREATION PROPERTY</b>	206
<b>CAPSTORAGE PROPERTY</b>	207
<b>JOINTLIST PROPERTY</b>	207
<b>MOTIONLIST PROPERTY</b>	208
<b>POSECREATIONMODE PROPERTY</b>	208
<b>POSELIST PROPERTY</b>	208
<b>STORAGE PROPERTY</b>	209
<b>TABLE OF GESTURE CONTROL DEVICE LISTED ITEMS IN PROPERTY</b>	210
<b>METHODS (UML OPERATIONS)</b>	<b>211</b>
<b>CREATEMOTION METHOD</b>	211
<b>CREATEPOSE METHOD</b>	211
<b>GETPOSITION METHOD</b>	212
<b>SETPOSITION METHOD</b>	213
<b>SETSPEED METHOD</b>	214
<b>STARTMOTION METHOD</b>	215
<b>STARTPOSE METHOD</b>	216
<b>STOPCONTROL METHOD</b>	216
<b>EVENTS (UML INTERFACES)</b>	<b>217</b>
<b>DIRECTIOEVENT</b>	217
<b>ERROR EVENT</b>	218
<b>OUTPUTCOMPLETEEVENT</b>	219
<b>STATUSUPDATEEVENT</b>	219

<b>UPOS Ver1.16 RCSD Specification</b>	220
<b>C H A P T E R 4 6</b>	<b>220</b>
<b>DEVICE MONITOR</b>	<b>220</b>
<b>SUMMARY</b>	<b>220</b>
<b>GENERAL INFORMATION</b>	<b>223</b>
<b>CAPABILITIES</b>	223
<b>DEVICE MONITOR CLASS DIAGRAM</b>	223
<b>MODEL</b>	<b>224</b>
<b>DEVICE SHARING</b>	224
<b>PROPERTIES (UML ATTRIBUTES)</b>	<b>225</b>
<b>DEVICEDATA PROPERTY</b>	225
<b>DEVICELIST PROPERTY</b>	226
<b>MONITORINGDEVICELIST PROPERTY</b>	227
<b>METHODS (UML OPERATIONS)</b>	<b>228</b>
<b>ADDMONITORINGDEVICE METHOD</b>	228
<b>CLEARMONITORINGDEVICES METHOD</b>	230
<b>DELETEMONITORINGDEVICE METHOD</b>	230
<b>GETDEVICEVALUE METHOD</b>	230
<b>EVENTS (UML INTERFACES)</b>	<b>231</b>
<b>DATAEVENT</b>	231
<b>DIRECTIOEVENT</b>	231
<b>ERROR EVENT</b>	232
<b>STATUSUPDATEEVENT</b>	233
<b>C H A P T E R 4 7</b>	<b>234</b>
<b>GRAPHIC DISPLAY</b>	<b>234</b>
<b>SUMMARY</b>	<b>234</b>
<b>GENERAL INFORMATION</b>	<b>238</b>
<b>CAPABILITIES</b>	238
<b>GRAPHICS DISPLAY CLASS DIAGRAM</b>	239
<b>MODEL</b>	<b>240</b>
<b>Image Display Mode</b>	240
<b>Video Display Mode</b>	240
<b>Web Display Mode</b>	241
<b>DEVICE SHARING</b>	242
<b>PROPERTIES (UML ATTRIBUTES)</b>	<b>243</b>
<b>BRIGHTNESS PROPERTY</b>	243
<b>CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY</b>	243
<b>CAPBRIGHTNESS PROPERTY</b>	243
<b>CAPIMAGETYPE PROPERTY</b>	243
<b>CAPSTORAGE PROPERTY</b>	244
<b>CAPURLBACK PROPERTY</b>	244
<b>CAPURLFORWARD PROPERTY</b>	244
<b>CAPVIDEOTYPE PROPERTY</b>	244
<b>CAPVOLUME PROPERTY</b>	245
<b>DISPLAYMODE PROPERTY</b>	245
<b>IMAGETYPE PROPERTY</b>	246
<b>IMAGETYPELIST PROPERTY</b>	247

<b>UPOS Ver1.16 RCSD Specification</b>	
<b>LOADSTATUS PROPERTY</b> .....	247
<b>STORAGE PROPERTY</b> .....	247
<b>URL PROPERTY</b> .....	248
<b>VIDEOTYPE PROPERTY</b> .....	249
<b>VIDEOTYPELIST PROPERTY</b> .....	249
<b>VOLUME PROPERTY</b> .....	249
<b>METHODS (UML OPERATIONS)</b> .....	<b>250</b>
<b>CANCELURLLOADING METHOD</b> .....	250
<b>GOURLBACK METHOD</b> .....	250
<b>GOURLFORWARD METHOD</b> .....	250
<b>LOADIMAGE METHOD</b> .....	251
<b>LOADURL METHOD</b> .....	251
<b>PLAYVIDEO METHOD</b> .....	252
<b>STOPVIDEO METHOD</b> .....	252
<b>UPDATEURLPAGE METHOD</b> .....	253
<b>EVENTS (UML INTERFACES)</b> .....	<b>254</b>
<b>DIRECTIOEVENT</b> .....	254
<b>ERROREVENT</b> .....	254
<b>OUTPUTCOMPLETEEVENT</b> .....	256
<b>STATUSUPDATEEVENT</b> .....	256
<b>RELATIONSHIP TO OTHER OMG SPECIFICATION AND ACTIVITIES</b> .....	<b>257</b>
<b>ROBOTICS DOMAIN TASK FORCE</b> .....	257
Activities in Robotics Domain Task Force.....	257
RoIS Specification.....	257
Scope of RoIS specification .....	257
Robot Service Ontology [RoSO] RFP .....	258
<b>INTEROPERABILITY BETWEEN UPOS RCSD AND ROIS</b> .....	<b>259</b>
<b>RELATIONSHIP BETWEEN UPOS RCSD AND ROIS</b> .....	259
<b>DOCUMENT HISTORY</b> .....	<b>261</b>
<b>VERSION HISTORY</b> .....	261
<b>GLOSSARY</b> .....	<b>262</b>



Date: ~~Nov.~~ April 2021<sup>6</sup>

# Unified POS RCSD, v1.16.1

~~FTF Beta 1~~

[Goto Table145](#)

This specification adds to and extends the UPOS 1.1<sup>5</sup>6 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](mailto: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.

## **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, 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

## **UPOS Ver1.16 RCSD Specification**

Needham, MA 02494

USA

Tel: +1-781-444-0404

Fax: +1-781-444-0320

Email: [pubs@omg.org](mailto: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](http://www.omg.org/report_issue.htm).



**UPOS Ver1.16 RCSD Specification**  
**UPOS 1.16.1 RCSD Specification Overview**

**Updated Items in Release 1.16.1**

In the current UPOS 1.16, heavy technical review was done and found some area to be changed. Those changes are pointed out and all of changes are listed in the attached table. Please refer to the changes details in the table.

## UPOS Ver1.16 RCSD Specification

### Table 1. Edited Items List for the dtc/20-12-01

Note: Characters in Blue with double erase lines were eliminated, Characters in Red were added.

No Page	Chapter/ Device Name	Items to be corrected	Applied Resolution	JIRA # Issue #				
<u>1</u> 38	21 Lights	<b>checkHealth</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue1</a>				
<u>2</u> 38	21 Lights	<b>compareFirmwareVersion</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue2</a>				
<u>3</u> 38	21 Lights	<b>resetStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue3</a>				
<u>4</u> 38	21 Lights	<b>retrieveStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue4</a>				
<u>5</u> 38	21 Lights	<b>updateFirmware</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue5</a>				
<u>6</u> 38	21 Lights	<b>updateStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue6</a>				
<u>7</u> 52	29 POS Power	<b>UPSChargeState</b> Property in Summary section.	This property may use after <b>open &amp; enabled</b> . However, mistakenly claim was added. Therefore, <del>claim</del> was eliminated	UPOS 1161-3 <a href="#">Issue7</a>				
<u>8</u> 67	29 POS Power	<b>BatteryCriticallyLowThreshold</b> Property in Property section.	This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below. <b>Some possible values of the exception's ErrorCode property are:</b> <table style="width: 100%; border-collapse: collapse;"><tr><td style="border-bottom: 1px solid red;"><b>Value</b></td><td style="border-bottom: 1px solid red;"><b>Meaning</b></td></tr><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.	UPOS 1161-3 <a href="#">Issue8</a>
<b>Value</b>	<b>Meaning</b>							
E_ILLEGAL	An invalid value was specified. Or it does not support this function.							
<u>9</u> 68	29 POS Power	<b>BatteryCriticallyLowThresholdInSeconds</b> Property in Property section.	This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below. <b>Some possible values of the exception's ErrorCode property are:</b> <table style="width: 100%; border-collapse: collapse;"><tr><td style="border-bottom: 1px solid red;"><b>Value</b></td><td style="border-bottom: 1px solid red;"><b>Meaning</b></td></tr><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.	UPOS 1161-3 <a href="#">Issue9</a>
<b>Value</b>	<b>Meaning</b>							
E_ILLEGAL	An invalid value was specified. Or it does not support this function.							
<u>10</u> 68	29 POS Power	<b>BatteryLowThreshold</b> Property in Property section.	This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below. <b>Some possible values of the exception's ErrorCode property are:</b> <table style="width: 100%; border-collapse: collapse;"><tr><td style="border-bottom: 1px solid red;"><b>Value</b></td><td style="border-bottom: 1px solid red;"><b>Meaning</b></td></tr><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.	UPOS 1161-3 <a href="#">Issue10</a>
<b>Value</b>	<b>Meaning</b>							
E_ILLEGAL	An invalid value was specified. Or it does not support this function.							
<u>11</u> 69	29 POS Power	<b>BatteryLowThresholdInSeconds</b> Property in Property section.	This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below. <b>Some possible values of the exception's ErrorCode property are:</b> <table style="width: 100%; border-collapse: collapse;"><tr><td style="border-bottom: 1px solid red;"><b>Value</b></td><td style="border-bottom: 1px solid red;"><b>Meaning</b></td></tr><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.	UPOS 1161-3 <a href="#">Issue11</a>
<b>Value</b>	<b>Meaning</b>							
E_ILLEGAL	An invalid value was specified. Or it does not support this function.							
<u>12</u>	29	<b>EnforcedShutdownDelayTime</b> Property	This property is able to read and write. In that case there	UPOS 1161-3				

**UPOS Ver1.16 RCSD Specification**

74	POS Power	In Property section.	should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below. <b>Some possible values of the exception's ErrorCode property are:</b> <b>Value                      Meaning</b> <hr/> <b>E_ILLEGAL              An invalid value was specified. Or it does not support this function.</b>	<a href="#">Issue12</a>
<a href="#">13</a> 84	39 Video Capturer	<b>VideoCaptureMode Property</b> in Summary section.	This property's mutability should be <b>read-write</b> . Therefore, this sections <del>read-only</del> description was corrected as <b>read-write</b> .	UPOS 1161-4 <a href="#">Issue13</a>
<a href="#">14</a> 85	39 Video Capturer	<b>checkHealth Method</b> in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue1</a>
<a href="#">15</a> 85	39 Video Capturer	<b>clearInput Method</b> in Summary section	This method behavior was incorrect. Therefore, those description should be corrected as follows. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-3 <a href="#">Issue15</a>
<a href="#">16</a> 85	39 Video Capturer	<b>compareFirmwareVersion Method</b> in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue2</a>
<a href="#">17</a> 85	39 Video Capturer	<b>resetStatistics Method</b> in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue3</a>
<a href="#">18</a> 85	39 Video Capturer	<b>retrieveStatistics Method</b> in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue4</a>
<a href="#">19</a> 85	39 Video Capturer	<b>updateFirmware Method</b> in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue5</a>
<a href="#">20</a> 85	39 Video Capturer	<b>updateStatistics Method</b> in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue6</a>
<a href="#">21</a> 89	39 Video Capturer	VCAP_CST_HARTTOTALS_ONLY had a TYPO (1 <sup>st</sup> one) in Device behaviors Photo Mode	This should be corrected as follows. VCAP_CST_HARD <b>T</b> TOTALS_ONLY	UPOS 1161-4 <a href="#">Issue21</a>
<a href="#">22</a> 89	39 Video Capturer	Model Section Device behaviors of Photo mode	This method is executed synchronously. This is not enough explanation. Therefore, it was changed as "This method is <del>executed</del> <b>performed</b> synchronously <b>as the process of taking photo.</b> " And added those since <b>StatusUpdateEvent</b> use are newly added. <b>"The process of recorded data storing is performed asynchronously. StatusUpdateEvents are invoked for the application when the start and the end of device states are changed."</b> And the description "An attempt to nest the video capture device operations will result in an UPOSException being thrown." is changed as follows. <b>"If you try <del>An attempt</del> to nest the video capture device operations of the device, before the storing is finished, <del>will result in</del> an UPOSException <del>will being</del> thrown."</b>	UPOS 1161-10 <a href="#">Issue22</a>
<a href="#">23</a> 90	39 Video Capturer	Model Section Device behaviors of Photo mode ER_CONTINUEINPUT was incorrect and it was changed to ER_RETRY	<b>Error Event</b> behavior was incorrect and they were changed as follows. "If <b>ErrorEvent</b> response is ER_CONTINUEINPUT, the process of input processing continues. was changed as follows." "If <b>ErrorEvent</b> response is ER_ <del>RETRY</del> <b>CONTINUEINPUT</b> , the process of <b>recorded data storing was retried</b> <del>input processing continues</del> . However, as long as the cause of the error is not resolved, the <b>ErrorEvent</b> will occur again immediately.	UPOS 1161-6 <a href="#">Issue23</a>
<a href="#">24</a>	39	Model Section Device behaviors of Photo	ER_CLEAR description was not good enough.	UPOS

**UPOS Ver1.16 RCSD Specification**

90	Video Capture	mode ER_CLEAR description was changed.	“If <b>ErrorEvent</b> is ER_CLEAR, the input processing process is terminated and the taking photo is discarded. Is as changed as follows.” was changed as follows. “If <b>ErrorEvent</b> is ER_CLEAR, <b>all of the device buffered data is cleared</b> <del>the input processing process is terminated</del> and the <b>takePhoto</b> method <del>taking photo</del> is discarded.”	1161-6 <a href="#">Issue24</a>
<a href="#">25</a> 90	39 Video Capture	Model Section Device behaviors of Video mode	Added the <b>startVideo</b> method behavior as follows, since current description was not good enough. “ <b>This method is executed synchronously. During video image capturing, recorded data storing is processed asynchronously and when the start and the end device state is changed, StatusUpdateEvents are delivered to the application. In addition, remaining device recording time is updated in the RemainingRecordingTimeInSec property.</b> ”	UPOS 1161-10 <a href="#">Issue25</a>
<a href="#">26</a> 90	39 Video Capture	VCAP_CST_HARTTOTALS_ONLY had a TYPO (2nd one) in Device behaviors Vide Mode	This should be corrected as follows. VCAP_CST_HARD <del>T</del> TOTALS_ONLY	UPOS 1161-4 <a href="#">Issue21</a>
<a href="#">27</a> 90	39 Video Capture	Model Section Device behaviors of Video mode	“The video capturing ends after the specified time has elapsed or when <b>stopVideo</b> method is called or when <b>clearInput</b> method is called” was changed as follows since it needs additional description. “The video capturing ends after the specified time has elapsed or when <b>stopVideo</b> method is called or when <b>clearInput</b> method is called <b>even startVideo method is called.</b> ”	UPOS 1161-6 <a href="#">Issue27</a>
<a href="#">28</a> 91	39 Video Capture	VCAP_CST_HARTTOTALS_ONLY had a TYPO (3rd and 4th ones) in Device behavior Video Mode.	This should be corrected as follows. VCAP_CST_HARD <del>T</del> TOTALS_ONLY	UPOS 1161-4 <a href="#">Issue21</a>
<a href="#">29</a> 91	39 Video Capture	Model Section Device behaviors of Video mode ER_CONTINUEINPUT was incorrect and it was changed to ER_RETRY	<b>Error Event</b> behavior was incorrect. “If ErrorEvent response is ER_CONTINUEINPUT, the process of input processing continues. was changed as follows.” was changed as follows. “If <b>ErrorEvent</b> response is ER_ <del>RETRY</del> <b>CONTINUEINPUT</b> , the process of <b>recorded data storing was retried</b> <del>input processing continues</del> . However, as long as the cause of the error is not resolved, the <b>ErrorEvent</b> will occur again immediately.”	UPOS 1161-6 <a href="#">Issue23</a>
<a href="#">30</a> 91	39 Video Capture	Model Section Device behaviors of Video mode ER_CLEAR description was changed.	ER_CLEAR description needs to change.. “If <b>ErrorEvent</b> is ER_CLEAR, the input processing process is terminated and the taking photo is discarded. is changed as follows.” was changed as follows. “If <b>ErrorEvent</b> is ER_CLEAR, <b>all of the device buffered data is cleared</b> and the error state is exited <del>input processing process is terminated</del> and the taking <b>video capturing process</b> is discarded.”	UPOS 1161-6 <a href="#">Issue24</a>
<a href="#">31</a> 91	39 Video Capture	Model Section Device behaviors of Video mode Error Event description. Word of “input” was changed to “the data”.	<b>ErrorEvent</b> description was changed, since description was good enough. “An <b>ErrorEvent</b> event (or events) is enqueued if an error occurs while gathering or processing input.” was changed as follows. “An <b>ErrorEvent</b> event (or events) is enqueued if an error occurs while gathering or processing <b>the data-</b> <del>input.</del> ”	UPOS 1161-6 <a href="#">Issue31</a>
<a href="#">32</a> 91	39 Video Capture	VCAP_CST_HARTTOTAL_ONLY had a TYPO (5th one)	This should be corrected as follows. VCAP_CST_HARD <del>T</del> TOTALS_ONLY	UPOS 1161-4 <a href="#">Issue21</a>
<a href="#">33</a> 91	39 Video Capture	Model Section Device behaviors of Video mode <b>clearInput</b> description. Word of “input” was	Regarding <b>clearInput</b> method description was changed as follows. All enqueued <b>data</b> <del>input</del> may be deleted by calling	UPOS 1161-6 <a href="#">Issue33</a>

**UPOS Ver1.16 RCSD Specification**

		changed to “data”	<b>clearInput</b> method. See the <b>clearInput</b> method description for more details.	
<a href="#">34</a> 92	39 Video Capture	<b>AutoExposure</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the access after section <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b> .	UPOS 1161-3 <a href="#">Issue34</a>
<a href="#">35</a> 92	39 Video Capture	<b>AutoFocus</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the access after section <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim- enable</b> .	UPOS 1161-3 <a href="#">Issue34</a>
<a href="#">36</a> 92	39 Video Capture	<b>AutoGain</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the access after section <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim- enable</b> .	UPOS 1161-3 <a href="#">Issue34</a>
<a href="#">37</a> 93	39 Video Capture	<b>AutoWhiteBalance</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the access after section <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim- enable</b> .	UPOS 1161-3 <a href="#">Issue34</a>
<a href="#">38</a> 93	39 Video Capture	<b>Brightness</b> property	This property needs to be used under the <b>open, claim enable</b> , however in the access after section <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim- enable</b> .	UPOS 1161-3 <a href="#">Issue34</a>
<a href="#">39</a> 93	39 Video Capture	<b>CapAssociatedHardTotalsDevice</b> Property	Capability needs to be used under the <b>read-only</b> , however description was read-write. Therefore, it was changed from read-write to <b>read-<del>write</del>only</b> . In addition, regarding the <b>remarks</b> section, description was not good enough to explain this property’s behavior. Therefore, “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 <b>CapStorage</b> is either VCAP_CST_ALL or VCAP_CST_HARDTOTALS_ONLY. If <b>CapStorage</b> is VCAP_CST_HOST_ONLY this property value must be the empty string. “was corrected as follows. “Indicate that the device is able to store the recorded data into the Associated Hard Totals device and <del>H</del> holds <del>the its</del> open name <del>of the associated Hard Totals Device,</del> <del>if the device is able to write to such devices which is the case,</del> if <b>CapStorage</b> is either VCAP_CST_ALL or VCAP_CST_HARDTOTALS_ONLY. If <b>CapStorage</b> is VCAP_CST_HOST_ONLY, <del>the device is not able to store the data into the Associated Hard Totals device and</del> this property value must be the empty string. <b>This property is initialized by the open method.</b>	UPOS 1161-4 <a href="#">Issue39</a>
<a href="#">40</a> 96	39 Video Capture	<b>CapPhotoType</b> Property	This property description’s <b>See Also</b> section missing the reference property name. Added the <b>See Also</b> and described below. <b>See also PhotoType Property</b>	UPOS 1161-4 <a href="#">Issue40</a>
<a href="#">41</a> 97	39 Video Capture	<b>CapStorage</b> Property	This property description’s <b>Remarks</b> did have the incorrect description. Current “This is an enumeration and announces where the device is able to write the recorded sound data file to.” Corrected as follows. “This is an enumeration and announces where the device is able to write the recorded <b>video or photo</b> <del>sound</del> data file to.”	UPOS 1161-4 <a href="#">Issue41</a>
<a href="#">42</a> 99	39 Video Capture	<b>Contrast</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-</b>	UPOS 1161-3 <a href="#">Issue42</a>

**UPOS Ver1.16 RCSD Specification**

			<b>claim-enable.</b>	
<a href="#">43</a> 99	39 Video Capture	<b>Exposure</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec claim and enable were missing. In the access after section changed as <b>open-claim-enable.</b>	UPOS 1161-3 <a href="#">Issue42</a>
<a href="#">44</a> 99	39 Video Capture	<b>Gain</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec claim and enable were missing and newly added. In the access after section changed as <b>open-claim-enable.</b>	UPOS 1161-3 <a href="#">Issue42</a>
<a href="#">45</a> 100	39 Video Capture	<b>HorizontalFlip</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec claim and enable were missing. In the access after section changed as <b>open-claim-enable.</b>	UPOS 1161-3 <a href="#">Issue42</a>
<a href="#">46</a> 100	39 Video Capture	<b>Hue</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec claim and enable were missing. In the access after section changed as <b>open-claim-enable.</b>	UPOS 1161-3 <a href="#">Issue42</a>
<a href="#">47</a> 100	39 Video Capture	<b>PhotoColorSpace</b> Property	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable.</b></p> <p>And this property’s <b>Remarks</b> description was incorrect therefore, it was corrected.</p> <p>Currently “Indicates the photo color space ID of the frame data to be acquired by the Video Capture Device, if <b>CapPhotoColorSpace</b> property is true. Valid values are one of the values listed in the <b>CapPhotoColorSpaceList</b> property. This property is initialized by the <b>open</b> method.”</p> <p>Revised “Indicates the photo color space ID of the frame data to be acquired by the Video Capture Device, if <b>CapPhotoColorSpace</b> property is true <b>and it is used takePhoto method.</b> Valid values are one of the values listed in the <b>CapPhotoColorSpaceList</b> property. <b>This property is referred to when VideoCaptureMode</b> property value is <b>VCAP_VCMODE_PHOTO</b> and <b>CapPhoto</b> is true. This property is initialized by the <b>open</b> method.”</p> <p>In this property, in the <b>See also</b> section several property and method are missing. Therefore, those property and method are added.</p> <p><b>See also</b> <b>PhotoColorSpaceList</b> Property, <b>VideoCaptureMode</b> property, <b>CapPhoto</b> Property, <b>CapPhotoColorSpace</b> Property, <b>takePhoto Method</b></p>	UPOS 1161-3 <a href="#">Issue47</a>
<a href="#">48</a> 101	39 Video Capture	<b>PhotoColorSpaceList</b> Property	<p>In <b>Remarks</b>, there was a word of Photo Color space, this should be Photo color space.</p> <p>In the Color space ID parameter, Description was not good enough. RGB depth description was changed.</p> <p>Currently: ID for identifying the color space of RGB, YUV422, etc. And they are indicating like “RGB;YUV422;.....”</p> <p>Revised: ID for identifying the color space of RGB, YUV422, etc. <b>Then if RGB Depth was 16 bits, YUV422 Depth was 32 bits, And</b> they are indicating like “<b>RGB:16, YUV422:32, .....</b>”</p> <p>In the <b>See Also</b> section, <b>CapPhotoColorSpace</b> Property was missing, therefore it was added.</p>	UPOS 1161-3 <a href="#">Issue48</a>
<a href="#">49</a>	39	<b>PhotoFrameRate</b> Property	This property needs to be used under the <b>open, claim</b>	UPOS 1161-3



**UPOS Ver1.16 RCSD Specification**

101	Video Capture		<p><b>enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p> <p>In <b>Remarks</b>, the device behavior description was not good enough, therefore, description was changed as follows.</p> <p>Currently:          “Indicates the frame rate of frame data to be acquired by the Video Capture Device. Valid values range from 1 to <b>PhotoMaxFrameRate</b> property.”</p> <p>Revised:          “Indicates the frame rate of frame data <del>recorded to be acquired</del> by the Video Capture Device and the photo image capturing and recorded with the <b>takePhoto</b> method. This property is only applied when <b>VCAP_VCMODE_PHOTO</b> is set in <b>VideoCaptureMode</b> property. Valid values range from 1 to <b>PhotoMaxFrameRate</b> property and <b>CapPhoto</b> property is true.”</p> <p>In the <b>See Also</b> section, <b>CapPhoto</b> property, <b>takePhoto</b> method were missing, therefore they were added.</p>	<a href="#">Issue49</a>				
<a href="#">50</a> 102	39 Video Capture	<b>PhotoResolution</b> Property	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p> <p>In <b>Remarks</b>, <b>VideoCaptureMode</b> property related information was missing.</p> <p>Therefore, this description was added.</p> <p><b>This property is only applied when VideoCaptureMode property is set to VCAP_VCMODE_PHOTO and if CapPhoto is true.</b></p> <p>In the <b>See Also</b> section, <b>CapPhoto</b> Property was missing, therefore it was added.</p>	UPOS 1161-3 <a href="#">Issue50</a>				
<a href="#">51</a> 102	39 Video Capture	<b>PhotoResolutionList</b> Property	<p>In the <b>See Also</b> section, <b>CapPhotoResolution</b> Property was missing, therefore it was added.</p>	UPOS 1161-4 <a href="#">Issue51</a>				
<a href="#">52</a> 102	39 Video Capture	<b>PhotoType</b> Property	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p> <p>In <b>Remarks</b>, there is <del>CapPhotoTypeList</del> property, but this is not existing property and it is corrected as <b>PhotoTypeList</b> property.</p> <p>In the <b>See Also</b> section, <b>CapPhoto</b> Property was missing therefore, it was added. And <b>CapPhotoTypeList</b> property was corrected as <del>CapPhotoTypeList</del> property.</p>	UPOS 1161-3 <a href="#">Issue52</a>				
<a href="#">53</a> 103	39 Video Capture	<b>Saturation</b> Property	<p>This property needs to be used under the <b>open, claim enable</b>, however in the description claim and enable were missing. In the access after section changed as <b>open-claim-enable</b>.</p>	UPOS 1161-3 <a href="#">Issue53</a>				
<a href="#">54</a> 104	39 Video Capture	<b>Storage</b> Property	<p>This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.</p> <p><b>Some possible values of the exception’s ErrorCode property are:</b></p> <table border="1" data-bbox="820 1877 1458 1966"> <thead> <tr> <th><b>Value</b></th> <th><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td>E_ILLEGAL</td> <td>An invalid value was specified or recording is ongoing.</td> </tr> </tbody> </table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified or recording is ongoing.	UPOS 1161-3 <a href="#">Issue54</a>
<b>Value</b>	<b>Meaning</b>							
E_ILLEGAL	An invalid value was specified or recording is ongoing.							
<a href="#">55</a> 104	39 Video Capture	<b>VerticalFlip</b> Property	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p>	UPOS 1161-3 <a href="#">Issue55</a>				

UPOS Ver1.16 RCSD Specification

<p><a href="#">56</a> 105</p>	<p>39 Video Capture</p>	<p><b>VideoCaptureMode</b> Property</p>	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p> <p>This property is able to <b>read</b> and <b>write</b>. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.</p> <p><b>Some possible values of the exception's ErrorCode property are:</b></p> <table border="1"> <thead> <tr> <th><b>Value</b></th> <th><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td>E_ILLEGAL</td> <td>An invalid value was specified.</td> </tr> </tbody> </table> <p>In the <b>See also</b> section, there was a description of <b>VideoCaptureMode</b> property, since this is the <b>VideoCaptureMode</b> property section, therefore it is eliminated from the See also section like this.</p> <p><del>VideoCaptureMode property.</del></p>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified.	<p>UPOS 1161-3 <a href="#">Issue56</a></p>
<b>Value</b>	<b>Meaning</b>							
E_ILLEGAL	An invalid value was specified.							
<p><a href="#">57</a> 106</p>	<p>39 Video Capture</p>	<p><b>VideoColorSpace</b> Property</p>	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p> <p>In <b>Remarks</b>, description of “Indicates the video color space ID of the frame data to be used by <b>startVideo</b> method.” was not good enough.</p> <p>Therefore, corrected as follows.</p> <p>“Indicates the video color space ID of the frame data to be <b>acquired by the Video Capture Device, if CapVideoColorSpace property is true and it is</b> used by <b>startVideo</b> method.”</p> <p>In the <b>See also</b> section, <b>CapVideoColorSpace</b> property was missing therefore, it was added.</p>	<p>UPOS 1161-3 <a href="#">Issue57</a></p>				
<p><a href="#">58</a> 106</p>	<p>39 Video Capture</p>	<p><b>VideoColorSpaceList</b> Property</p>	<p>In <b>Remarks</b>, regarding the Color space ID parameter, Description was not good enough. RGB depth description was changed.</p> <p>Currently: ID for identifying the color space of RGB, YUV422, etc. And they are indicating like “RGB;YUV422;.....”</p> <p>Revised: ID for identifying the color space of RGB, YUV422, etc. <b>Then if RGB Depth was 16 bits, YUV422 Depth was 32 bits, And</b> they are indicating like “RGB:16, YUV422:32, .....”</p> <p>In the <b>See Also</b> section, <b>VideoCaptureMode</b> Property was missing, therefore it was added.</p>	<p>UPOS 1161-4 <a href="#">Issue58</a></p>				
<p><a href="#">59</a> 106</p>	<p>39 Video Capture</p>	<p><b>VideoFrameRate</b> Property</p>	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p> <p>In the <b>See also</b> section, <b>CapVideoFrameRate</b> Property was missing, therefore it was added.</p>	<p>UPOS 1161-3 <a href="#">Issue59</a></p>				
<p><a href="#">60</a> 107</p>	<p>39 Video Capture</p>	<p><b>VideoMaxFrameRate</b> Property</p>	<p>In the <b>See also</b> section, <b>VideoCaptureMode</b> Property was missing, therefore it was added.</p>	<p>UPOS 1161-4 <a href="#">Issue60</a></p>				
<p><a href="#">61</a> 107</p>	<p>39 Video Capture</p>	<p><b>VideoResolution</b> Property</p>	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p>	<p>UPOS 1161-3 <a href="#">Issue61</a></p>				
<p><a href="#">62</a> 107</p>	<p>39 Video Capture</p>	<p><b>VideoType</b> Property</p>	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p>	<p>UPOS 1161-3 <a href="#">Issue61</a></p>				
<p><a href="#">63</a> 108</p>	<p>39 Video Capture</p>	<p><b>VideoTypeList</b> Property</p>	<p>In the <b>Remarks</b>, list of image description missing the word of video and it was corrected from “list of image</p>	<p>UPOS 1161-4</p>				



**UPOS Ver1.16 RCSD Specification**

			format” to “list of <b>video</b> image format”. In the <b>See also</b> section, <b>VideoCaptureMode Property</b> was missing and it was added. <del>CapVideoType property</del> is not needed in this section therefore it is eliminated.	<a href="#">Issue63</a>														
<a href="#">64</a> 109	39 Video Capture	<b>startVideo</b> Method	In the <b>Remarks, StatusUpdateEvent</b> related description was not precise therefore, this description was added. “During the video image capturing, the recorded data storing is processed asynchronously and when the start and stop state are changed, <b>StatusUpdateEvents</b> are delivered to the application.” And this description was eliminated. <del>“StatusUpdateEvent will notify the application that there is a change in the power status or a state change during video capturing and recording.”</del> In the Errors section, in the E_ILLEGAL, need to add the description after VCAP_VCMODE_VIDEO. It is “and <b>CapVideo</b> is not true.”	UPOS 1161-10 <a href="#">Issue64</a>														
<a href="#">65</a> 110	39 Video Capture	<b>stopVideo</b> Method	In the <b>Remarks, StatusUpdateEvent</b> related description was changed for the better description. Currently “ <b>StatusUpdateEvent</b> will notify the application that there is a change in the power status or a state change during taking video and recording.” Revised “ <b>StatusUpdateEvent</b> <del>will notify</del> is delivered to notify the application that <del>there is a change in the power status or a state change during taking video and recording</del> the device video capturing and recording were stopped.”	UPOS 1161-10 <a href="#">Issue65</a>														
<a href="#">66</a> 111	39 Video Capture	<b>takePhoto</b> Method	In the <b>Remarks, StatusUpdateEvent</b> related description was not good enough. Therefore, those description was added. “The process of recorded data storing is performed asynchronously. <b>StatusUpdateEvents</b> are delivered to the application when the start and the end states were changed.”	UPOS 1161-10 <a href="#">Issue66</a>														
<a href="#">67</a> 113	39 Video Capture	<b>ErrorEvent</b>	In the <i>ErrorLocus</i> section, EL_OUTPUT should be corrected since OUTPUT cancelling method is <b>clearInput</b> method. Therefore, it is changed from EL_OUTPUT to EL_INPUT. Then description will be as follows. <table border="1"> <thead> <tr> <th><b>Attributes</b></th> <th><b>Type</b></th> <th><b>Description</b></th> </tr> </thead> <tbody> <tr> <td>ErrorLocus</td> <td>int32</td> <td>Location of the error. If EL_<del>IN</del>OUTPUT is specified. An error occurred during asynchronous <del>input. action.</del></td> </tr> </tbody> </table> In the ErrorLocus attribute, EL_OUTPUT related descriptions are changed to EL_INPUT. The <i>ErrorLocus</i> attribute has <del>one of</del> the following values: <table border="1"> <thead> <tr> <th><b>Value</b></th> <th><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td>EL_<del>IN</del>OUTPUT</td> <td>Error occurred while processing asynchronous inoutput.</td> </tr> </tbody> </table> Also, ErrorResponse will be changed as follows. <table border="1"> <thead> <tr> <th><b>Value</b></th> <th><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td>ER_RETRY</td> <td>Retry <del>recorded sending the data storing.</del> The error state is exited. Typically, valid for asynchronous <del>recorded data storing</del> <del>output devices</del> when the locus is EL_<del>IN</del>OUTPUT, in</td> </tr> </tbody> </table>	<b>Attributes</b>	<b>Type</b>	<b>Description</b>	ErrorLocus	int32	Location of the error. If EL_ <del>IN</del> OUTPUT is specified. An error occurred during asynchronous <del>input. action.</del>	<b>Value</b>	<b>Meaning</b>	EL_ <del>IN</del> OUTPUT	Error occurred while processing asynchronous inoutput.	<b>Value</b>	<b>Meaning</b>	ER_RETRY	Retry <del>recorded sending the data storing.</del> The error state is exited. Typically, valid for asynchronous <del>recorded data storing</del> <del>output devices</del> when the locus is EL_ <del>IN</del> OUTPUT, in	UPOS 1161-4 <a href="#">Issue67</a>
<b>Attributes</b>	<b>Type</b>	<b>Description</b>																
ErrorLocus	int32	Location of the error. If EL_ <del>IN</del> OUTPUT is specified. An error occurred during asynchronous <del>input. action.</del>																
<b>Value</b>	<b>Meaning</b>																	
EL_ <del>IN</del> OUTPUT	Error occurred while processing asynchronous inoutput.																	
<b>Value</b>	<b>Meaning</b>																	
ER_RETRY	Retry <del>recorded sending the data storing.</del> The error state is exited. Typically, valid for asynchronous <del>recorded data storing</del> <del>output devices</del> when the locus is EL_ <del>IN</del> OUTPUT, in																	

UPOS Ver1.16 RCSD Specification

			<p>which case the asynchronous <b>recorded data storing output</b> is retried, and the error state is exited. This is the default response <del>when the locus is EL_OUTPUT.</del></p> <p>ER_CLEAR <del>Valid for loci: EL_OUTPUT.</del> Clear all buffered <b>captured input</b> or <b>stored output</b> data <del>(including all asynchronous output).</del> The error state is exited</p> <p>In the <b>See also</b> section, <del>“Device Output Models” on page Intro-25.</del> was eliminated and <b>“Device information Reporting Model “ on Page Intro-30”</b> was added.</p>					
<a href="#">68</a> 116	40 Individual Recognition	<b>clearInput</b> Method in Summary section	This method behavior was incorrect. Therefore, those description should be corrected as follows. <b>void:{raises-exception, use after open, claim }</b>	UPOS 1161-8 <a href="#">Issue15</a>				
<a href="#">69</a> 116	40 Individual Recognition	<b>clearInputProperties</b> Method in Summary section	This method behavior was incorrect. Therefore, those description should be corrected as follows. <b>void:{raises-exception, use after open, claim }</b>	UPOS 1161-8 <a href="#">Issue69</a>				
<a href="#">70</a> 116	40 Individual Recognition	<b>compareFirmwareVersion</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue2</a>				
<a href="#">71</a> 116	40 Individual Recognition	<b>resetStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue3</a>				
<a href="#">72</a> 116	40 Individual Recognition	<b>retrieveStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue4</a>				
<a href="#">73</a> 116	40 Individual Recognition	<b>updateFirmware</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue5</a>				
<a href="#">74</a> 116	40 Individual Recognition	<b>updateStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue6</a>				
<a href="#">75</a> 126	40 Individual Recognition	<b>IndividualRecognitionFilter</b> Property	<p>This property needs to be used under the <b>open, claim enable</b>, however in the spec <b>claim</b> and <b>enable</b> were missing. In the access after section changed as <b>open-claim-enable</b>.</p> <p>This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.</p> <p><b>Some possible values of the exception’s ErrorCode property are:</b></p> <table border="1"> <thead> <tr> <th><b>Value</b></th> <th><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td>E_ILLEGAL</td> <td>An invalid value was specified.</td> </tr> </tbody> </table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified.	UPOS 1161-3 <a href="#">Issue75</a>
<b>Value</b>	<b>Meaning</b>							
E_ILLEGAL	An invalid value was specified.							
<a href="#">76</a> 129	40 Individual Recognition	ErrorEvent	<p>In the <i>ErrorLocus</i> description, EL_OUTPUT related description was existing. However, there is no EL_OUTPUT error since this is input device. Therefore, Added <b>See values below</b>. And added those since there is no need the EL_OUTPUT related description and it is eliminated.</p> <p>It is <del>if EL_OUTPUT is specified. An error occurred during asynchronous action.</del></p> <p>IN the <i>ErrorLocus</i> attribute, EL_OUTPUT related descriptions are eliminated.</p> <table border="1"> <thead> <tr> <th><b>Value</b></th> <th><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td><del>EL_OUTPUT</del></td> <td><del>Error occurred while processing asynchronous output.</del></td> </tr> </tbody> </table>	<b>Value</b>	<b>Meaning</b>	<del>EL_OUTPUT</del>	<del>Error occurred while processing asynchronous output.</del>	UPOS 1161-4 <a href="#">Issue76</a>
<b>Value</b>	<b>Meaning</b>							
<del>EL_OUTPUT</del>	<del>Error occurred while processing asynchronous output.</del>							

UPOS Ver1.16 RCSD Specification

			<p>Also, since this device is input device there is no need the ER_RETRY. Then ER_RETRY description was eliminated as follows.</p> <p><b>Value            Meaning</b> <del>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.</del></p> <p>And ER_CLEAR description was corrected as follows. Loci: was a typo of locus and it was corrected as "Valid for all loci: to all locus:" And there is no EL_OUTPUT since this device is input device. Therefore, <del>and EL_OUTPUT</del> description was eliminated. Then they are described as follows. Then they are described as follows.</p> <p><b>Value            Meaning</b> ER_CLEAR    Valid for all locus<del>loci</del>: EL_INPUT, EL_INPUT_DATA, <del>and EL_OUTPUT</del>. 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.</p> <p>In the <b>See also</b> section, <del>"Device Output Models" on page Intro 25.</del> was eliminated since this is input device.</p>	
<a href="#">77</a> 132	41 Sound Recorder	<b>clearInput</b> Method in Summary section	This method behavior was incorrect. Therefore, those description should be corrected as follows. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue15</a>
<a href="#">78</a> 133	41 Sound Recorder	<b>clearInputProperties</b> Method in Summary section	This method behavior was incorrect. Therefore, those description should be corrected as follows. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue69</a>
<a href="#">79</a> 133	41 Sound Recorder	<b>compareFirmwareVersion</b> Method in Summary section.	This method can be used after open claim enable, however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue2</a>
<a href="#">80</a> 133	41 Sound Recorder	<b>resetStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue3</a>
<a href="#">81</a> 133	41 Sound Recorder	<b>retrieveStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue4</a>
<a href="#">82</a> 133	41 Sound Recorder	<b>updateFirmware</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue5</a>
<a href="#">83</a> 133	41 Sound Recorder	<b>updateStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue6</a>
<a href="#">84</a> 138	41 Sound Recorder	<b>CapAssociatedHardTotalsDevice</b> Property	Capability needs to be used under the <b>read-only</b> , however description was read-write. Therefore, it was changed from read-write to <b>read-<del>write</del>only</b> . In the <b>Remarks</b> initialization description was missing. Therefore, this description was added. <b>This property is initialized by the open method.</b>	UPOS 1161-4 <a href="#">Issue84</a>

**UPOS Ver1.16 RCSD Specification**

<p><a href="#">85</a> 139</p>	<p align="center">41 Sound Recorder</p>	<p><b>Channel</b> Property</p>	<p>This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.</p> <p>Some possible values of the exception's <i>ErrorCode</i> property are:</p> <table border="1"> <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.	<p align="right">UPOS 1161-3 <a href="#">Issue85</a></p>		
<u>Value</u>	<u>Meaning</u>									
E_ILLEGAL	An invalid value was specified.									
<p><a href="#">86</a> 140</p>	<p align="center">41 Sound Recorder</p>	<p><b>SamplingRate</b> Property</p>	<p>This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.</p> <p>Some possible values of the exception's <i>ErrorCode</i> property are:</p> <table border="1"> <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.	<p align="right">UPOS 1161-3 <a href="#">Issue86</a></p>		
<u>Value</u>	<u>Meaning</u>									
E_ILLEGAL	An invalid value was specified.									
<p><a href="#">87</a> 141</p>	<p align="center">41 Sound Recorder</p>	<p><b>SoundType</b> Property</p>	<p>This property can be accessed after open claim enable. And current description missing the <b>enable</b> description and it is corrected as <b>access after open, -claim-enable</b>. This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.</p> <p>Some possible values of the exception's <i>ErrorCode</i> property are:</p> <table border="1"> <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.	<p align="right">UPOS 1161-3 <a href="#">Issue87</a></p>		
<u>Value</u>	<u>Meaning</u>									
E_ILLEGAL	An invalid value was specified.									
<p><a href="#">88</a> 146</p>	<p align="center">41 Sound Recorder</p>	<p><b>ErrorEvent</b></p>	<p>In the <b>See also</b> section, <del>Status, Error code, State-model</del> description was not needed therefore it is eliminated.</p> <p>Then <b>“Device Input Model”</b> on page Intro-22” and <b>“Error Handling”</b> on page Intro-23 was added instead.</p>	<p align="right">UPOS 1161-4 <a href="#">Issue88</a></p>						
<p><a href="#">89</a> 149</p>	<p align="center">42 Voice Recognition</p>	<p><b>checkHealth</b> Method in Summary section</p>	<p>This method can be used after <b>open claim enable</b>, however summary section missing the <b>claim</b> and it was added in the use after section.</p>	<p align="right">UPOS 1161-3 <a href="#">Issue1</a></p>						
<p><a href="#">90</a> 149</p>	<p align="center">42 Voice Recognition</p>	<p><b>clearInput</b> Method in Summary section</p>	<p>This method behavior description missing the several conditions. Therefore, those are newly added.</p> <p><b>void:{raises-exception, use after open, claim}</b></p>	<p align="right">UPOS 1161-8 <a href="#">Issue15</a></p>						
<p><a href="#">91</a> 149</p>	<p align="center">42 Voice Recognition</p>	<p><b>clearInputProperties</b> Method in Summary section</p>	<p>This method behavior description missing the several conditions. Therefore, those are newly added.</p> <p><b>void:{raises-exception, use after open, claim}</b></p>	<p align="right">UPOS 1161-8 <a href="#">Issue69</a></p>						
<p><a href="#">92</a> 164</p>	<p align="center">42 Voice Recognition</p>	<p><b>ErrorEvent</b></p>	<p>In the Attribute section, regarding the ErrorLocus, there is no EL_OUTPUT since this device is input device. Therefore, description of <b>See values below.</b> was added and <del>if EL_OUTPUT is specified. An error occurred during asynchronous action</del> was eliminated. Then corrected as follows</p> <table border="1"> <thead> <tr> <th><u>Attributes</u></th> <th><u>Type</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>ErrorLocus</td> <td>nt32</td> <td>Location of the error. <b>See values below.</b> <del>if EL_OUTPUT is specified. An error occurred during asynchronous action.</del></td> </tr> </tbody> </table> <p>In the <i>ErrorLocus</i> attribute vale, EL_OUTPUT related descriptions are eliminated and corrected as follows.</p>	<u>Attributes</u>	<u>Type</u>	<u>Description</u>	ErrorLocus	nt32	Location of the error. <b>See values below.</b> <del>if EL_OUTPUT is specified. An error occurred during asynchronous action.</del>	<p align="right">UPOS 1161-4 <a href="#">Issue92</a></p>
<u>Attributes</u>	<u>Type</u>	<u>Description</u>								
ErrorLocus	nt32	Location of the error. <b>See values below.</b> <del>if EL_OUTPUT is specified. An error occurred during asynchronous action.</del>								

UPOS Ver1.16 RCSD Specification

			<p><b>Value</b>                      <b>Meaning</b></p> <p><del>EL_OUTPUT</del>      <del>Error occurred while processing asynchronous output.</del></p> <p>In the <i>ErrorResponse</i> attribute value, even this is the input device, ER_RETRY will be used and when locus is EL_INPUT or EL_INPUT_DATA sending data is re-tried. Descriptions are corrected as follows.</p> <p><b>Value</b>                      <b>Meaning</b></p> <p>ER_RETRY              Retry sending the data. The error state is exited. May be valid for some input devices when the locus is EL_INPUT or EL_INPUT_DATA, <del>in</del> which case the input is re-tried, and the error state is exited. <del>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</del></p> <p>And ER_CLEAR description was corrected, since loci: was a typo of locus and it was corrected as “Valid for all loci: to all locus:”</p> <p>And EL_OUTPUT was eliminated, since this is an input device. Therefore, they are corrected as follows.</p> <p><b>Value</b>                      <b>Meaning</b></p> <p>ER_CLEAR              Valid for all <del>locus</del><del>loci</del>: EL_INPUT, EL_INPUT_DATA <del>and EL_OUTPUT</del>. Clear all buffered input <del>or output</del> data <del>(including all asynchronous output).</del></p> <p>In the <b>See also</b> section, <del>“Device Output Models” on page Intro-25.</del> was eliminated since this is an input device.</p>	
<a href="#">93</a> 168	43 Sound Player	<b>checkHealth</b> Method in Summary section	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue1</a>
<a href="#">94</a> 168	43 Sound Player	<b>clearInput</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue15</a>
<a href="#">95</a> 168	43 Sound Player n	<b>clearInputProperties</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue69</a>
<a href="#">96</a> 173	43 Sound Player	<b>CapAssociatedHardTotalsDevice</b> Property	This is the capability property, therefore, can be used as read only, therefore, <del>read-write</del> was corrected as <b>read-only</b> . In the <b>Remarks</b> initialization description was missing. Therefore, this description was added. <b>This property is initialized by the open method.</b>	UPOS 1161-9 <a href="#">Issue84</a>
<a href="#">97</a> 174	43 Sound Player	<b>CapStorage</b> Property	In the <b>See also</b> section, referenced properties were missing, therefore, <b>Storage Property</b> , <b>CapAssociatedHardTotalsDevice Property</b> were added.	UPOS 1161-4 <a href="#">Issue97</a>
<a href="#">98</a> 175	43 Sound Player	<b>Storage</b> Property	This property is able to read and write. In that case there should be the E_ILLEGAL error code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.  <b>Some possible values of the exception’s ErrorCode property are:</b>	UPOS 1161-3 <a href="#">Issue98</a>
			<p><b>Value</b>                      <b>Meaning</b></p> <p>E_ILLEGAL              An invalid value was specified or recording is ongoing.</p>	

UPOS Ver1.16 RCSD Specification

<p><a href="#">99</a> 178</p>	<p>43 Sound Player</p>	<p><b>ErrorEvent</b></p>	<p>In the Attribute section, regarding the ErrorLocus description was incorrect, therefore, “<del>An error occurred during asynchronous action.</del>” was eliminated and added “<b>It is indicating that the error occurred while processing asynchronous output.</b>” instead. They are corrected as follows. <b>Attributes Type Description</b> <hr/><i>ErrorLocus</i> int32 Location of the error. If EL_OUTPUT is specified. <b>It is indicating that the error occurred while processing asynchronous output.</b> <del>An error occurred during asynchronous action.</del></p> <p>In the <i>ErrorLocus</i> description of EL_OUTPUT, the description, “The <i>ErrorLocus</i> attribute has the one of following values:” was changed since only EL_OUTPUT was existing. That is to say, “The <i>ErrorLocus</i> attribute has the <del>one of</del> following values:”</p> <p>In the <i>ErrorResponse</i> attribute values, ER_RETRY and ER_CLEAR were changed since this is the output device and there is no EL_OUTPUT. They are corrected as follows. <b>Value Meaning</b> <hr/>ER_RETRY Retry <del>sending the asynchronous output.</del> <del>data.</del> The error state is exited. <del>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 synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del></p> <p>ER_CLEAR <del>Valid for all loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data (including all asynchronous output). <b>(The effect is the same as when clearOutput method is called.)</b> The error state is exited.</p>	<p>UPOS 1161-4 <a href="#">Issue99</a></p>
<p><a href="#">100</a> 181</p>	<p>44 Speech Synthesis</p>	<p><b>checkHealth</b> Method in Summary section</p>	<p>This method can be used after <b>open claim enable</b>, however summary section missing the <b>claim</b> and it was added in the use after section.</p>	<p>UPOS 1161-3 <a href="#">Issue1</a></p>
<p><a href="#">101</a> 181</p>	<p>44 Speech Synthesis</p>	<p><b>clearInput</b> Method in Summary section</p>	<p>This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b></p>	<p>UPOS 1161-8 <a href="#">Issue15</a></p>
<p><a href="#">102</a> 181</p>	<p>44 Speech Synthesis</p>	<p><b>clearInputProperties</b> Method in Summary section</p>	<p>This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b></p>	<p>UPOS 1161-8 <a href="#">Issue69</a></p>
<p><a href="#">103</a> 182</p>	<p>44 Speech Synthesis</p>	<p><b>clearOutput</b> Method in Summary section</p>	<p>This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b></p>	<p>UPOS 1161-8 <a href="#">Issue103</a></p>
<p><a href="#">104</a> 196</p>	<p>44 Speech Synthesis</p>	<p><b>ErrorEvent</b></p>	<p>In the Attribute section, regarding the ErrorLocus description “<del>An error occurred during asynchronous action.</del>” was eliminated and added “<b>It is indicating that the error occurred while processing asynchronous output.</b>” instead since this is output device. In the ErrorLocus description of EL_OUTPUT, the description, “The <i>ErrorLocus</i> attribute has the one of</p>	<p>UPOS 1161-4 <a href="#">Issue99</a></p>



UPOS Ver1.16 RCSD Specification

			<p>following values:” was changed since only EL_OUTPUT was existing. That is to say, “The ErrorLocus attribute has the <del>one of</del> following values:” In the <i>ErrorResponse</i>, ER_RETRY and ER_CLEAR was changed since this is the output device and there is no EL_OUTPUT.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>ER_RETRY</td> <td> <p>Retry <del>sending the asynchronous output data.</del> The error state is exited. <del>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 synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del></p> </td> </tr> <tr> <td>ER_CLEAR</td> <td> <p><del>Valid for all loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data <del>(including all asynchronous output).</del> The error state is exited. <del>(The effect is the same as when clearOutput method is called.)</del> The error state is existed.</p> </td> </tr> </tbody> </table>	Value	Meaning	ER_RETRY	<p>Retry <del>sending the asynchronous output data.</del> The error state is exited. <del>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 synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del></p>	ER_CLEAR	<p><del>Valid for all loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data <del>(including all asynchronous output).</del> The error state is exited. <del>(The effect is the same as when clearOutput method is called.)</del> The error state is existed.</p>	
Value	Meaning									
ER_RETRY	<p>Retry <del>sending the asynchronous output data.</del> The error state is exited. <del>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 synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del></p>									
ER_CLEAR	<p><del>Valid for all loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data <del>(including all asynchronous output).</del> The error state is exited. <del>(The effect is the same as when clearOutput method is called.)</del> The error state is existed.</p>									
<a href="#">105</a> 199	45 Gesture Control	<b>checkHealth</b> Method in Summary section	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue1</a>						
<a href="#">106</a> 199	45 Gesture Control	<b>clearInput</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue15</a>						
<a href="#">107</a> 199	45 Gesture Control	<b>clearInputProperties</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue69</a>						
<a href="#">108</a> 199	45 Gesture Control	<b>clearOutput</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue103</a>						
<a href="#">109</a> 199	45 Gesture Control	<b>compareFirmwareVersion</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue2</a>						
<a href="#">110</a> 199	45 Gesture Control	<b>resetStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue3</a>						
<a href="#">111</a> 199	45 Gesture Control	<b>retrieveStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue4</a>						
<a href="#">112</a> 199	45 Gesture Control	<b>updateFirmware</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue5</a>						
<a href="#">113</a> 199	45 Gesture Control	<b>updateStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue6</a>						
<a href="#">114</a> 205	45 Gesture Control	<b>CapAssociatedHardTotalsDevice</b> Property	In the Remarks initialization description was missing. Therefore, this description was added. <b>This property is initialized by the open method.</b>	UPOS 1161-4 <a href="#">Issue114</a>						
<a href="#">115</a> 218	45 Gesture Control	<b>ErrorEvent</b>	In the Attribute section, regarding the <i>ErrorLocus</i> description “ <del>An error occurred during asynchronous action.</del> ” was eliminated and added “ <b>It is indicating that the error occurred while processing asynchronous output.</b> ” instead, since this is output device.	UPOS 1161-4 <a href="#">Issue99</a>						

UPOS Ver1.16 RCSD Specification

			<p>In the <i>ErrorLocus</i> description of EL_OUTPUT, the description, “The <i>ErrorLocus</i> attribute has the one of following values:” was changed since only EL_OUTPUT was existing. That is to say, “The <i>ErrorLocus</i> attribute has the <del>one of</del> following values:”</p> <p>In the <i>ErrorResponse</i>, ER_RETRY and ER_CLEAR was changed since this is the output device and there is a EL_OUTPUT only regarding the <i>ErrorLocus</i>.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>ER_RETRY</td> <td>Retry <del>sending the asynchronous output data.</del> The error state is exited. <del>Typically, valid for synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del></td> </tr> <tr> <td>ER_CLEAR</td> <td><del>Valid for all loci: EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data <del>(including all asynchronous output).</del> The error state is exited. <del>(The effect is the same as when clearOutput method is called.)</del> The error state is existed.</td> </tr> </tbody> </table>	Value	Meaning	ER_RETRY	Retry <del>sending the asynchronous output data.</del> The error state is exited. <del>Typically, valid for synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del>	ER_CLEAR	<del>Valid for all loci: EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data <del>(including all asynchronous output).</del> The error state is exited. <del>(The effect is the same as when clearOutput method is called.)</del> The error state is existed.	
Value	Meaning									
ER_RETRY	Retry <del>sending the asynchronous output data.</del> The error state is exited. <del>Typically, valid for synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del>									
ER_CLEAR	<del>Valid for all loci: EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data <del>(including all asynchronous output).</del> The error state is exited. <del>(The effect is the same as when clearOutput method is called.)</del> The error state is existed.									
<a href="#">116</a> 221	46 Device Monitor	<b>checkHealth</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, in use after <b>claim</b> was added.	UPOS 1161-3 <a href="#">Issue1</a>						
<a href="#">117</a> 221	46 Device Monitor	<b>clearInput</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue15</a>						
<a href="#">118</a> 221	46 Device Monitor	<b>clearInputProperties</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue69</a>						
<a href="#">119</a> 221	46 Device Monitor	<b>compareFirmwareVersion</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue2</a>						
<a href="#">120</a> 221	46 Device Monitor	<b>resetStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue3</a>						
<a href="#">121</a> 221	46 Device Monitor	<b>retrieveStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue4</a>						
<a href="#">122</a> 221	46 Device Monitor	<b>updateFirmware</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue5</a>						
<a href="#">123</a> 221	46 Device Monitor	<b>updateStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue6</a>						
<a href="#">124</a> 232	46 Device Monitor	<b>ErrorEvent</b>	<p>In Attribute, regarding the <i>ErrorResponse</i> value, ER_RETRY should be eliminated, since this is the input device. Also, in the ER_CLEAR output device related description was corrected since there are typo of loci and there is not output data because of input device. Therefore, these descriptions were changed as follows.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td><del>ER_RETRY</del></td> <td><del>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</del></td> </tr> </tbody> </table>	Value	Meaning	<del>ER_RETRY</del>	<del>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</del>	UPOS 1161-4 <a href="#">Issue124</a>		
Value	Meaning									
<del>ER_RETRY</del>	<del>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</del>									



**UPOS Ver1.16 RCSD Specification**

			<p align="center"><del>error state is exited.</del></p> <p>ER_CLEAR Valid for all locus: EL_INPUT and EL_INPUT_DATA. Clear all buffered input <del>or output</del> data (<del>including all asynchronous output</del>). The error state is exited. This is the default response when the locus is EL_INPUT.</p>	
<a href="#">125</a> 235	47 Graphic Display	<b>checkHealth</b> Method in Summary section	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue1</a>
<a href="#">126</a> 235	47 Graphic Display	<b>clearInput</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue15</a>
<a href="#">127</a> 235	47 Graphic Display	<b>clearInputProperties</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue69</a>
<a href="#">128</a> 236	47 Graphic Display	<b>clearOutput</b> Method in Summary section	This method behavior description missing the several conditions. Therefore, those are newly added. <b>void:{raises-exception, use after open, claim}</b>	UPOS 1161-8 <a href="#">Issue103</a>
<a href="#">129</a> 236	47 Graphic Display	<b>compareFirmwareVersion</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue2</a>
<a href="#">130</a> 236	47 Graphic Display	<b>resetStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue3</a>
<a href="#">131</a> 236	47 Graphic Display	<b>retrieveStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue4</a>
<a href="#">132</a> 236	47 Graphic Display	<b>updateFirmware</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue5</a>
<a href="#">133</a> 236	47 Graphic Display	<b>updateStatistics</b> Method in Summary section.	This method can be used after <b>open claim enable</b> , however summary section missing the <b>claim</b> and it was added in the use after section.	UPOS 1161-3 <a href="#">Issue6</a>
<a href="#">134</a> 240	47 Graphic Display	Model section, Image Display Mode	The description " <del>The ImageTypeList property lists image files that the device can display.</del> " is duplicated the previous description, therefore, it is eliminated.	UPOS 1161-4 <a href="#">Issue134</a>
<a href="#">135</a> 241	47 Graphic Display	Model section, Video Display Mode	There was a description regarding the <b>CapMultiDisplay</b> property. However, it was not an existing property. Therefore, the description, " <del>To check if the device supports simultaneous playback, check the CapMultiPlay property</del> " was eliminated.	UPOS 1161-4 <a href="#">Issue135</a>
<a href="#">136</a> 241	47 Graphic Display	Model section, Web Display Mode	There was a description of <b>cancelLoading</b> method. This was incorrect and corrected as <b>cancelURLLoading</b> method.	UPOS 1161-4 <a href="#">Issue136</a>
<a href="#">137</a> 243	47 Graphic Display	<b>CapAssociatedHardTotalsDevice</b> Property	Capability can be used <b>read-only</b> , therefore, <b>read-write</b> was corrected as <b>read-only</b> . In the <b>Remarks</b> initialization description was missing. Therefore, this description was added. <b>This property is initialized by the open method.</b>	UPOS 1161-9 <a href="#">Issue84</a>
<a href="#">138</a> 246	47 Graphic Display	<b>DisplayMode</b> Property	In the <b>See also</b> section, referencing method and property was incorrect. Therefore, <b>loadImage Method</b> and <b>playVideo Method</b> were newly added and Video Capture Device Property of <del>CapCaptureColorSpaceList Property</del> and <del>VideoCaptureMode Property</del> were eliminated.	UPOS 1161-4 <a href="#">Issue138</a>
<a href="#">139</a> 247	47 Graphic Display	<b>ImageType</b> Property	This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.	UPOS 1161-3 <a href="#">Issue139</a>

**UPOS Ver1.16 RCSD Specification**

			<p>Some possible values of the exception's <i>ErrorCode</i> property are:</p> <table border="1"> <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									
<a href="#">140</a> 247	47 Graphic Display	<b>LoadStatus</b> Property	This property can be accessed after open, therefore, current description of <del>claim, enable</del> were eliminated.	UPOS 1161-4 <a href="#">Issue140</a>						
<a href="#">141</a> 249	47 Graphic Display	<b>VideoType</b> Property	<p>This property is able to read and write. In that case there should be the E_ILLEGAL error Code. However, that description was missing. Therefore, E_ILLEGAL related description was added as listed below.</p> <p>Some possible values of the exception's <i>ErrorCode</i> property are:</p> <table border="1"> <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	UPOS 1161-3 <a href="#">Issue141</a>		
<u>Value</u>	<u>Meaning</u>									
E_ILLEGAL	An invalid value was specified									
<a href="#">142</a> 252	47 Graphic Display	<b>playVideo</b> Method	<p>There was a description in the <b>Remarks</b> regarding the images. However, this is the <b>playVideo</b> method and explain regarding the video files. Therefore, image related information was corrected as video related information.</p> <p>Then description was corrected as follows. Play the <del>specified</del> video type file content that is specified using <b>VideoType</b> property are loaded in the storage area by the <del>loadImage</del> method. All of loaded the video file <del>images</del> values are listed in the <b>ImageVideoTypeList</b> property, if <b>CapVideoType</b> property is true.</p> <p>Regarding the <b>DisplayMode</b> property values, they were written as GD4SP_DMODE_VIDEO_NORMAL, GD4SP_DMODE_VIDEO_FULL. Since GD4SP was incorrect should be GDSP. Therefore, they were corrected as follows. GDSP_DMODE_VIDEO_NORMAL, GDSP_DMODE_VIDEO_FULL</p>	UPOS 1161-4 <a href="#">Issue142</a>						
<a href="#">143</a> 252	47 Graphic Display	<b>stopVideo</b> Method	<p>In the <b>Remarks</b> section, there was a word of Image file, since this method handling the Video therefore, this was corrected from <del>Image</del> file to <b>Video</b> file.</p> <p>In the <b>See also</b> section, referencing method <del>startVideo</del> method was incorrect and corrected as <b>playVideo</b> method.</p>	UPOS 1161-4 <a href="#">Issue143</a>						
<a href="#">144</a> 254	47 Graphic Display	<b>ErrorEvent</b>	<p>In the Attribute section, regarding the <i>ErrorLocus</i> description "<del>An error occurred during asynchronous action.</del>" was eliminated and added as "It is indicating that the error occurred while processing asynchronous recorded output." instead, since this is output device. Therefore, they were corrected as follows.</p> <table border="1"> <thead> <tr> <th><u>Attributes</u></th> <th><u>Type</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td><i>ErrorLocus</i></td> <td>int32</td> <td>Location of the error. If EL_OUTPUT is specified. It is indicating that the error occurred while processing asynchronous output. <del>An error occurred during asynchronous action.</del></td> </tr> </tbody> </table> <p>In the <i>ErrorLocus</i> description of EL_OUTPUT, the description, "The <i>ErrorLocus</i> attribute has the one of following values:" was changed since only</p>	<u>Attributes</u>	<u>Type</u>	<u>Description</u>	<i>ErrorLocus</i>	int32	Location of the error. If EL_OUTPUT is specified. It is indicating that the error occurred while processing asynchronous output. <del>An error occurred during asynchronous action.</del>	UPOS 1161-4 <a href="#">Issue99</a>
<u>Attributes</u>	<u>Type</u>	<u>Description</u>								
<i>ErrorLocus</i>	int32	Location of the error. If EL_OUTPUT is specified. It is indicating that the error occurred while processing asynchronous output. <del>An error occurred during asynchronous action.</del>								

UPOS Ver1.16 RCSD Specification

			<p>EL_OUTPUT was existing. That is to say, “The <i>ErrorLocus</i> attribute has the <del>one of</del> following values:” In the <i>ErrorResponse</i>, ER_RETRY and ER_CLEAR was changed since this is the output device and there is a EL_OUTPUT only regarding the <i>ErrorLocus</i>.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>ER_RETRY</td> <td>Retry <del>sending</del> the <b>asynchronous output.</b> <del>data.</del> The error state is exited. <del>Typically, valid for synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del></td> </tr> <tr> <td>ER_CLEAR</td> <td><del>Valid for all loci: EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data (including all asynchronous output). The error state is exited. <b>(The effect is the same as when <del>clearOutput</del> method is called.)</b> The error state is existed.</td> </tr> </tbody> </table>	Value	Meaning	ER_RETRY	Retry <del>sending</del> the <b>asynchronous output.</b> <del>data.</del> The error state is exited. <del>Typically, valid for synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del>	ER_CLEAR	<del>Valid for all loci: EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data (including all asynchronous output). The error state is exited. <b>(The effect is the same as when <del>clearOutput</del> method is called.)</b> The error state is existed.	
Value	Meaning									
ER_RETRY	Retry <del>sending</del> the <b>asynchronous output.</b> <del>data.</del> The error state is exited. <del>Typically, valid for synchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited.</del> This is the default response <del>when the locus is EL_OUTPUT.</del>									
ER_CLEAR	<del>Valid for all loci: EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data (including all asynchronous output). The error state is exited. <b>(The effect is the same as when <del>clearOutput</del> method is called.)</b> The error state is existed.									
<a href="#">145</a> 14	Cover Page	Specification Version No. and Submission Date Correction	<p>In the cover page there was an old description. This specification proposing the UPOS 1.16.1 and it editing the UPOS1.16. Also, documentation submission date is Sept. 2021. Such information was added and corrected as follows. <b>Date:</b> <del>Nov.</del> <b>April</b> 2021<b>10</b> Unified POS RCSD, v1.16.1 <del>FTF Beta 1</del> <b>This specification adds to and extends the UPOS 1.156 specification.</b></p>	UPOS 1161-5 <a href="#">Issue145</a>						
<a href="#">146</a> 90	39/ Video Capture	<b>StatusUpdateEvent</b> value in Video Mode	<p>The <b>StatusUpdateEvent</b> value was not VCAP_SUE_END_VIDEO but VCAP_SUE_STOP_VIDEO therefore, the following descriptions were corrected.</p> <p>When the taking video is finished, or the specified time out has been exceeded, a <b>StatusUpdateEvent</b> with status VCAP_SUE_<del>STOP</del><b>END</b>_VIDEO is delivered to notify the application that taking video has been ended.</p>	UPOS 1161-10 <a href="#">Issue146</a>						
<a href="#">147</a> 114	39/ Video Capture	<b>StatusUpdateEvent</b> values in StatusUpdateEvent Section	<p>The <b>StatusUpdateEvent</b> values were incorrect and they were corrected as follows.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>VCAP_SUE_START_VIDEO_<del>RECORDING</del></td> <td>It will be notified when video recording starts.</td> </tr> <tr> <td>VCAP_SUE_STOP_VIDEO_<del>RECORDING</del></td> <td>It will be notified when video recording stops.</td> </tr> </tbody> </table>	Value	Meaning	VCAP_SUE_START_VIDEO_ <del>RECORDING</del>	It will be notified when video recording starts.	VCAP_SUE_STOP_VIDEO_ <del>RECORDING</del>	It will be notified when video recording stops.	UPOS 1161-10 <a href="#">Issue147</a>
Value	Meaning									
VCAP_SUE_START_VIDEO_ <del>RECORDING</del>	It will be notified when video recording starts.									
VCAP_SUE_STOP_VIDEO_ <del>RECORDING</del>	It will be notified when video recording stops.									
<a href="#">148</a> 128	40/ Individual Recognition	<b>DataEvent</b> description in <b>DataEvent</b> section	<p>The <b>DataEvent</b> descriptions were incorrect and they were corrected as follows.</p> <p><b>Remarks</b> Before this event is delivered, the <b>data is copied into corresponding properties</b> <del>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.</del></p> <p><b>See Also</b> <del>VideoCaptureMode Property, “Events” on page Intro-19, directIO method.</del> <b>VideoCaptureMode Property, “Events” on page Intro-19.</b></p>	UPOS 1161-4 <a href="#">Issue148</a>						

## UPOS Ver1.16 RCSD Specification

<a href="#">149</a> 143	41/ Sound Recorder	<b>startRecording</b> Method <i>recordingTime</i> parameter value.	This method's <i>recordingTime</i> value description <del>UPOS=</del> FOREVER(-1) was incorrect. Ando it was changed as FOREVER(-1).	UPOS 1161-4 <a href="#">Issue149</a>
<a href="#">150</a> 168	43/ Sound Player	<b>CapAssociatedHardTotals</b> property in Summary section.	Capability can be used read-only condition. Therefore it is corrected from <del>read-write</del> to <b>read-only</b> .	UPOS 1161-9 <a href="#">Issue150</a>
<a href="#">151</a> 77	29/ POS Power	<b>restartPOS</b> Method	This method use condition description is not consistent with other ones. Therefore, <del>use after open, enable</del> is changed to <b>use after open-enable</b> .	UPOS 1161-7 <a href="#">Issue151</a>
<a href="#">152</a> 77	29/ POS Power	<b>shutdownPOS</b> Method	This method use condition description is not consistent with other ones. Therefore, <del>use after open, enable</del> is changed to <b>use after open-enable</b> .	UPOS 1161-7 <a href="#">Issue151</a>
<a href="#">153</a> 78	29/ POS Power	<b>standbyPOS</b> Method	This method use condition description is not consistent with other ones. Therefore, <del>use after open, enable</del> is changed to <b>use after open-enable</b> .	UPOS 1161-7 <a href="#">Issue151</a>
<a href="#">154</a> 78	29/ POS Power	<b>suspendPOS</b> Method	This method use condition description is not consistent with other ones. Therefore, <del>use after open, enable</del> is changed to <b>use after open-enable</b> .	UPOS 1161-7 <a href="#">Issue151</a>
<a href="#">155</a> 174	43/ Sound Player	<b>OutputIDList</b> Property	This property use condition description is not consistent with other ones. Therefore, <del>use after open, claim-enable</del> is changed to <b>use after open-claim-enable</b> .	UPOS 1161-4 <a href="#">Issue155</a>
<a href="#">156</a> 188	44/ Speech Synthesis	<b>OutputIDList</b> Property	This property can be used <b>read-only</b> condition. Therefore, it is corrected from <del>read-write</del> to <b>read-only</b> .	UPOS 1161-9 <a href="#">Issue155</a>
<a href="#">157</a> 194	44/ Speech Synthesis	<b>stopSpeaking</b> Method	This method use condition description is not consistent with other ones. Therefore, <del>use after open, claim, enable</del> is changed to <b>use after open-claim-enable</b> .	UPOS 1161-7 <a href="#">Issue157</a>

# 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>
<b>AutoDisable:</b>	<i>boolean</i>	{read-write}	1.12	<i>Not supported</i>
<b>CapCompareFirmwareVersion:</b>	<i>boolean</i>	{read-only}	1.12	open
<b>CapPowerReporting:</b>	<i>int32</i>	{read-only}	1.12	open
<b>CapStatisticsReporting:</b>	<i>boolean</i>	{read-only}	1.12	open
<b>CapUpdateFirmware:</b>	<i>boolean</i>	{read-only}	1.12	open
<b>CapUpdateStatistics:</b>	<i>boolean</i>	{read-only}	1.12	open
<b>CheckHealthText:</b>	<i>string</i>	{read-only}	1.12	open
<b>Claimed:</b>	<i>boolean</i>	{read-only}	1.12	open
<b>DataCount:</b>	<i>int32</i>	{read-only}	1.12	<i>Not supported</i>
<b>DataEventEnabled:</b>	<i>boolean</i>	{read-write}	1.12	<i>Not supported</i>
<b>DeviceEnabled:</b>	<i>boolean</i>	{read-write}	1.12	open & claim
<b>FreezeEvents:</b>	<i>boolean</i>	{read-write}	1.12	open
<b>OutputID:</b>	<i>int32</i>	{read-only}	1.12	<i>Not supported</i>
<b>PowerNotify:</b>	<i>int32</i>	{read-write}	1.12	open
<b>PowerState:</b>	<i>int32</i>	{read-only}	1.12	open
<b>State:</b>	<i>int32</i>	{read-only}	1.12	--
<b>DeviceControlDescription:</b>	<i>string</i>	{read-only}	1.12	--
<b>DeviceControlVersion:</b>	<i>int32</i>	{read-only}	1.12	--
<b>DeviceServiceDescription:</b>	<i>string</i>	{read-only}	1.12	open
<b>DeviceServiceVersion:</b>	<i>int32</i>	{read-only}	1.12	open
<b>PhysicalDeviceDescription:</b>	<i>string</i>	{read-only}	1.12	open
<b>PhysicalDeviceName:</b>	<i>string</i>	{read-only}	1.12	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>
<b>CapAlarm:</b>	<i>int32</i>	{read-only}	1.12	open
<b>CapBlink:</b>	<i>boolean</i>	{read-only}	1.12	open
<b>CapColor:</b>	<i>int32</i>	{read-only}	1.12	open
<b>CapPattern:</b>	<i>int32</i>	{read-only}	1.16	open
<b>MaxLights:</b>	<i>int32</i>	{read-only}	1.12	open

### Methods (UML operations)

#### Common

<i>Name</i>	<i>Version</i>	
<b>open (logicalDeviceName: <i>string</i>):</b> void {raises-exception}	1.12	
<b>close ():</b> void {raises-exception, use after open}	1.12	
<b>claim (timeout: <i>int32</i>):</b> void {raises-exception, use after open}	1.12	
<b>release ():</b> void {raises-exception, use after open, claim}	1.12	
<b>checkHealth (level: <i>int32</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.12	<a href="#">Goto Table1</a>
<b>clearInput ():</b> void { }	<i>Not supported</i>	
<b>clearInputProperties ():</b> void { }	<i>Not supported</i>	
<b>clearOutput ():</b> void { }	<i>Not supported</i>	
<b>directIO (command: <i>int32</i>, inout data: <i>int32</i>, inout obj: <i>object</i>):</b> void {raises-exception, use after open}	1.12	
<b>compareFirmwareVersion (firmwareFileName: <i>string</i>, out result: <i>int32</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.12	<a href="#">Goto Table2</a>
<b>resetStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.12	<a href="#">Goto Table3</a>
<b>retrieveStatistics (inout statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.12	<a href="#">Goto Table4</a>
<b>updateFirmware (firmwareFileName: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.12	<a href="#">Goto Table5</a>
<b>updateStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.12	<a href="#">Goto Table6</a>

## UPOS Ver1.16 RCSD Specification

### Specific

#### *Name*

<b>switchOff</b> ( <b>lightNumber:</b> <i>int32</i> ): void {raises-exception, use after open, claim, enable}	1.12
<b>switchOn</b> ( <b>lightNumber:</b> <i>int32</i> , <b>blinkOnCycle:</b> <i>int32</i> , <b>blinkOffCycle:</b> <i>int32</i> , <b>color:</b> <i>int32</i> , <b>alarm:</b> <i>int32</i> ): void {raises-exception, use after open, claim, enable}	1.12
<b>switchOnMultiple</b> ( <b>lightNumbers:</b> string, <b>blinkOnCycle:</b> <i>int32</i> , <b>blinkOffCycle:</b> <i>int32</i> , <b>color:</b> <i>int32</i> , <b>alarm:</b> <i>int32</i> ): void {raises-exception, use after open, claim, enable}	1.16
<b>switchOnPattern</b> ( <b>pattern:</b> <i>int32</i> , <b>alarm:</b> <i>int32</i> ): void {raises-exception, use after open, claim, enable}	1.16
<b>switchOffPattern</b> (): 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>
<b>upos::events::DataEvent</b>		<i>Not supported</i>	
<b>upos::events::DirectIOEvent</b>			1.12
<b>EventNumber:</b>	<i>int32</i>	{read-only}	
<b>Data:</b>	<i>int32</i>	{read-write}	
<b>Obj:</b>	<i>object</i>	{read-write}	
<b>upos::events::ErrorEvent</b>		<i>Not supported</i>	
<b>upos::events::OutputCompleteEvent</b>		<i>Not supported</i>	
<b>upos::events::StatusUpdateEvent</b>			1.12
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	1.16

## 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.



## Lights Class Diagram

*Updated in Release 1.16*

The following diagram shows the relationships between the Lights classes

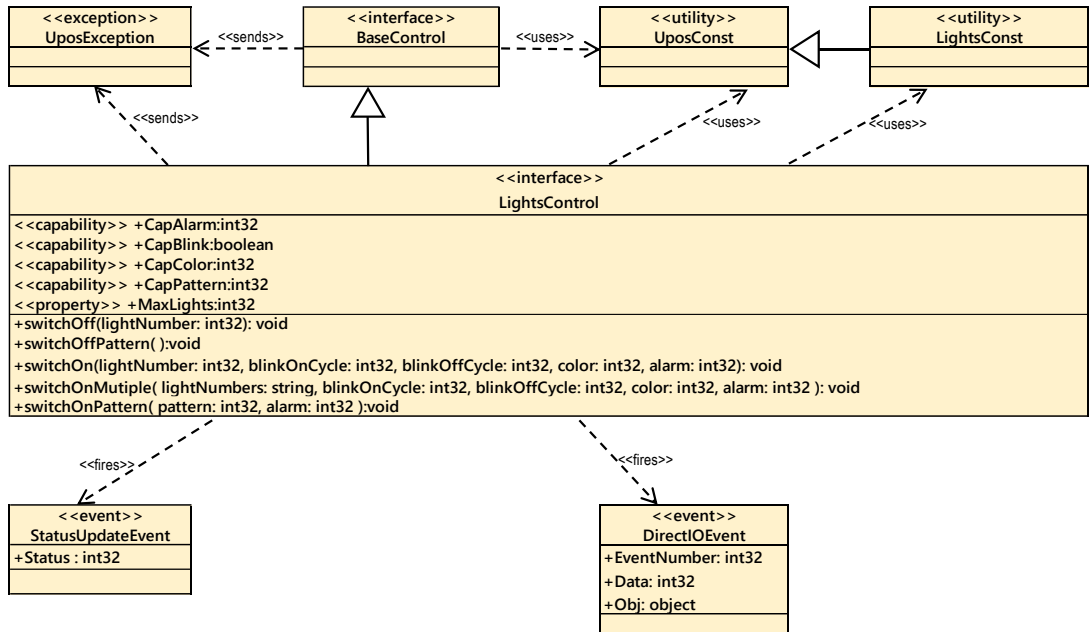


Fig. Chap. 21-1 Lights Class Diagram

## 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, lights4 is CoinDispenser) and that CapBlink is true.

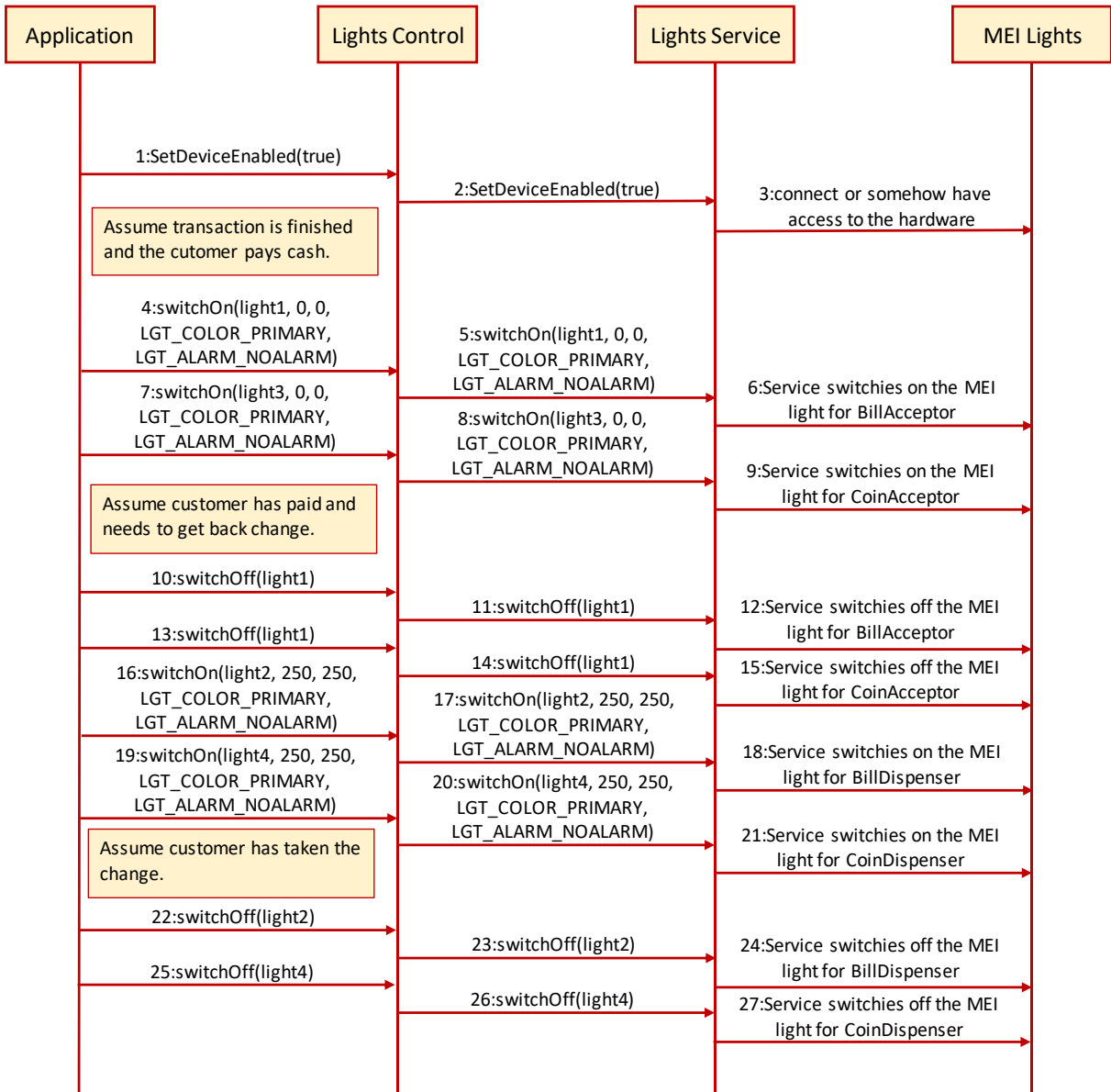


Fig. Chap. 21-2 Lights Sequence Diagram (handling of the media entry indicator lights)

## 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.

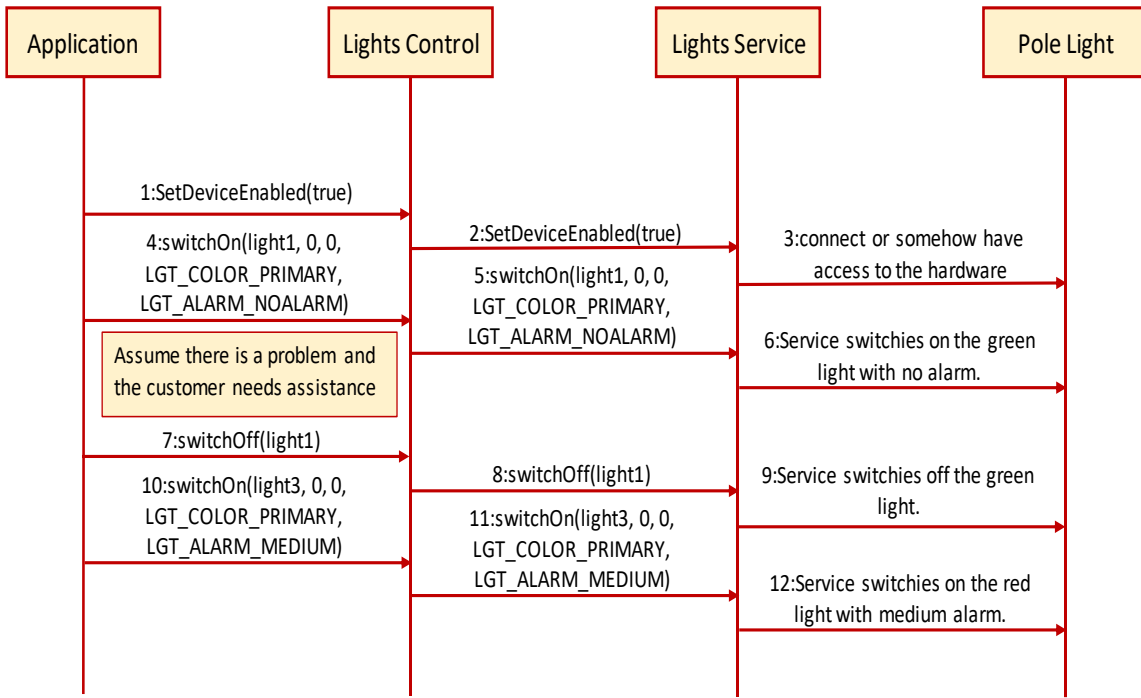


Fig. Chap. 21-3 Lights Sequence Diagram (handling of the pole lights)

# UPOS Ver1.16 RCSD Specification

## Properties(UML attributes)

### CapAlarm Property

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

**Remarks**      This capability indicates if the device supports different alarms.

**CapAlarm** is a logical OR combination of any of the following values:

<b>Value</b>	<b>Meaning</b>
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.

This property is initialized by the **open** method. If the device does not support alarms, it is initialized to LGT\_ALARM\_NOALARM.

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

### CapBlink Property

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

**Remarks**      If true, a blinking capability is supported. It may be either a physical capability of the device or emulated by the 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.

### CapColor Property

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

**Remarks**      This capability indicates if the device supports different colors.

**CapColor** is a logical OR combination of any of the following values:

<b>Value</b>	<b>Meaning</b>
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.

This property is initialized by the **open** method. If the device supports only one color, it is initialized to LGT\_COLOR\_PRIMARY.

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

**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:

<b>Value</b>	<b>Meaning</b>
--------------	----------------

LGT_PATTERN_NOPATTERN	
-----------------------	--

Lighting patterns are not supported.

LGT_PATTERN_CUSTOM	
--------------------	--

1~32 Supports 1<sup>st</sup> to 32<sup>th</sup> Lighting Pattern.

This property is initialized by the **open** method. If the device does not support lighting pattern, it is initialized to LGT\_PATTERN\_NOPATTERN.

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

**See Also**      **switchOnPattern** Method.

**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.

## Methods (UML operations)

### switchOff Method

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

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

**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:

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

**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:

Value	Meaning
E_ILLEGAL	Pattern lighting is not executed.

**See Also**      **switchOnPattern** Method.

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

<b>Parameter</b>	<b>Description</b>
<i>lightNumber</i>	Specifies the light number. Valid light numbers are 1 through <b>MaxLights</b> .
<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>	Specifies the color of the light, must be one of the colors defined by <b>CapColor</b> .
<i>alarm</i>	Specifies the used alarm type, must be one of the alarms defined by <b>CapAlarm</b> .

**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:

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

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

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

<b>Parameter</b>	<b>Description</b>
<i>lightNumbers</i>	Specifies the comma-delimited list of light number. Valid light numbers are 1 through <b>MaxLights</b> .
<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>	Specifies the color of the light, must be one of the colors defined by <b>CapColor</b> .
<i>alarm</i>	Specifies the used alarm type, must be one of the alarms defined by <b>CapAlarm</b> .

**Remarks**      This method does the same as **switchOn** 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:

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

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



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

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

**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:

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

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

## 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 " <b>StatusUpdateEvent</b> " 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>
<b>AutoDisable:</b>	<i>boolean</i>	{read-write}	1.5	<i>Not supported</i>
<b>CapCompareFirmwareVersion:</b>	<i>boolean</i>	{read-only}	1.9	open
<b>CapPowerReporting:</b>	<i>int32</i>	{read-only}	1.3	open
<b>CapStatisticsReporting:</b>	<i>boolean</i>	{read-only}	1.8	open
<b>CapUpdateFirmware:</b>	<i>boolean</i>	{read-only}	1.9	open
<b>CapUpdateStatistics:</b>	<i>boolean</i>	{read-only}	1.8	open
<b>CheckHealthText:</b>	<i>string</i>	{read-only}	1.5	open
<b>Claimed:</b>	<i>boolean</i>	{read-only}	1.5	open
<b>DataCount:</b>	<i>int32</i>	{read-only}	1.5	<i>Not supported</i>
<b>DataEventEnabled:</b>	<i>boolean</i>	{read-write}	1.5	<i>Not supported</i>
<b>DeviceEnabled:</b>	<i>boolean</i>	{read-write}	1.5	open & claim
<b>FreezeEvents:</b>	<i>boolean</i>	{read-write}	1.5	open
<b>OutputID:</b>	<i>int32</i>	{read-only}	1.5	<i>Not supported</i>
<b>PowerNotify:</b>	<i>int32</i>	{read-write}	1.5	open
<b>PowerState:</b>	<i>int32</i>	{read-only}	1.5	open
<b>State:</b>	<i>int32</i>	{read-only}	1.5	--
<b>DeviceControlDescription:</b>	<i>string</i>	{read-only}	1.5	--
<b>DeviceControlVersion:</b>	<i>int32</i>	{read-only}	1.5	--
<b>DeviceServiceDescription:</b>	<i>string</i>	{read-only}	1.5	open
<b>DeviceServiceVersion:</b>	<i>int32</i>	{read-only}	1.5	open
<b>PhysicalDeviceDescription:</b>	<i>string</i>	{read-only}	1.5	open
<b>PhysicalDeviceName:</b>	<i>string</i>	{read-only}	1.5	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>
<b>CapBatteryCapacityRemaining:</b>	<i>boolean</i>	{read-only}	1.9	open
<b>CapBatteryCapacityRemainingInSeconds:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapChargeTime:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapFanAlarm:</b>	<i>boolean</i>	{read-only}	1.5	open
<b>CapHeatAlarm:</b>	<i>boolean</i>	{read-only}	1.5	open
<b>CapQuickCharge:</b>	<i>boolean</i>	{read-only}	1.5	open
<b>CapRestartPOS:</b>	<i>boolean</i>	{read-only}	1.9	open
<b>CapShutdownPOS:</b>	<i>boolean</i>	{read-only}	1.5	open
<b>CapStandbyPOS:</b>	<i>boolean</i>	{read-only}	1.9	open
<b>CapSuspendPOS:</b>	<i>boolean</i>	{read-only}	1.9	open
<b>CapUPSChargeState:</b>	<i>int32</i>	{read-only}	1.5	open
<b>CapVariableBatteryCriticallyLowThreshold:</b>	<i>boolean</i>	{read-only}	1.9	open
<b>CapVariableBatteryCriticallyLowThresholdInSeconds:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVariableBatteryLowThreshold:</b>	<i>boolean</i>	{read-only}	1.9	open
<b>CapVariableBatteryLowThresholdInSeconds:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>BatteryCapacityRemaining:</b>	<i>int32</i>	{read-only}	1.9	open
<b>BatteryCapacityRemainingInSeconds:</b>	<i>int32</i>	{read-only}	1.16	open
<b>BatteryCriticallyLowThreshold:</b>	<i>int32</i>	{read-write}	1.9	open
<b>BatteryCriticallyLowThresholdInSeconds:</b>	<i>int32</i>	{read-write}	1.16	open
<b>BatteryLowThreshold:</b>	<i>int32</i>	{read-write}	1.9	open
<b>BatteryLowThresholdInSeconds:</b>	<i>int32</i>	{read-write}	1.16	open
<b>ChargeTime:</b>	<i>int32</i>	{read-only}	1.16	open
<b>EnforcedShutdownDelayTime:</b>	<i>int32</i>	{read-write}	1.5	open
<b>PowerFailDelayTime:</b>	<i>int32</i>	{read-only}	1.5	open
<b>PowerSource:</b>	<i>int32</i>	{read-only}	1.9	open
<b>QuickChargeMode:</b>	<i>boolean</i>	{read-only}	1.5	open
<b>QuickChargeTime:</b>	<i>int32</i>	{read-only}	1.5	open
<b>UPSChargeState:</b>	<i>int32</i>	{read-only}	1.5	open, <del>claim</del> & enable

[Goto Table7](#)

## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)

#### Common

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

#### Specific

<i>Name</i>	
<b>restartPOS ():</b> void {raises-exception, use after open, enable}	1.9
<b>shutdownPOS ():</b> void {raises-exception, use after open, enable}	1.5
<b>standbyPOS (reason: <i>int32</i>):</b> void {raises-exception, use after open, enable}	1.9
<b>suspendPOS (reason: <i>int32</i>):</b> 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>
<b>upos::events::DataEvent</b>		<i>Not supported</i>	
<b>upos::events::DirectIOEvent</b>			1.5
<b>EventNumber:</b>	<i>int32</i>	{ read-only }	
<b>Data:</b>	<i>int32</i>	{ read-write }	
<b>Obj:</b>	<i>object</i>	{ read-write }	
<b>upos::events::ErrorEvent</b>		<i>Not supported</i>	
<b>upos::events::OutputCompleteEvent</b>		<i>Not supported</i>	
<b>upos::events::StatusUpdateEvent</b>			1.5
<b>Status:</b>	<i>int32</i>	{ read-only }	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	1.16

## 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.

## 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.

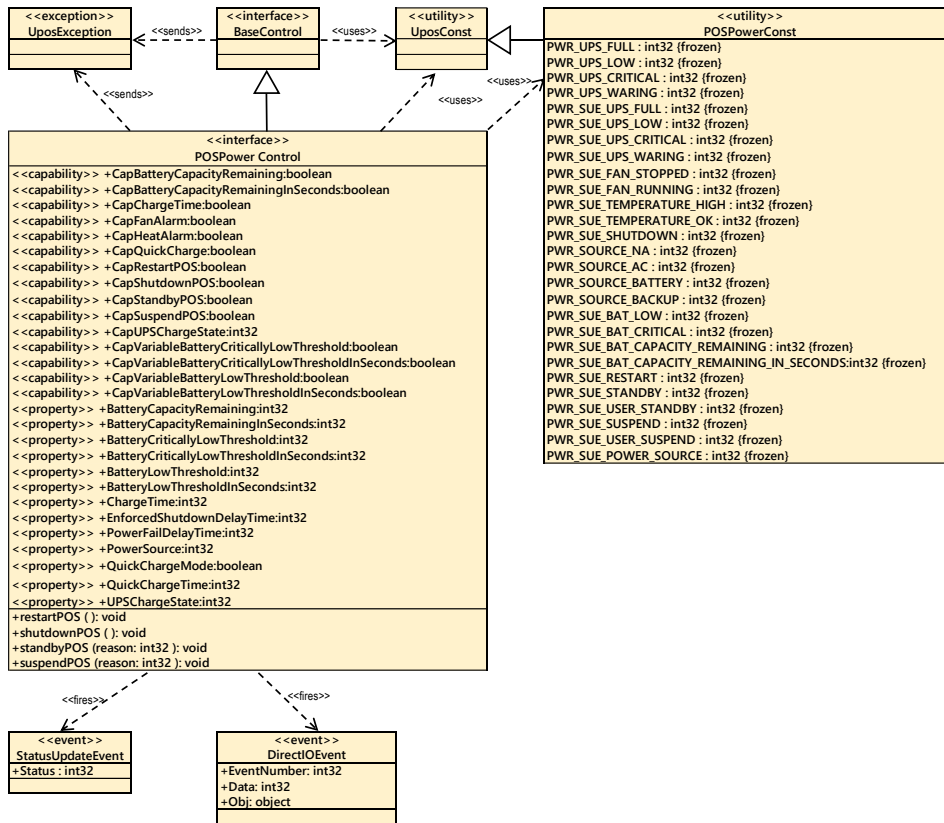


Fig. Chap.29-1 POSPower Class Diagram

# 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.

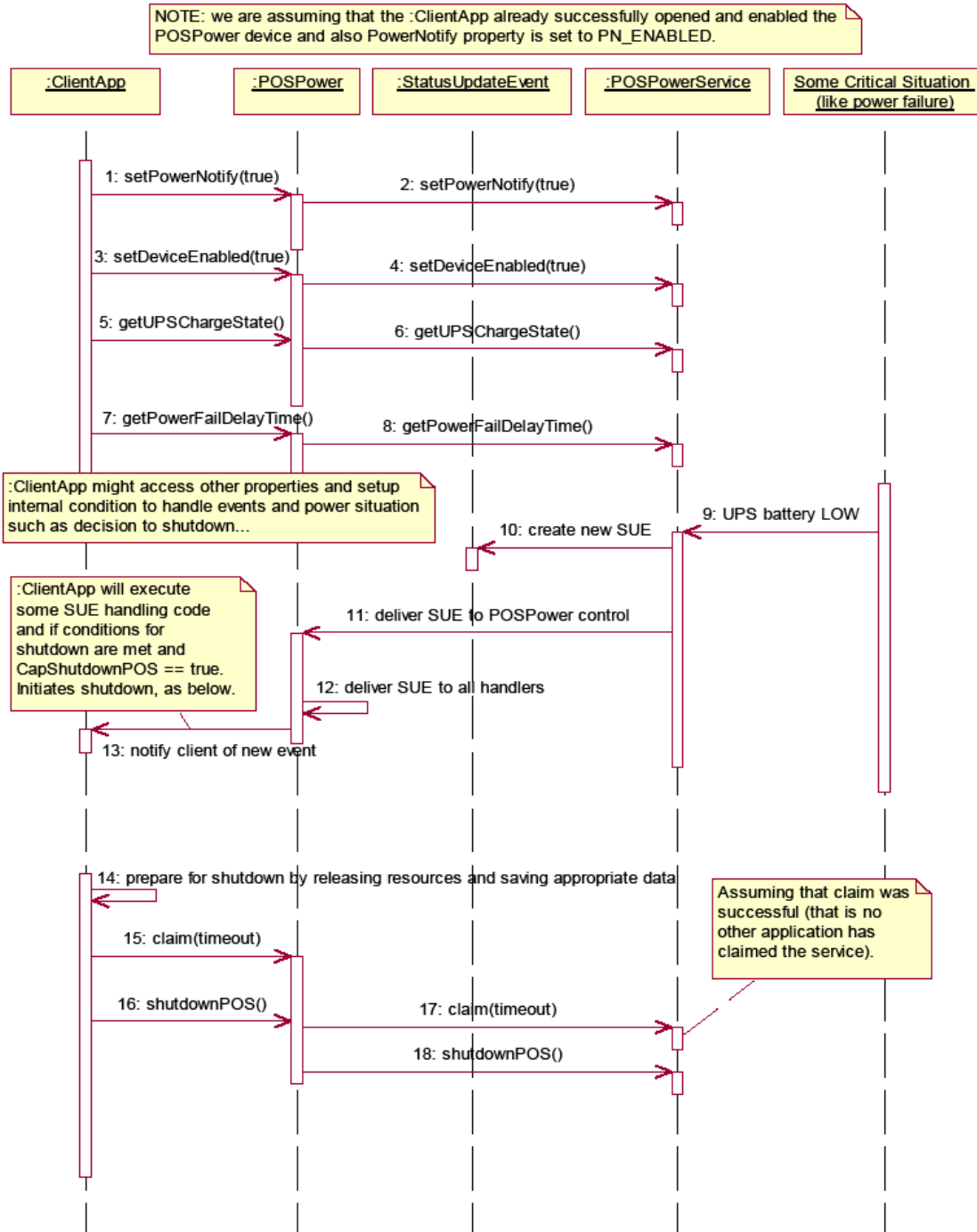


Fig. Chap. 29-2 POSPower Sequence Diagram

## POSPower Standby Sequence Diagram

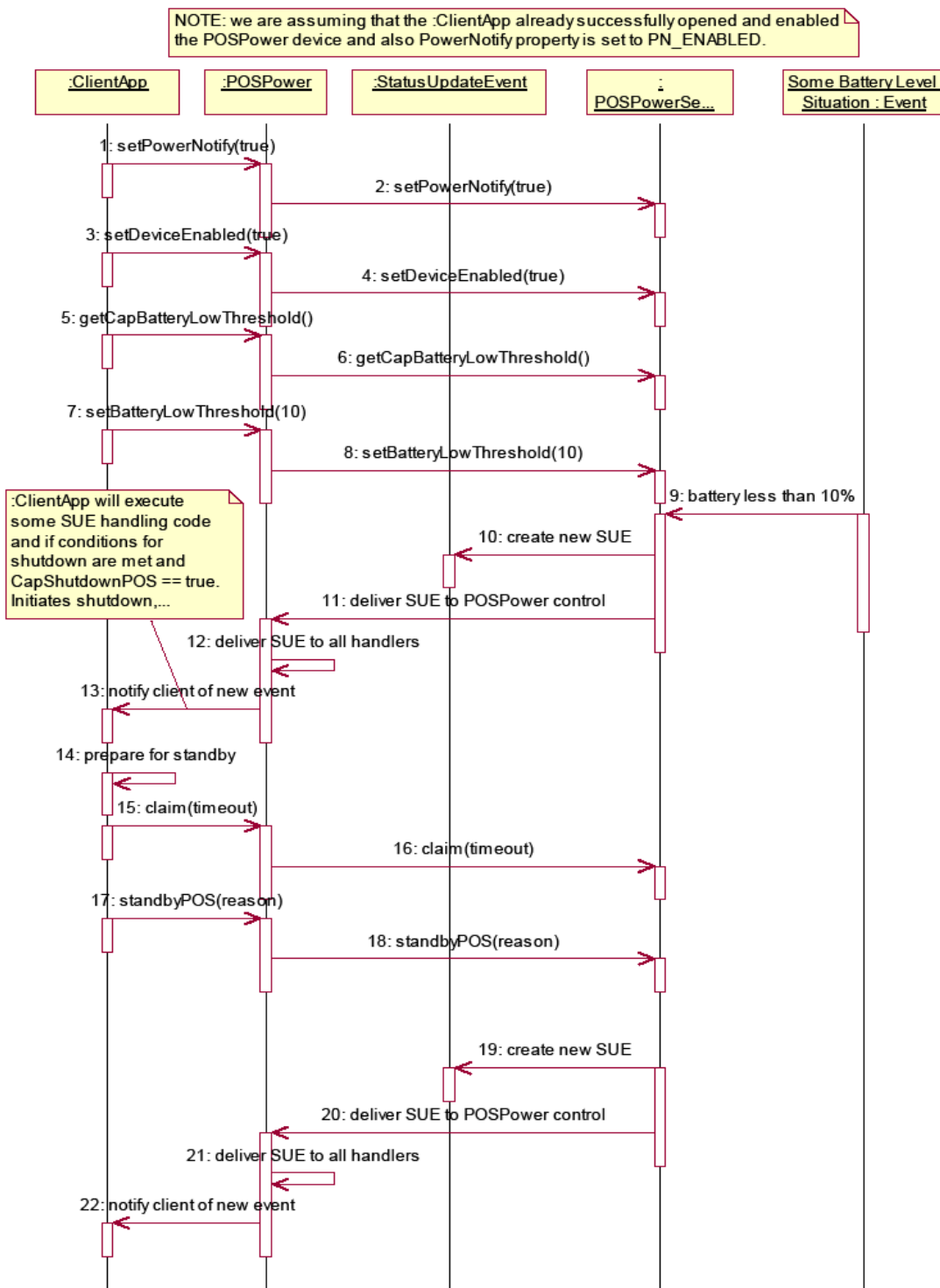


Fig. Chap. 29-3 POSPower Standby Sequence Diagram

## POSPower State Diagram

The following state diagram depicts the POSPower Control device model.

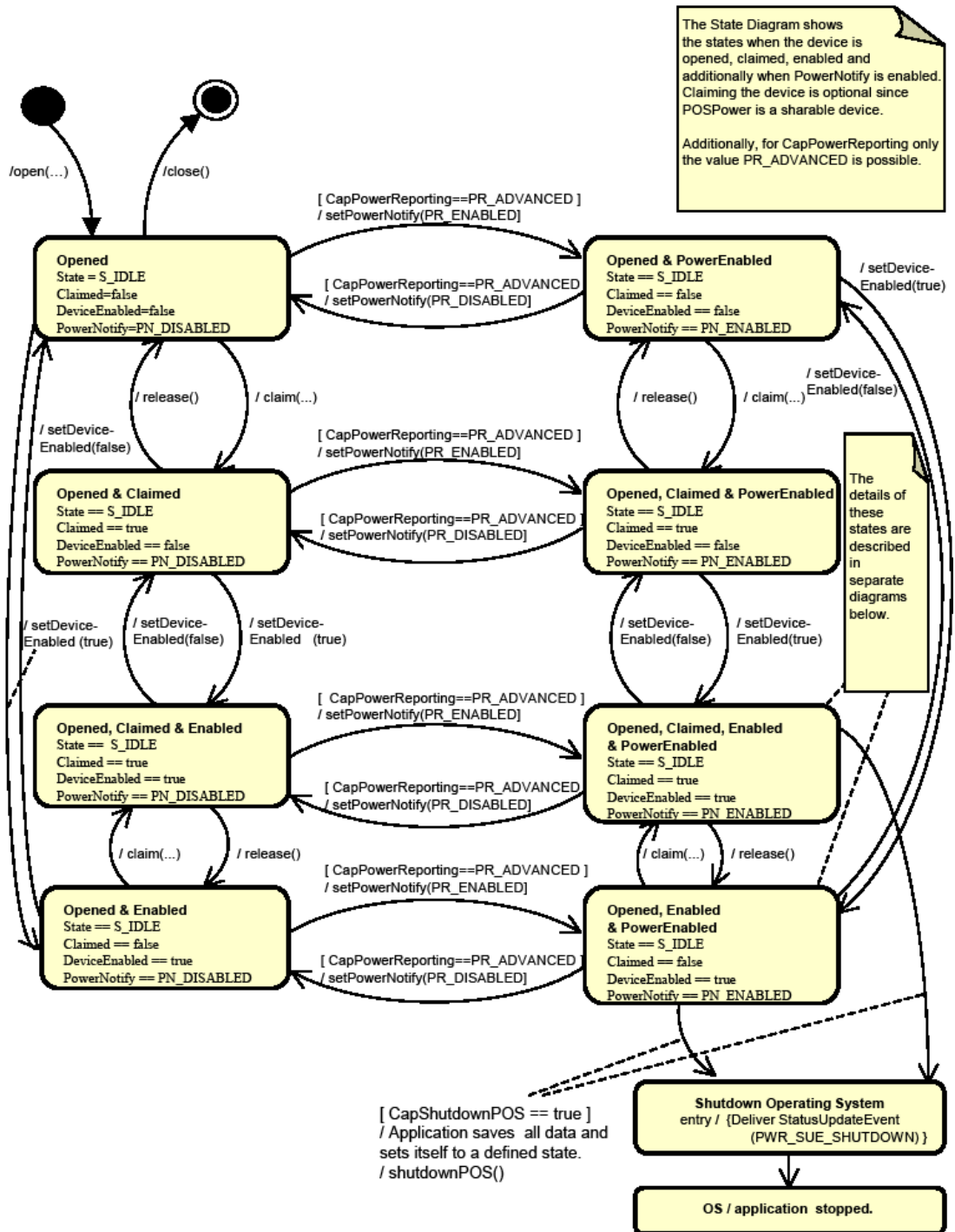


Fig. Chap. 29-4 Power State Diagram (POSPower Control Device Model)

# POSPower PowerState Diagram - Part 1

The following state diagram depicts the POSPower Power States.

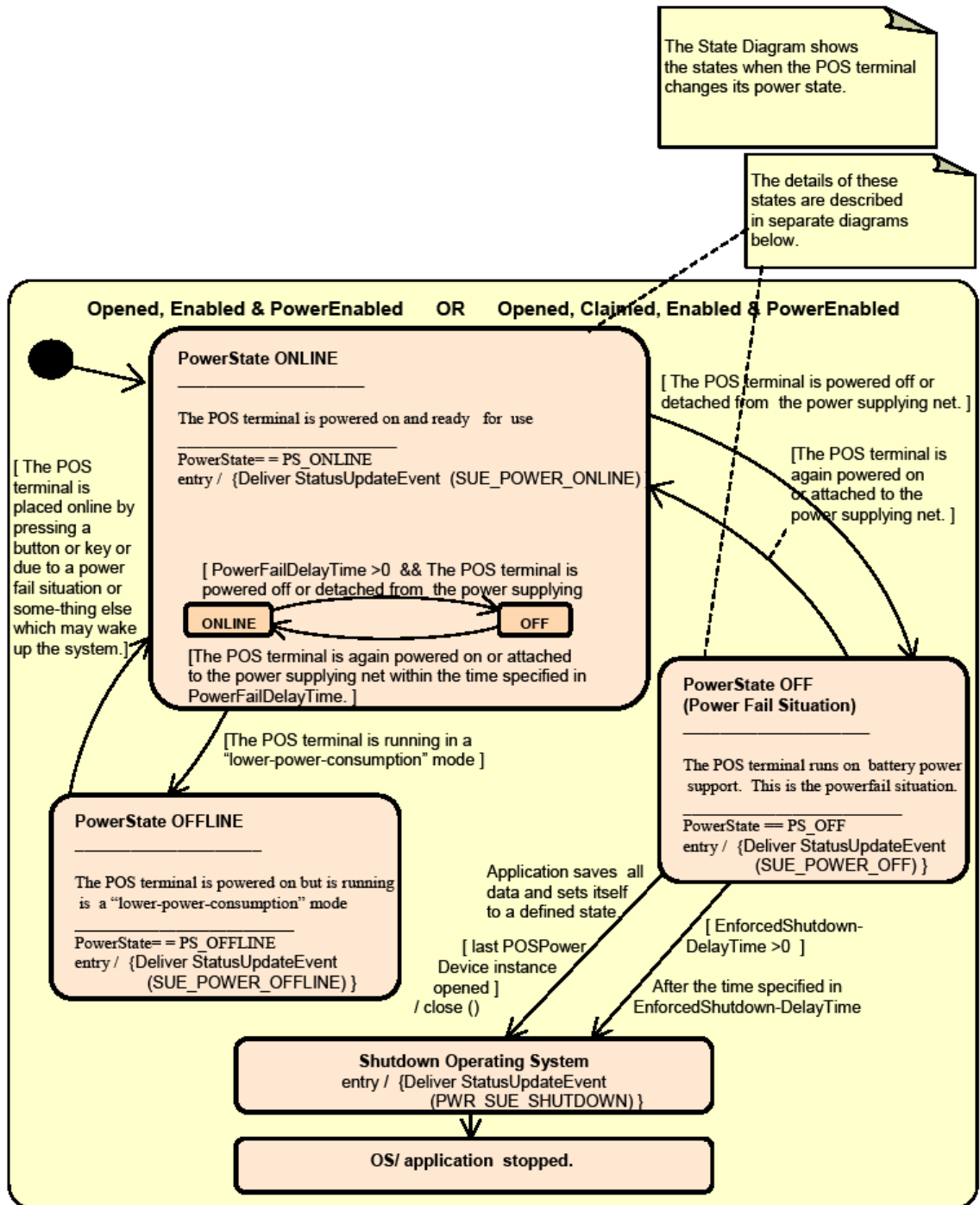


Fig. Chap. 29-5 POSPower PowerState Diagram (Part 1)

## 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.

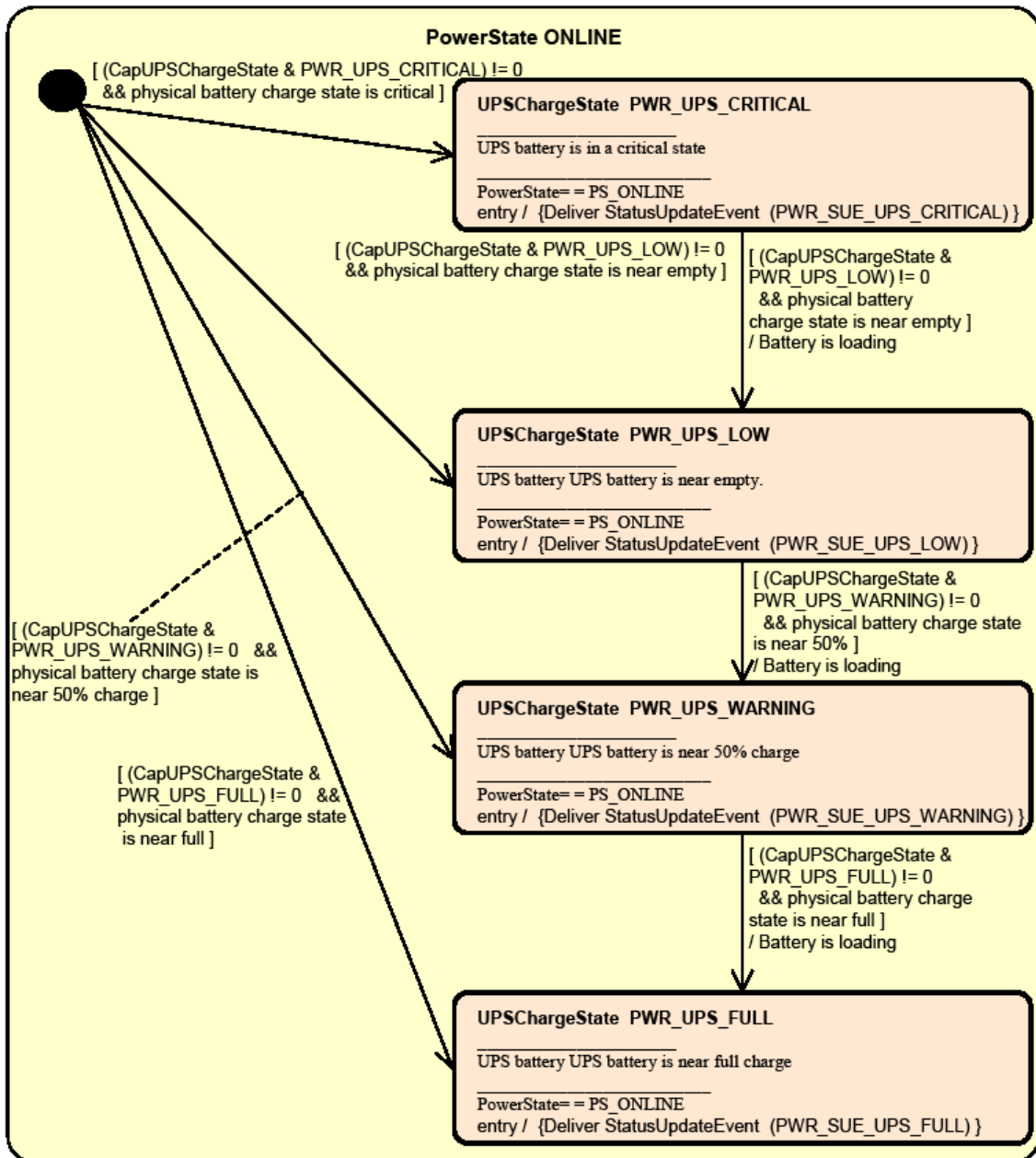


Fig. Chap. 29-6 POSPower PowerState Diagram (Part 2)

## 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.

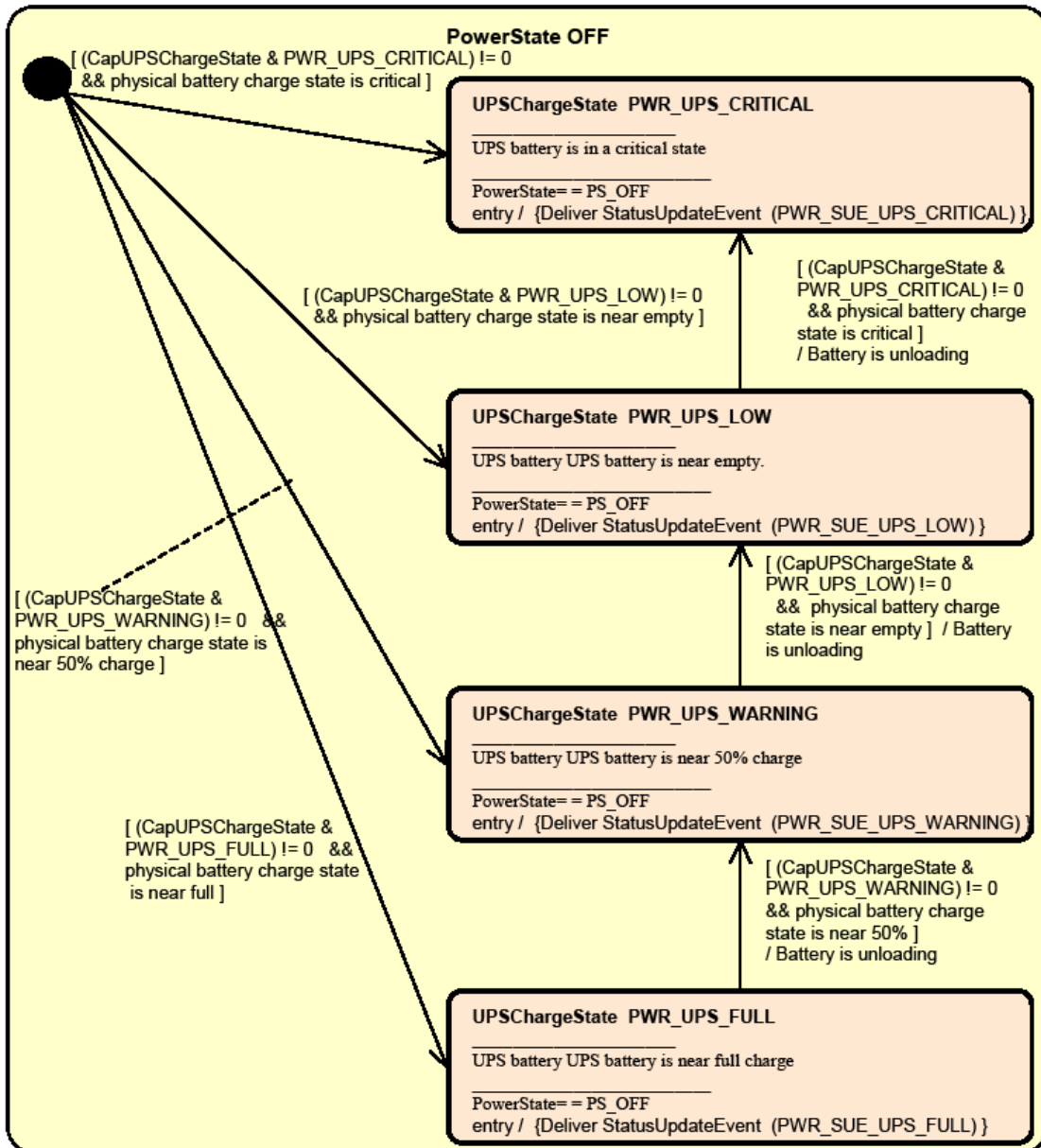


Fig. Chap. 29-7 POSPower PowerState Diagram (Part 3)

## 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.

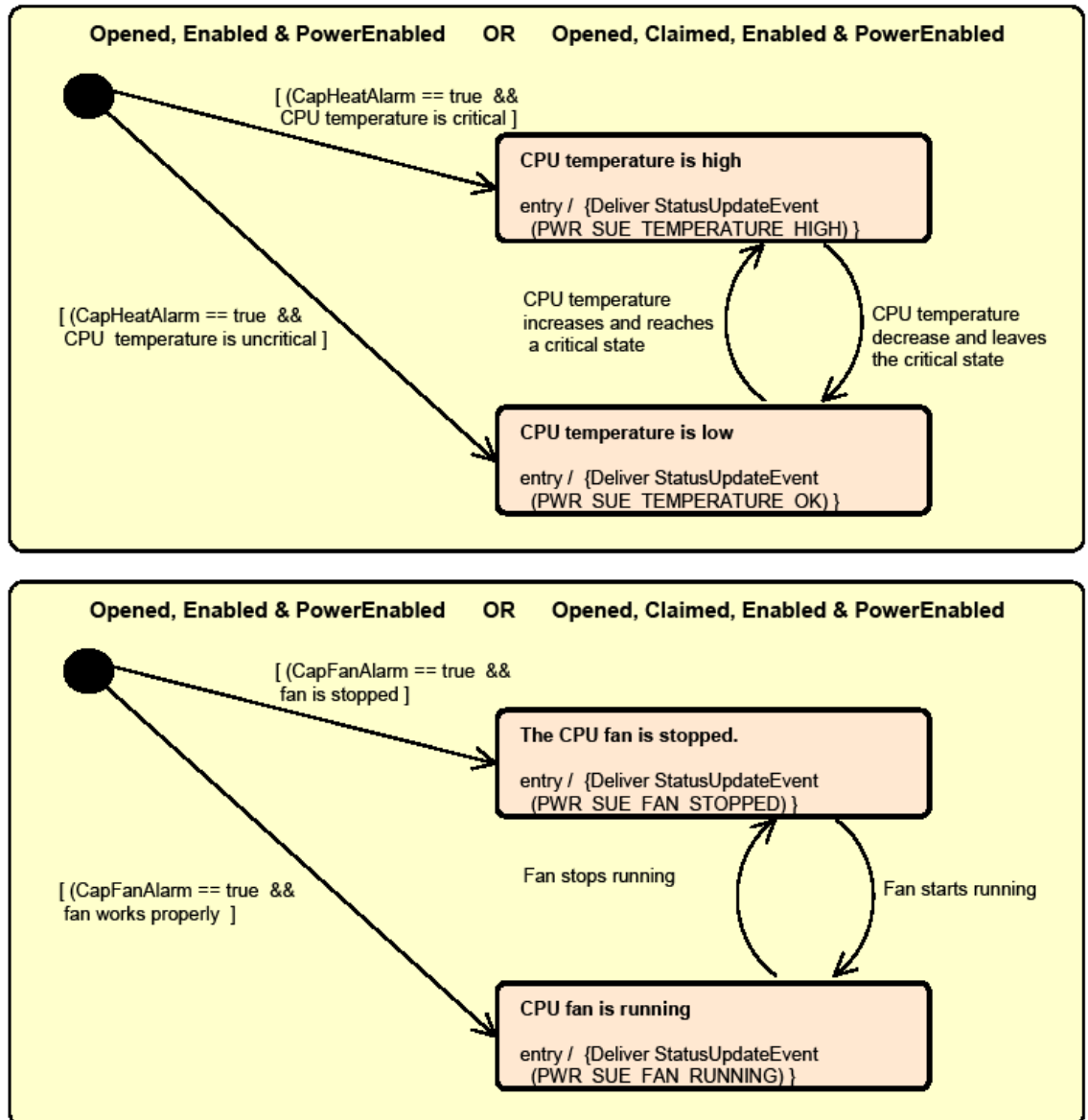


Fig. Chap. 29-8 POSPower State Chart Diagram (Fan and Temperature)



## POSPower Battery State Diagram

Illustrates the transition of states when the POS is only powered by the battery. It is assumed that the battery threshold is already set.

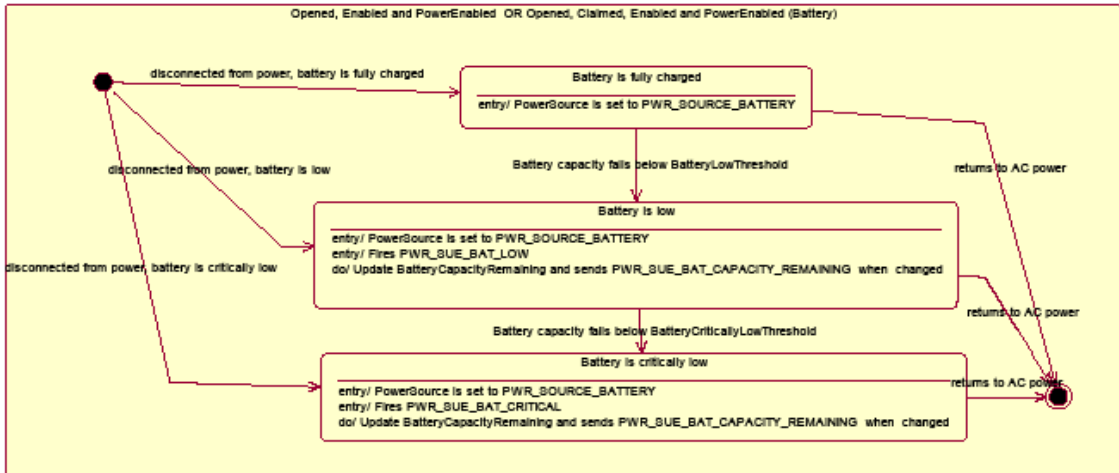


Fig. Chap. 29-9 POSPower Battery State Diagram

## POSPower Power Transitions State Diagram

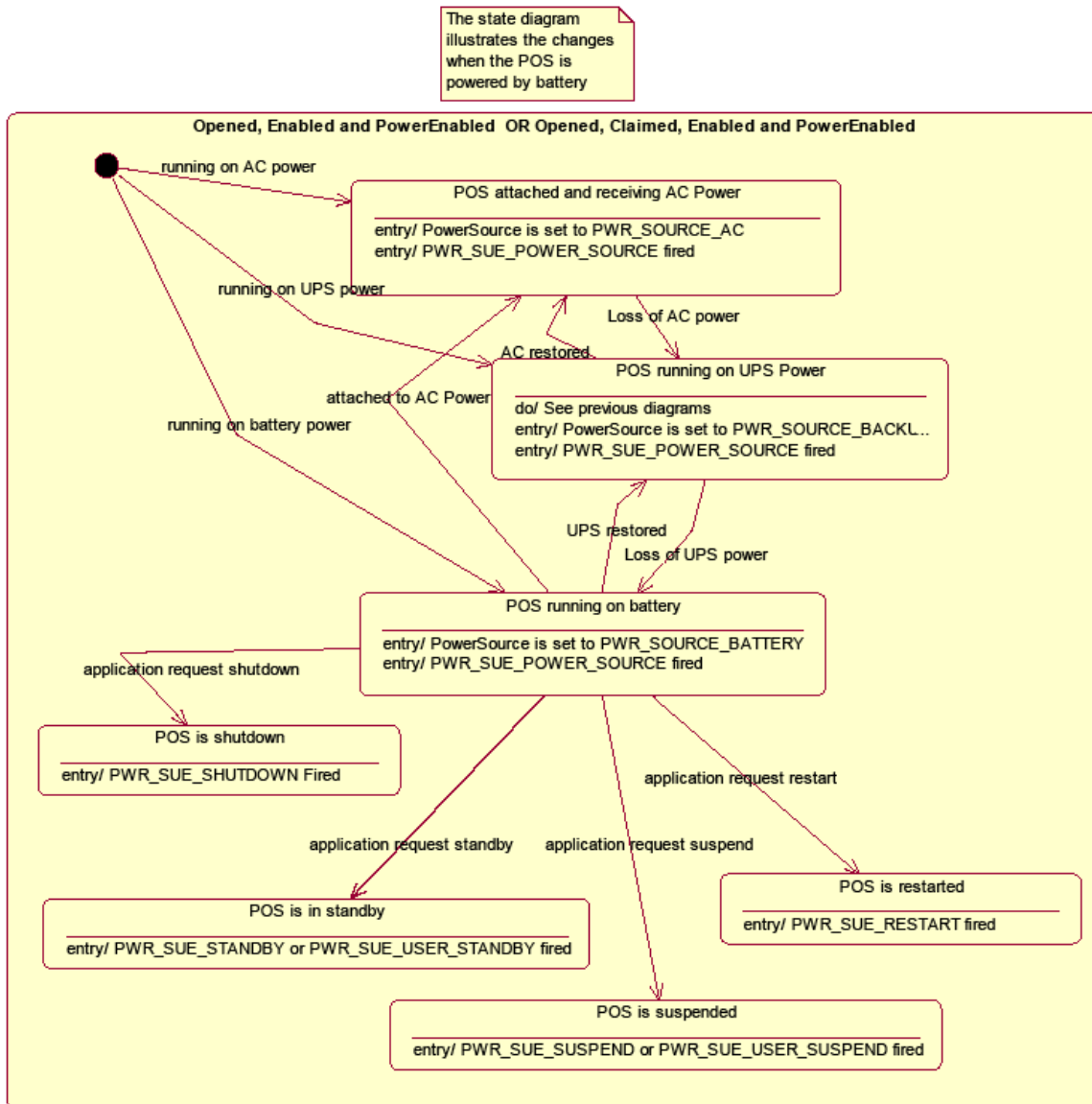


Fig. Chap. 29-10 POSPower Power Transitions State Diagram

## Properties (UML attributes)

### BatteryCapacityRemaining Property

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

### BatteryCapacityRemainingInSeconds Property *Added in Release 1.16*

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

### BatteryCriticallyLowThreshold Property

[Goto Table8](#)

<b>Syntax</b>	<b>BatteryCriticallyLowThreshold:</b> <i>int32</i> {read-write, access after open}
<b>Remarks</b>	If not zero, this property holds the threshold at which a PWR_SUE_BAT_CRITICAL <b>StatusUpdateEvent</b> 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 <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. <b>Some possible values of the exception’s ErrorCode property are:</b>

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

<b>See Also</b>	<b>CapVariableBatteryCriticallyLowThreshold</b> Property, <b>StatusUpdateEvent</b>
-----------------	--

## BatteryCriticallyLowThresholdInSeconds Property

*Added in Release 1.16*

<b>Syntax</b>	<b>BatteryCriticallyLowThresholdInSeconds: <i>int32</i> {read-write, access after open}</b>				
<b>Remarks</b>	<p>If not zero, this property holds the threshold at which a PWR_SUE_BAT_CRITICAL <b>StatusUpdateEvent</b> 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.</p> <p>This property is initialized by the <b>open</b> method.</p>				
<b>Errors</b>	<p>A UposException may be thrown when this property is accessed. For further information, see “<b>Errors</b>” on page Intro-20. <b>Some possible values of the exception’s <i>ErrorCode</i> property are:</b></p> <table border="0" style="width: 100%;"> <thead> <tr> <th style="text-decoration: underline;">Value</th> <th style="text-decoration: underline;">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.				
<b>See Also</b>	<b>CapVariableBatteryCriticallyLowThresholdInSeconds</b> Property, <b>StatusUpdateEvent</b>				

[Goto Table9](#)

## BatteryLowThreshold Property

<b>Syntax</b>	<b>BatteryLowThreshold: <i>int32</i> {read-write, access after open}</b>				
<b>Remarks</b>	<p>If not zero, this property holds the threshold at which a PWR_SUE_BAT_LOW <b>StatusUpdateEvent</b> 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.</p> <p>This property is initialized by the <b>open</b> method.</p>				
<b>Errors</b>	<p>A UposException may be thrown when this property is accessed. For further information, see “<b>Errors</b>” on page Intro-20. <b>Some possible values of the exception’s <i>ErrorCode</i> property are:</b></p> <table border="0" style="width: 100%;"> <thead> <tr> <th style="text-decoration: underline;">Value</th> <th style="text-decoration: underline;">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.				
<b>See Also</b>	<b>CapVariableBatteryLowThreshold</b> Property, <b>StatusUpdateEvent</b>				

[Goto Table10](#)

**Syntax**      **BatteryLowThresholdInSeconds: *int32* {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. *Some possible values of the exception's **ErrorCode** property are:*

[Goto Table11](#)

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

**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**

### CapBatteryCapacityRemaining Property

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

### CapBatteryCapacityRemainingInSeconds Property

***Added in Release 1.16***

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

### CapChargeTime Property

***Added in Release 1.16***

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

### CapFanAlarm Property

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

### CapHeatAlarm Property

- Syntax**      **CapHeatAlarm: *boolean* {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: *boolean* {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: *boolean* {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: *boolean* {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

<b>Syntax</b>	<b>CapStandbyPOS:</b> <i>boolean</i> { <b>read-only, access after open</b> }
<b>Remarks</b>	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 <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>standbyPOS</b> Method.

### CapSuspendPOS Property

<b>Syntax</b>	<b>CapSuspendPOS:</b> <i>boolean</i> { <b>read-only, access after open</b> }
<b>Remarks</b>	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 <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>suspendPOS</b> Method.

### CapUPSChargeState Property

<b>Syntax</b>	<b>CapUPSChargeState:</b> <i>int32</i> { <b>read-only, access after open</b> }										
<b>Remarks</b>	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. <table><thead><tr><th><b>Value</b></th><th><b>Meaning</b></th></tr></thead><tbody><tr><td>PWR_UPS_FULL</td><td>UPS battery is near full charge.</td></tr><tr><td>PWR_UPS_WARNING</td><td>UPS battery is near 50% charge.</td></tr><tr><td>PWR_UPS_LOW</td><td>UPS battery is near empty. Application shutdown should be started to ensure that 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.</td></tr><tr><td>PWR_UPS_CRITICAL</td><td>UPS battery is in a critical state and could be disconnected at any time without further warning. This property is initialized by the <b>open</b> method.</td></tr></tbody></table>	<b>Value</b>	<b>Meaning</b>	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 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 <b>open</b> method.
<b>Value</b>	<b>Meaning</b>										
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 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 <b>open</b> method.										
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.										
<b>See Also</b>	<b>UPSChargeState</b> Property.										



### CapVariableBatteryCriticallyLowThreshold Property

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

### CapVariableBatteryCriticallyLowThresholdInSeconds Property

*Added in Release 1.16*

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

### CapVariableBatteryLowThreshold Property

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

### CapVariableBatteryLowThresholdInSeconds Property

*Added in Release 1.16*

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

### 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. **Some possible values of the exception’s ErrorCode property are:**
- | <b>Value</b>     | <b>Meaning</b>   |
|------------------|--|
| <b>E_ILLEGAL</b> | An invalid value was specified.<br>Or it does not support this function. |
- [Goto Table12](#)
- 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

<b>Syntax</b>	<b>PowerSource: <i>int32</i> {read-only, access after open}</b>										
<b>Remarks</b>	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>	<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.										
	This property is initialized by the <b>open</b> method.										
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.										
<b>See Also</b>	<b>StatusUpdateEvent</b>										

### QuickChargeMode Property

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

### QuickChargeTime Property

<b>Syntax</b>	<b>QuickChargeTime: <i>int32</i> {read-only, access after open}</b>
<b>Remarks</b>	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.  This time is specified in milliseconds.  This property is only set if <b>CapQuickCharge</b> is true.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>CapQuickCharge</b> Property, <b>QuickChargeTime</b> Property.

## UPOS Ver1.16 RCSD Specification

### 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:

<u>Value</u>	<u>Meaning</u>
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 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.

## Methods (UML operations)

### restartPOS Method

[Goto Table 151](#)

<b>Syntax</b>	<b>restartPOS ( ):void {raises-exception, use after open,enable}</b>				
<b>Remarks</b>	<p>Call to restart the POS terminal. This method will always restart the system independent of the system power state.</p> <p>If the POSPower is claimed, only the application which claimed the device is able to restart the POS terminal.</p> <p>Applications will be informed about an initiated restart.</p>				
<b>Errors</b>	<p>A UposException may be thrown when this method is invoked. For further information, see “<b>Errors</b>” on page Intro-20</p> <p>Some possible values of the exception’s <i>ErrorCode</i> property are:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><b>Value</b></th> <th style="text-align: left;"><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td>E_ILLEGAL</td> <td>This method is not supported (see the <b>CapRestartPOS</b> property)</td> </tr> </tbody> </table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	This method is not supported (see the <b>CapRestartPOS</b> property)
<b>Value</b>	<b>Meaning</b>				
E_ILLEGAL	This method is not supported (see the <b>CapRestartPOS</b> property)				
<b>See Also</b>	<b>CapRestartPOS</b> Property				

### shutdownPOS Method

[Goto Table 152](#)

<b>Syntax</b>	<b>shutdownPOS ( ):void {raises-exception, use after open,enable}</b>				
<b>Remarks</b>	<p>Call to shut down the POS terminal. This method will always shut down the system independent of the system power state.</p> <p>If the POSPower is claimed, only the application which claimed the device is able to shut down the POS terminal.</p> <p>Applications will be informed about an initiated shutdown.</p> <p>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.</p> <p>If the <b>EnforcedShutdownDelayTime</b> property specifies a time greater than zero and the application did not call the <b>shutdownPOS</b> method within the time specified in <b>EnforcedShutdownDelayTime</b>, 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.</p> <p>This method is only supported if <b>CapShutdownPOS</b> is true.</p>				
<b>Errors</b>	<p>A UposException may be thrown when this method is invoked. For further information, see “<b>Errors</b>” on page Intro-20</p> <p>Some possible values of the exception’s <i>ErrorCode</i> property are:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><b>Value</b></th> <th style="text-align: left;"><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td>E_ILLEGAL</td> <td>This method is not supported. (See the <b>CapShutdownPOS</b> property)</td> </tr> </tbody> </table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	This method is not supported. (See the <b>CapShutdownPOS</b> property)
<b>Value</b>	<b>Meaning</b>				
E_ILLEGAL	This method is not supported. (See the <b>CapShutdownPOS</b> property)				
<b>See Also</b>	<b>CapShutdownPOS</b> Property, <b>EnforcedShutdownDelayTime</b> Property.				

**UPOS Ver1.16 RCSD Specification**  
**standbyPOS Method**

[Goto Table 153](#)

**Syntax**     **standbyPOS (reason: *int32*):**  
    **void {raises-exception, use after open<sub>5</sub>-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:

<b>Value</b>	<b>Description</b>
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:

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

**See Also**     **CapStandbyPOS** Property.

**suspendPOS Method**

[Goto Table 154](#)

**Syntax**     **suspendPOS (reason: *int32*):**  
    **void {raises-exception, use after open<sub>5</sub>-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:

<b>Value</b>	<b>Description</b>
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:

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	This method is not supported (see the <b>CapSuspendPOS</b> 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 device control.

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

<u>Attributes</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 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.

### 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:

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

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

<u>Value</u>	<u>Meaning</u>
PWR_SUE_UPS_FULL	UPS battery is near full charge. Can be returned if <b>CapUPSChargeState</b> 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 <b>CapUPSChargeState</b> contains PWR_UPS_LOW.

## UPOS Ver1.16 RCSD Specification

### PWR\_SUE\_UPS\_CRITICAL

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.



## UPOS Ver1.16 RCSD Specification

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

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>
<b>AutoDisable:</b>	<i>boolean</i>	{read-write}	1.16	<i>Not supported</i>
<b>CapCompareFirmwareVersion:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPowerReporting:</b>	<i>int32</i>	{read-only}	1.16	open
<b>CapStatisticsReporting:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateFirmware:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateStatistics:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CheckHealthText:</b>	<i>string</i>	{read-only}	1.16	open
<b>Claimed:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>DataCount:</b>	<i>int32</i>	{read-only}	1.16	<i>Not supported</i>
<b>DataEventEnabled:</b>	<i>boolean</i>	{read-write}	1.16	<i>Not supported</i>
<b>DeviceEnabled:</b>	<i>boolean</i>	{read-write}	1.16	open & claim
<b>FreezeEvents:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>OutputID:</b>	<i>int32</i>	{read-only}	1.16	<i>Not supported</i>
<b>PowerNotify:</b>	<i>int32</i>	{read-write}	1.16	open
<b>PowerState:</b>	<i>int32</i>	{read-only}	1.16	open
<b>State:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceControlDescription:</b>	<i>string</i>	{read-only}	1.16	--
<b>DeviceControlVersion:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceServiceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>DeviceServiceVersion:</b>	<i>int32</i>	{read-only}	1.16	open
<b>PhysicalDeviceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>PhysicalDeviceName:</b>	<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>
<b>CapAssociatedHardTotalsDevice:</b>	<i>string</i>	{read-only}	1.16	open
<b>CapAutoExposure:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapAutoFocus:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapAutoGain:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapAutoWhiteBalance:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapBrightness:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapContrast:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapExposure:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapGain:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapHorizontalFlip:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapHue:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPhoto:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPhotoColorSpace:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPhotoFrameRate:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPhotoResolution:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPhotoType:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapSaturation:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapStorage:</b>	<i>int32</i>	{read-only}	1.16	open
<b>CapVerticalFlip:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVideo:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVideoColorSpace:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVideoFrameRate:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVideoResolution:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVideoType:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>AutoExposure:</b>	<i>boolean</i>	{read-write}	1.16	open, claim & enable
<b>AutoFocus:</b>	<i>boolean</i>	{read-write}	1.16	open, claim & enable
<b>AutoGain:</b>	<i>boolean</i>	{read-write}	1.16	open, claim & enable
<b>AutoWhiteBalance:</b>	<i>boolean</i>	{read-write}	1.16	open, claim & enable
<b>Brightness:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>Contrast:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>Exposure:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>Gain:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable

## UPOS Ver1.16 RCSD Specification

<b>HorizontalFlip:</b>	<i>boolean</i>	{read-write}	1.16	open, claim & enable
<b>Hue:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>PhotoColorSpace:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>PhotoColorSpaceList:</b>	<i>string</i>	{read-only}	1.16	open
<b>PhotoFrameRate:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>PhotoMaxFrameRate:</b>	<i>int32</i>	{read-only}	1.16	open
<b>PhotoResolution:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>PhotoResolutionList:</b>	<i>string</i>	{read-only}	1.16	open
<b>PhotoType:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>PhotoTypeList:</b>	<i>string</i>	{read-only}	1.16	open
<b>RemainingRecordingTimeInSec:</b>	<i>int32</i>	{read-only}	1.16	open, claim & enable
<b>Saturation:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>Storage:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>VerticalFlip:</b>	<i>boolean</i>	{read-write}	1.16	open, claim & enable
<b>VideoCaptureMode:</b>	<i>int32</i>	{read-write <del>only</del> }	1.16	open, claim & enable
<b>VideoColorSpace:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>VideoColorSpaceList:</b>	<i>string</i>	{read-only}	1.16	open
<b>VideoFrameRate:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>VideoMaxFrameRate:</b>	<i>int32</i>	{read-only}	1.16	open
<b>VideoResolution:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>VideoResolutionList:</b>	<i>string</i>	{read-only}	1.16	open
<b>VideoType:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>VideoTypeList:</b>	<i>string</i>	{read-only}	1.16	open

[Goto Table13](#)

## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)

#### Common

<i>Name</i>	<i>Version</i>	
<b>open</b> (logicalDeviceName: <i>string</i> ): void {raises-exception}	1.16	
<b>close</b> (): void {raises-exception, use after open}	1.16	
<b>claim</b> (timeout: <i>int32</i> ): void {raises-exception, use after open}	1.16	
<b>release</b> (): void {raises-exception, use after open, claim}	1.16	
<b>checkHealth</b> (level: <i>int32</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table14</a>
<b>clearInput</b> (): void {raises-exception, use after open, <b>claim</b> }	1.16	<a href="#">Goto Table15</a>
<b>clearInputProperties</b> (): void { }	<i>Not supported</i>	
<b>clearOutput</b> (): void { }	<i>Not supported</i>	
<b>directIO</b> (command: <i>int32</i> , inout data: <i>int32</i> , inout obj: <i>object</i> ): void {raises-exception, use after open}	1.16	
<b>compareFirmwareVersion</b> (firmwareFileName: <i>string</i> , out result: <i>int32</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table16</a>
<b>resetStatistics</b> (statisticsBuffer: <i>string</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table17</a>
<b>retrieveStatistics</b> (inout statisticsBuffer: <i>string</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table18</a>
<b>updateFirmware</b> (firmwareFileName: <i>string</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table19</a>
<b>updateStatistics</b> (statisticsBuffer: <i>string</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table20</a>

#### Specific

<i>Name</i>	<i>Version</i>
<b>startVideo</b> (fileName: <i>string</i> , overwrite: <i>boolean</i> , recordingTime: <i>int32</i> ): void {raises-exception, use after open, claim, enable}	1.16
<b>stopVideo</b> (): void {raises-exception, use after open, claim, enable}	1.16
<b>takePhoto</b> (fileName: <i>string</i> , overwrite: <i>boolean</i> , timeout: <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>
<b>upos::events::DataEvent</b>		<i>Not supported</i>	
<b>Status:</b>			
<b>upos::events::DirectIOEvent</b>			1.16
<b>EventNumber:</b>	<i>int32</i>	{ read-only }	
<b>Data:</b>	<i>int32</i>	{ read-write }	
<b>Obj:</b>	<i>object</i>	{ read-write }	
<b>upos::events::ErrorEvent</b>			1.16
<b>ErrorCode:</b>	<i>int32</i>	{ read-only }	
<b>ErrorCodeExtended:</b>	<i>int32</i>	{ read-only }	
<b>ErrorLocus:</b>	<i>int32</i>	{ read-only }	
<b>ErrorResponse</b>	<i>int32</i>	{ read-write }	
<b>upos::events::OutputCompleteEvent</b>		<i>Not supported</i>	
<b>upos::events::StatusUpdateEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{ read-only }	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	1.16

## **General Information**

The Video Capture Device name is “Video Capture”.

### **Capabilities**

Video capture device class has the following capabilities:

- Take a photo and record it as a file in a host and may store it in the targeted storage device.
- Take a video and record it as a file in a host and may store it in the targeted storage device.
- May 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.

# UPOS Ver1.16 RCSD Specification

## Video Capture Class Diagram

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

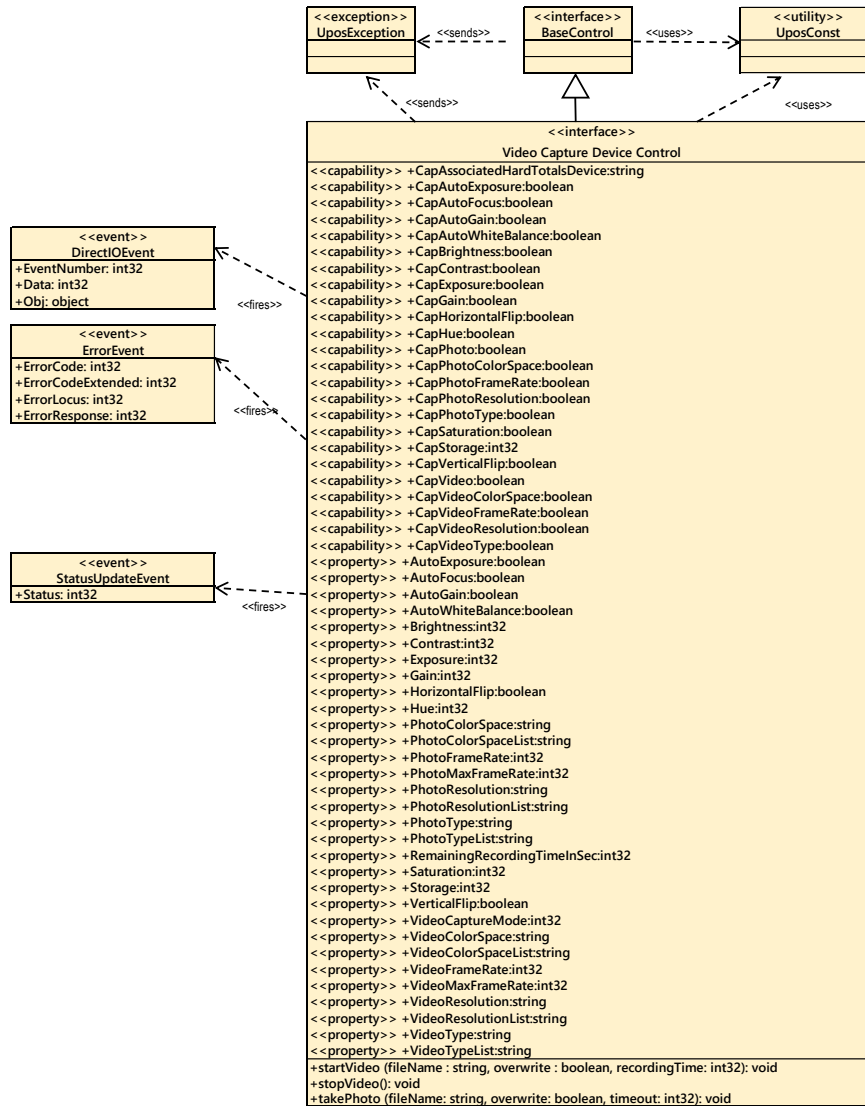


Fig. Chap. 39-1 Video Capture Class Diagram



# UPOS Ver1.16 RCSD Specification

## Model

### Modes

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.

### 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\_HARDTOTALS\_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.

[Goto Table21](#)

This method is ~~performed~~~~executed~~ synchronously as the process of taking photo. The process of recorded data storing is performed asynchronously. **StatusUpdateEvents** are delivered to the application when the start and the end of device states are changed. Only one call to **takePhoto** method can be in progress at a time. If you try ~~An attempt~~ to nest the video capture device operation of the device, before the storing is finished, ~~will result in~~ an UPOSException will be ~~ing~~ thrown.

[Goto Table22](#)

## UPOS Ver1.16 RCSD Specification

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\_RETRYCONTINUEINPUT, the process of recorded data storing was retried ~~input processing continues~~. However, as long as the cause of the error is not resolved, the **ErrorEvent** will occur again immediately.

[Goto Table23](#)

If **ErrorEvent** is ER\_CLEAR, all of the device buffered data is cleared ~~the input processing process is terminated~~ and the **takePhoto** method ~~taking photo~~ is discarded.

[Goto Table24](#)

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. This method is executed synchronously. During video image capturing, recorded data storing is processed asynchronously and when the start and end the device state is changed, **StatusUpdateEvents** are delivered to the application. In addition, remaining device recording time is updated in the **RemainingRecordingTimeInSec** property.

[Goto Table25](#)

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\_HARD+TOTALS\_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.

[Goto Table26](#)

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

[Goto Table27](#)

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\_STOPEND\_VIDEO is evoked to notify the application that taking video has been ended.

[Goto Table146](#)

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

## UPOS Ver1.16 RCSD Specification

VCAP\_CST\_HARD~~WARE~~TOTALS\_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. [Goto Table28](#)

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\_HARD~~WARE~~TOTALS\_ONLY or VCAP\_CST\_ALL, and can be delivered to the application.

If **ErrorEvent** response is ER\_RETRY~~CONTINUEINPUT~~, the process of **recorded data storing was retried** ~~input processing continues~~. However, as long as the cause of the error is not resolved, the **ErrorEvent** will occur again immediately. [Goto Table29](#)

If **ErrorEvent** is ER\_CLEAR, **all of the device buffered data is cleared and the error state is exited** ~~input processing is terminated~~ and the taking **video capturing process** is discarded. [Goto Table30](#)

An **ErrorEvent** event (or events) is enqueued if an error occurs while gathering or processing **the data** ~~input~~. [Goto Table31](#)

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\_HARD~~WARE~~TOTALS\_ONLY or VCAP\_CST\_ALL, and can deliver it to the application. [Goto Table32](#)

All enqueued **data** ~~input~~ may be deleted by calling **clearInput** method. See the clearInput method description for more details. [Goto Table33](#)

## 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.

## Properties (UML attributes)

### AutoExposure Property

**Syntax**     **AutoExposure: *boolean* {read-write, access after open-claim-enable}**

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

[Goto Table34](#)

**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:

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

**See also**     **CapAutoExposure** Property

### AutoFocus Property

**Syntax**     **AutoFocus: *boolean* {read-write, access after open-claim-enable}**

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

[Goto Table35](#)

**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:

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

**See also**     **CapAutoFocus** Property

### AutoGain Property

**Syntax**     **AutoGain: *boolean* {read-write, access after open-claim-enable}**

[Goto Table36](#)

**Remarks**    If true, auto gain of camera is enabled. Otherwise, it is false.  
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 *ErrorCode* property are:

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

**See also**     **CapAutoGain** Property   **Gain** Property

## UPOS Ver1.16 RCSD Specification

### AutoWhiteBalance Property

**Syntax**      **AutoWhiteBalance:** *boolean* {**read-write, access after open-claim-enable**}

[Goto Table37](#)

**Remarks**      If true, auto white balance of camera is enabled. 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:

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

**See also**      **CapAutoWhiteBalance** Property

### Brightness property

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

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

[Goto Table38](#)

**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**      **CapBrightness** Property

### CapAssociatedHardTotalsDevice Property

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

[Goto Table39](#)

**Remarks**      Indicate that the device is able to store the recorded data into the Associated Hard Totals device and holds its 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 the device is not able to store the data into the Associated Hard Totals device and this property value must be the empty string. 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**      **CapStorage** Property

### CapAutoExposure Property

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

**Remarks**      If true, the auto exposure of camera can 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**      **AutoExposure** Property

### CapAutoFocus Property

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

### CapAutoGain Property

<b>Syntax</b>	<b>CapAutoGain: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, automatic gain change of the camera is possible. Otherwise, it is false. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>AutoGain</b> Property

### CapAutoWhiteBalance Property

<b>Syntax</b>	<b>CapAutoWhiteBalance: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, auto white balance of camera is possible. Otherwise, it is false. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>AutoWhiteBalance</b> Property

### CapBrightness Property

<b>Syntax</b>	<b>CapBrightness: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, the brightness of camera can be changed. Otherwise, it is false. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>Brightness</b> Property

### CapContrast Property

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

## UPOS Ver1.16 RCSD Specification

### CapExposure Property

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

### CapGain Property

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

### CapHorizontalFlip Property

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

### CapHue Property

<b>Syntax</b>	<b>CapHue: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, the hue of the camera can be changed. Otherwise, it is false. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>Hue Property</b>

### CapPhoto Property

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

## UPOS Ver1.16 RCSD Specification

### CapPhotoColorSpace Property

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

### CapPhotoFrameRate Property

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

### CapPhotoResolution Property

<b>Syntax</b>	<b>CapPhotoResolution: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, taking photo resolution is handled and can be changed. Otherwise, it is false. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>PhotoResolution</b> Property

### CapPhotoType Property

<b>Syntax</b>	<b>CapPhotoType: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, photo image format type can be changed. Otherwise, it is false. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>PhotoType</b> Property

[Goto Table40](#)

### CapSaturation Property

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



### 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 **video or photo** data file to. It holds one of the following values.

[Goto Table41](#)

<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

### CapVerticalFlip Property

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

**Remarks** If true, can 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** VerticalFlip Property

### CapVideo Property

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

**Remarks** If true, video function is supported. Otherwise, it is false. If this property is true, taking video and recording can be done by calling the **startVideo** method. And to activate the video mode, the **VideoCaptureMode** property value needs to set VCAP\_VCMODE\_VIDEO. If false, 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** StartVideo Method, VideoCaptureMode Property

## UPOS Ver1.16 RCSD Specification

### CapVideoColorSpace Property

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

### CapVideoFrameRate Property

<b>Syntax</b>	<b>CapVideoFrameRate: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, can change the video frame rate from 1 to up to <b>VideoMaxFrameRate</b> property value. Otherwise, it is false. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>VideoMaxFrameRate</b> Property, <b>VideoFrameRate</b> Property

### CapVideoResolution Property

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

### CapVideoType Property

<b>Syntax</b>	<b>CapVideoType: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, taking video type can be changed, and all of possible values are listed in the <b>VideoTypeList</b> values. Otherwise, it is false. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>VideoTypeList</b> Property, <b>VideoType</b> Property

### Contrast Property

[Goto Table42](#)

<b>Syntax</b>	<b>Contrast: <i>int32</i> {read-write, access after open-claim-enable}</b>				
<b>Remarks</b>	Indicate the contrast of the camera. Valid values range from 0 to 100. This property is initialized by the <b>open</b> method.				
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><b>Value</b></th> <th style="text-align: left;"><b>Meaning</b></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>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
<b>Value</b>	<b>Meaning</b>				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
<b>See Also</b>	<b>CapContrast</b> Property				

### Exposure Property

[Goto Table43](#)

<b>Syntax</b>	<b>Exposure: <i>int32</i> {read-write, access after open-claim-enable}</b>				
<b>Remarks</b>	Indicate the exposure of camera. Valid values range from 0 to 100. This property is initialized by the <b>open</b> method.				
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><b>Value</b></th> <th style="text-align: left;"><b>Meaning</b></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>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
<b>Value</b>	<b>Meaning</b>				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
<b>See also</b>	<b>CapExposure</b> Property				

### Gain Property

[Goto Table44](#)

<b>Syntax</b>	<b>Gain: <i>int32</i> {read-write, access after open-claim-enable}</b>				
<b>Remarks</b>	Indicate the gain of camera. Valid values range from 0 to 100. If <b>AutoGain</b> property is true, it is possible to read the value of <b>Gain</b> property. However, it is not possible to write and change the value of <b>Gain</b> property. If <b>AutoGain</b> property is false, then, it is possible to read, write and change the value of <b>Gain</b> property. This property is initialized by the <b>open</b> method.				
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><b>Value</b></th> <th style="text-align: left;"><b>Meaning</b></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>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
<b>Value</b>	<b>Meaning</b>				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
<b>See also</b>	<b>CapGain</b> Property, <b>AutoGain</b> Property				

## UPOS Ver1.16 RCSD Specification

### HorizontalFlip Property

<b>Syntax</b>	<b>HorizontalFlip: <i>boolean</i> {read-write, access after open-claim-enable}</b>				
<b>Remarks</b>	If true, horizontal flip of camera is enabled and it is possible to reverse the camera captured image horizontally. Otherwise, it is false. There is a similar property called <b>VerticalFlip</b> property. However, each <b>VerticalFlip</b> property and <b>HorizontalFlip</b> property value can be set independently. This property is initialized by the <b>open</b> method. <a href="#">Goto Table45</a>				
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><b>Value</b></th><th><b>Meaning</b></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>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
<b>Value</b>	<b>Meaning</b>				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
<b>See Also</b>	<b>CapHorizontalFlip</b> property, <b>VerticalFlip</b> property, <b>CapVerticalFlip</b> property				

### Hue Property

<b>Syntax</b>	<b>Hue: <i>int32</i> {read-write, access after open-claim-enable}</b>	<a href="#">Goto Table46</a>				
<b>Remarks</b>	Indicate the hue of camera. Valid values range from 0 to 100. This property is initialized by the <b>open</b> method.					
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><b>Value</b></th><th><b>Meaning</b></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>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.	
<b>Value</b>	<b>Meaning</b>					
E_ILLEGAL	An invalid value was specified. Or it does not support this function.					
<b>See also</b>	<b>CapHue</b> Property					

### PhotoColorSpace Property

<b>Syntax</b>	<b>PhotoColorSpace: <i>string</i> {read-write, access after open-claim-enable}</b>					
<b>Remarks</b>	Indicates the photo color space ID of the frame data to be acquired by the Video Capture Device, if <b>CapPhotoColorSpace</b> property is true <b>and it is used <b>takePhoto</b> method</b> . Valid values are one of the values listed in the <b>CapPhotoColorSpaceList</b> property. <b>This property is referred to when <b>VideoCaptureMode</b> property value is VCAP_VCMODE_PHOTO and <b>CapPhoto</b> is true</b> . This property is initialized by the <b>open</b> method. <a href="#">Goto Table47</a>					
<b>Errors</b>	A <b>UposException</b> may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><b>Value</b></th><th><b>Meaning</b></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified.	
<b>Value</b>	<b>Meaning</b>					
E_ILLEGAL	An invalid value was specified.					
<b>See also</b>	<b>PhotoColorSpaceList</b> Property, <b>VideoCaptureMode</b> property, <b>CapPhoto</b> Property, <b>CapPhotoColorSpace</b> Property, <b>takePhoto</b> Method					

## 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.

[Goto Table48](#)

<b>Parameter</b>	<b>Description</b>
<i>Color space ID</i>	ID for identifying the color space of RGB, YUV 422, etc. <b>Then if RGB Depth was 16 bits, YUV422 Depth was 32 bits, And</b> they are indicating like "RGB:16, YUV422:32,....."
<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** **CapPhotoColorSpace Property**, **PhotoColorSpace Property**, **VideoCaptureMode Property**

### PhotoFrameRate Property

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

**Remarks** Indicates the frame rate of frame data **recorded to-be-acquired** by the Video Capture Device **and the photo image capturing and recorded with the takePhoto method. This property is only applied when VideoCaptureMode property is set to VCAP\_VCMODE\_PHOTO. Valid values range from 1 to PhotoMaxFrameRate property and CapPhoto property is true.** This property is initialized by the **open** method.

[Goto Table49](#)

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

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	An invalid value was specified.

**See also** **CapPhoto Property**, **CapPhotoFrameRate Property**, **PhotoMaxFrameRate Property**, **VideoCaptureMode Property**, **takePhoto Method**

### 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**

## UPOS Ver1.16 RCSD Specification

### PhotoResolution Property

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

**Remarks**      It shows the resolution of the frame data acquired by the Video Capture Device and the photo taken and recorded with the **takePhoto** method. Valid values are one of those listed in **PhotoResolutionList** property. [Goto Table50](#)  
**This property is only applied when VideoCaptureMode property is set to VCAP\_VCMODE\_PHOTO and if CapPhoto 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 *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

**See also**      **CapPhoto Property, PhotoResolutionList Property, VideoCaptureMode Property, takePhoto Method**

### PhotoResolutionList Property

**Syntax**      **PhotoResolutionList:** *string* {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. [Goto Table51](#)

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

**See also**      **CapPhotoResolution Property, PhotoResolution Property, VideoCaptureMode property**

### PhotoType Property

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

**Remarks**      Indicates the data format of photo taken with the **takePhoto** method. Valid values are one of the values listed in the **CapPhotoTypeList** 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: [Goto Table52](#)

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

**See also**      **CapPhoto Property, CapPhotoTypeList Property, takePhoto Method, VideoCaptureMode Property**

### PhotoTypeList Property

<b>Syntax</b>	<b>PhotoTypeList:</b> <i>string</i> {read-only, access after open}
<b>Remarks</b>	A comma-separated list of photo image format values that can be set for the <b>PhotoType</b> 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 <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>PhotoType</b> Property, <b>VideoCaptureMode</b> property

### RemainingRecordingTimeInSec Property

<b>Syntax</b>	<b>RemainingRceordingTimeInSec:</b> <i>int32</i> {read-only, access after open-claim-enable}
<b>Remarks</b>	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 <b>startVideo</b> returns, this property initially holds the time passed as argument <i>recordingTime</i> to that call. If this argument value is FOREVER (-1), this property also holds this value unchanged until <b>stopVideo</b> method has been called. This property is initialized during device set <b>DeviceEnabled</b> method to 0.
<b>Errors</b>	UposException may be thrown when this property is accessed. For further information, see “Errors” on page Intro-20.
<b>See Also</b>	<b>startVideo</b> Method, <b>stopVideo</b> Method

### Saturation Property

<b>Syntax</b>	<b>Saturation:</b> <i>int32</i> {read-write, access after open-claim-enable }				
<b>Remarks</b>	Indicate the saturation of camera. Valid values range from 0 to 100. This property is initialized by the <b>open</b> method.				
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are:				
	<table border="1"> <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.				
<b>See also</b>	<b>CapSaturation</b> Property				

[Goto Table53](#)

## 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 <b>CapAssociatedHardTotalsDevice</b> 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 <b>CapAssociatedHardTotalsDevice</b> 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. **Some possible values of the exception's *ErrorCode* property are:**

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

[Goto Table54](#)

- See Also**      **CapStorage** Property

## VerticalFlip Property

- Syntax**      **VerticalFlip: *boolean* {read-write, access after open-claim-enable}**
- 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. 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.

[Goto Table55](#)

- 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**      **CapVerticalFlip** Property, **HorizontalFlip** Property, **CapHorizontalFlip** Property



**UPOS Ver1.16 RCSD Specification**  
**VideoCaptureMode Property**

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

**Remarks**            Indicate the operation mode of video capture device.  
Valid values are as follows

**Parameter            Description**

---

**VCAP\_VCMODE\_PHOTO**

This mode is for taking photo. and their data recording. Can be set when **CapPhoto** property is true. 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.

**VCAP\_VCMODE\_VIDEO**

This mode is for taking the videos and their data recording. Can be set when **CapVideo** property is true. 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 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.

[Goto Table56](#)

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

**Value                    Meaning**

---

**E\_ILLEGAL**    An invalid value was specified.

**See also**            **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.

## UPOS Ver1.16 RCSD Specification

### VideoColorSpace Property

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

**Remarks** Indicates the video color space ID of the frame data to be **acquired by the Video Capture Device, if CapVideoColorSpace property is true and it is 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.

[Goto Table57](#)

**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** **CapVideoColorSpace Property, VideoColorSpaceList Property, VideoCaptureMode Property, startVideo Method**

### VideoColorSpaceList Property

**Syntax** VideoColorSpaceList: *string* {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.

[Goto Table58](#)

<u>Parameter</u>	<u>Description</u>
<i>Color space ID</i>	ID for identifying the color space of RGB, YUV422, etc. <b>Then if RGB Depth was 16 bits, YUV422 Depth was 32 bits, And they are indicating like“RGB:16, †YUV422:32, †...”</b>
<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, VideoCaptureMode Property, VideoColorSpace Property**

### VideoFrameRate Property

**Syntax** VideoFrameRate; *int32* {read-write, access after open-claim-enable}

**Remarks** Indicates the frame rate of the frame data recorded by the Video Capture Device and the video image capturing and recorded with the **startVideo** method. This property is only applied when VCAP\_VCMODE\_VIDEO is set in **VideoCaptureMode** property. Valid values range from 1 to **VideoMaxFrameRate** property and **CapVideo** property is true. This property is initialized by the **open** method.

[Goto Table59](#)

**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** **CapVideo Property, CapVideoFrameRate, Property, VideoCaptureMode Property, VideoMaxFrameRate Property, startVideo Method**

**UPOS Ver1.16 RCSD Specification**  
**VideoMaxFrameRate Property**

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

[Goto Table60](#)

**VideoResolution Property**

<b>Syntax</b>	<b>VideoResolution: <i>string</i> {read-write, access after open-claim-enable}</b>
<b>Remarks</b>	Indicates the resolution of video image data acquired by the Video Capture Device and recorded with the execution of <b>startVideo</b> method. Valid values are one of the values listed in the <b>VideoResolutionList</b> property. This property is only applied when <b>VCAP_VCMODE_VIDEO</b> is set in <b>VideoCaptureMode</b> property and if <b>CapVideo</b> property is true. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are:

[Goto Table61](#)

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

<b>See also</b>	<b>VideoResolutionList</b> Property, <b>CapVideo</b> Property, <b>VideoCaptureMode</b> Property, <b>startVideo</b> Method
-----------------	---

**VideoResolutionList Property**

<b>Syntax</b>	<b>VideoResolutionList: <i>string</i> {read-only, access after open}</b>
<b>Remarks</b>	A comma-separated list of possible resolutions for the <b>VideoResolution</b> 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 <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<b>CapVideoResolution</b> Property, <b>VideoResolution</b> Property

**VideoType Property**

<b>Syntax</b>	<b>VideoType; <i>string</i> {read-write, access after open-claim-enable}</b>
<b>Remarks</b>	Indicate the shape of the taking video and recorded with the <b>startVideo</b> method. Valid values are one of those listed in <b>VideoTypeList</b> property. This property is applied when <b>VCAP_VCMODE_VIDEO</b> is set in <b>VideoCaptureMode</b> property and if <b>CapVideo</b> property is true. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are:

[Goto Table62](#)

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

<b>See also</b>	<b>VideoCaptureMode</b> Property, <b>CapVideo</b> Property, <b>VideoTypeList</b> Property, <b>startVideo</b> Method
-----------------	---

## VideoTypeList Property

[Goto Table63](#)

<b>Syntax</b>	<b>VideoTypeList:</b> <i>string</i> {read-only, access after open}
<b>Remarks</b>	A comma-separated list of <b>video</b> image format values that can be set for the <b>VideoType</b> property. * <sup>1</sup> 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 <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See also</b>	<a href="#">CapVideoType Property</a> , <b>VideoType</b> Property, <b>VideoCaptureMode</b> Property

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](https://en.wikipedia.org/wiki/Digital_container_format)

MJPEG : Motion JPEG :

[https://en.wikipedia.org/wiki/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**

## Methods (UML operations)

### startVideo Method

**Syntax**

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

Parameter	Description
filename	Specify the name of the 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 <b>stopVideo</b> method is called.

[Goto Table64](#)

**Remarks**

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. **During the video image capturing, the recorded data storing is processed asynchronously and when the start and stop states are changed, StatusUpdateEvents are delivered to the application. 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 **stopVideo** method, recording is completed and the video file specified by fileName is recorded and can deliver to the application. Also, S\_BUSY is set in the **Status** property during video capturing and recording. The place where video files are recorded is 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. <b>VideoCaptureMode</b> property is not VCAP_VCMODE_VIDEO and <b>CapVideo</b> is not true.
E_EXISTS	fileName already exists. (If overwrite is false)
E_BUSY	Cannot execute because it is recording.

**See also**

**VideoColorSpace** Property, **VideoResolution** Property, **VideoFrameRate** Property, **VideoType** Property, **stopVideo** Method, **StatusUpdateEvent** Event, **VideoCaptureMode** Property

## UPOS Ver1.16 RCSD Specification

### stopVideo Method

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

**Remarks**            The video capturing and recording process started by the **startVideo** method has been ended and the taking video is completed. This method processed synchronously. **StatusUpdateEvent** ~~will notify~~ **is delivered to notify** the application that ~~there is a change in the power status or a state change during taking video and recording~~ **the device video capturing and recording were stopped.**

[Goto Table65](#)

**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:

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

**See also**            **startVideo** Method, **StatusUpdateEvent** Event

**UPOS Ver1.16 RCSD Specification**  
**takePhoto Method**

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

<b>Parameter</b>	<b>Description</b>
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 <b>ErrorEvent</b> is sent to the application. If FOREVER (-1) the service will wait until a photograph is taken or an application error occurs.

[Goto Table66](#)

**Remarks**            Take photo and record with setting contents of **PhotoColorSpace** property, **PhotoResolution** property, **PhotoFrameRate** Property and **PhotoType** property. Before calling this method, it needs to set the **VideoCaptureMode** property to VCAP\_VCMODE\_PHOTO and this method can be executed if **CapPhoto** property is true. This method is executed synchronously. **The process of recorded data storing is performed asynchronously. StatusUpdateEvents are delivered to the application when the start and the end states were changed.** The location where photo files are recorded is 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:

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

**See also**            **VideoCaptureMode** Property, **PhotoColorSpace** Property, **PhotoResolution** Property, **CapPhoto** Property, **PhotoType** Property, **PhotoFrameRate** Property, **StatusUpdateEvent** Event

## 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 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:

<b>Attribute</b>	<b>Type</b>	<b>Description</b>
<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



## ErrorEvent

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

[Goto Table67](#)

**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_INOUTPUT is specified. An error occurred during asynchronous <del>process</del> <b>action</b> .
<i>ErrorResponse</i>	<i>int32</i>	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_INOUTPUT	Error occurred while processing asynchronous <del>output</del> <b>input</b> .

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

Value	Meaning
ER_RETRY	Retry sending the <b>recorded data or storing it</b> . The error state is exited. Typically, valid for asynchronous <b>recorded data storing output devices</b> when the locus is EL_INOUTPUT, <del>in</del> which case the asynchronous <b>recorded data storing output</b> is retried, and the error state is exited. This is the default response <del>when the locus is EL_OUTPUT</del> .
ER_CLEAR	<del>Valid for loci: EL_OUTPUT</del> Clear all buffered <b>captured input or stored output data (including all asynchronous output)</b> . 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~~ **Device information Reporting Model** " on Page Intro-30

**UPOS Ver1.16 RCSD Specification**  
**StatusUpdateEvent**

**<< event >>** **upos::events::StatusUpdateEvent**  
**Status** : *int32* {read-only}  
**Description** *Notifies the application that there is a change in the power status or a state change of the Video Capture device.*

**Attributes** This event contains the following attribute:

<b>Attributes</b>	<b>Type</b>	<b>Description</b>
<i>Status</i>	<i>int32</i>	Indicates a change in the power status or a state change 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.

<b>Value</b>	<b>Meaning</b>
--------------	----------------

VCAP_SUE_START_VIDEO_ <del>RECORDING</del>	It will be notified when video recording starts.
VCAP_SUE_STOP_VIDEO_ <del>RECORDING</del>	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.

[Goto Table147](#)

**Remarks** Enqueued when the Video Capture Device detects a power state change or a status change.

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

# 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>
<b>AutoDisable:</b>	<i>boolean</i>	{read-write}	1.16	Open
<b>CapCompareFirmwareVersion:</b>	<i>boolean</i>	{read-only}	1.16	Open
<b>CapPowerReporting:</b>	<i>int32</i>	{read-only}	1.16	open
<b>CapStatisticsReporting:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateFirmware:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateStatistics:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CheckHealthText:</b>	<i>string</i>	{read-only}	1.16	open
<b>Claimed:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>DataCount:</b>	<i>int32</i>	{read-only}	1.16	open
<b>DataEventEnabled:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>DeviceEnabled:</b>	<i>boolean</i>	{read-write}	1.16	open & claim
<b>FreezeEvents:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>OutputID:</b>	<i>int32</i>	{read-only}	1.16	<i>Not supported</i>
<b>PowerNotify:</b>	<i>int32</i>	{read-write}	1.16	open
<b>PowerState:</b>	<i>int32</i>	{read-only}	1.16	open
<b>State:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceControlDescription:</b>	<i>string</i>	{read-only}	1.16	--
<b>DeviceControlVersion:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceServiceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>DeviceServiceVersion:</b>	<i>int32</i>	{read-only}	1.16	open
<b>PhysicalDeviceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>PhysicalDeviceName:</b>	<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>
<b>CapIndividualList:</b>	<i>string</i>	{read-only}	1.16	open
<b>IndividualIDs:</b>	<i>string</i>	{read-only}	1.16	open, claim & enable
<b>IndividualRecognitionFilter:</b>	<i>string</i>	{read-write}	1.16	open
<b>IndividualRecognitionInformation</b>	<i>string</i>	{read-only}	1.16	open

### Methods (UML operations)

#### Common

<i>Name</i>	<i>Version</i>	
<b>open (logicalDeviceName: <i>string</i>):</b> void {raises-exception}	1.16	
<b>close ():</b> void {raises-exception, use after open}	1.16	
<b>claim (timeout: <i>int32</i>):</b> void {raises-exception, use after open}	1.16	
<b>release ():</b> void {raises-exception, use after open, claim}	1.16	
<b>checkHealth (level: <i>int32</i>):</b> void {raises-exception, use after open, enable}	1.16	
<b>clearInput ():</b> void {raises-exception, use after open, claim}	1.16	<a href="#">Goto Table68</a>
<b>clearInputProperties ():</b> void {raises-exception, use after open, claim}	1.16	<a href="#">Goto Table69</a>
<b>clearOutput ():</b> void { }		<i>Not supported</i>
<b>compareFirmwareVersion (firmwareFileName: <i>string</i>, out result: <i>int32</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table70</a>
<b>directIO (command: <i>int32</i>, inout data: <i>int32</i>, inout obj: <i>object</i>):</b> void {raises-exception, use after open}	1.16	
<b>resetStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table71</a>
<b>retrieveStatistics (inout statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table72</a>
<b>updateFirmware (firmwareFileName: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table73</a>
<b>updateStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table74</a>

## UPOS Ver1.16 RCSD Specification

### Events (UML interfaces)

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

## 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.

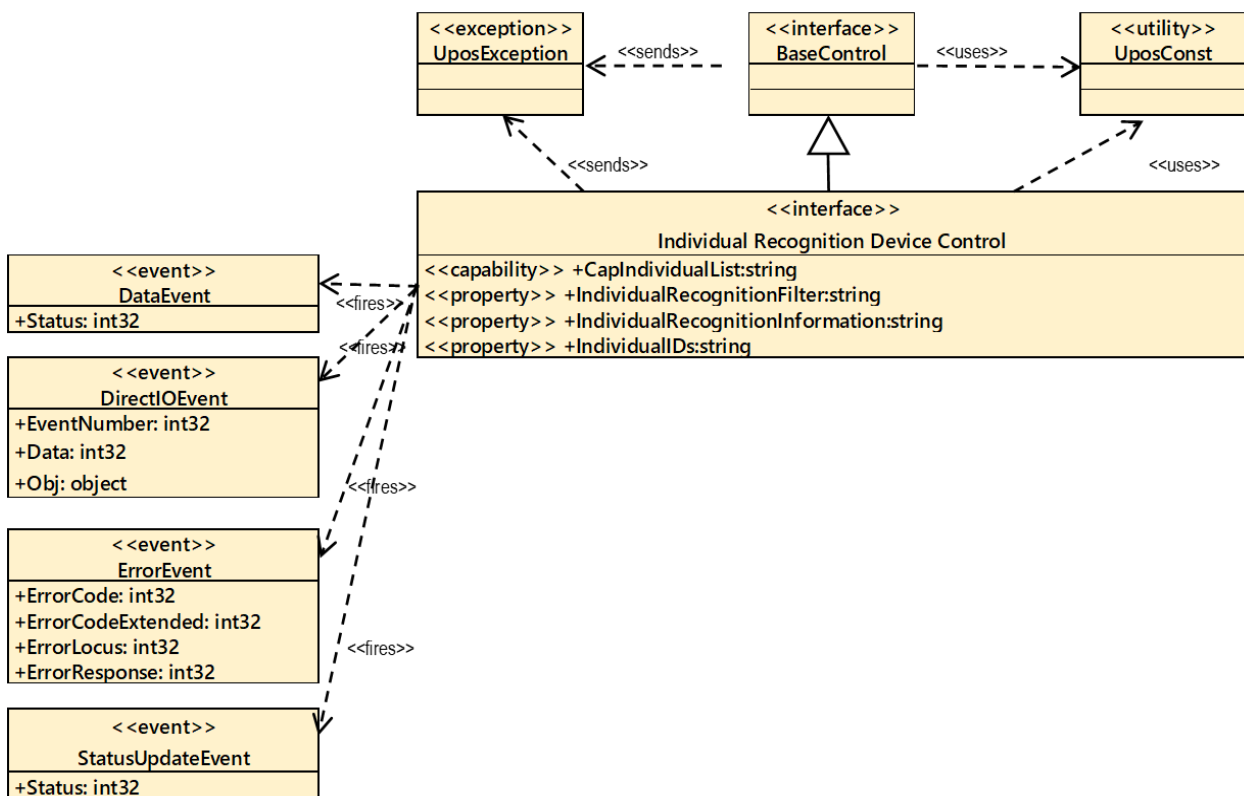


Fig. Chap.40-1 Individual Recognition Class Diagram

## Model

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

### 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.
- 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

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.



## 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]	-	-	

## 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.

**UPOS Ver1.16 RCSD Specification**

					Event is fired when it matches the image 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.	

## IndividualRecognition Information 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]	-	-	

## 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

## UPOS Ver1.16 RCSD Specification

# Properties (UML attributes)

### CapIndividualList Property

**Syntax** CapIndividualList: *string* {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 (":").

<u>Parameter</u>	<u>Meaning</u>
IndividualID	An ID indicated an identifiable Individual
IndividualName	A Name of an Individual.

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** IndividualIDs Property

### IndividualIDs Property

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

**Remarks** Set the IndividualIDs recognizable Individual recognition device.

IndividualIDs values are 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

### IndividualRecognitionFilter Property

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

**Remarks** Individual Recognition Function Information:

- Supporting the various functions (Refer to the Individual Recognition Filter Example Format written by JSON and supported function examples .).
- Various Valid / Invalid State functions.
- Various handled function types. (e.g., "male" "female" in gender recognition, etc.).
- Various filter function settings. All Individual Recognition function data information is defined by the device. By referring to these contents, the application can determine the supporting scope. 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. **Some possible values of the exception's ErrorCode property are:**

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

[Goto Table75](#)

**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.

## 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:

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

**Remarks** Before this event is delivered, the **data is copied into corresponding properties 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.**

[Goto Table148](#)

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

### 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:

<b>Attribute</b>	<b>Type</b>	<b>Description</b>
<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.



## 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:

<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. See values below. <del>EL_OUTPUT is specified. An error occurred during asynchronous action.</del>
<i>ErrorResponse</i>	<i>int32</i>	Error Response, whose default value may be overridden by the application. (i.e., this attribute is settable). See <i>ErrorResponse</i> below for values.

[Goto Table76](#)

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

<u>Value</u>	<u>Meaning</u>
<del>EL_OUTPUT</del>	<del>Error occurred while processing asynchronous output.</del>
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:

<u>Value</u>	<u>Meaning</u>
<del>ER_RETRY</del>	<del>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.</del>
ER_CLEAR	Valid for all locus: 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~~

## 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:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
------------------	-------------	--------------------

<i>Status</i>	<i>int32</i>	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

# 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>
<b>AutoDisable:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>CapCompareFirmwareVersion:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPowerReporting:</b>	<i>int32</i>	{read-only}	1.16	open
<b>CapStatisticsReporting:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateFirmware:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateStatistics:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CheckHealthText:</b>	<i>string</i>	{read-only}	1.16	open
<b>Claimed:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>DataCount:</b>	<i>int32</i>	{read-only}	1.16	open
<b>DataEventEnabled:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>DeviceEnabled:</b>	<i>boolean</i>	{read-write}	1.16	open & claim
<b>FreezeEvents:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>OutputID:</b>	<i>int32</i>	{read-only}	1.16	<i>Not supported</i>
<b>PowerNotify:</b>	<i>int32</i>	{read-write}	1.16	open
<b>PowerState:</b>	<i>int32</i>	{read-only}	1.16	open
<b>State:</b>	<i>int32</i>	{read-only}	1.16	open
<b>DeviceControlDescription:</b>	<i>string</i>	{read-only}	1.16	--
<b>DeviceControlVersion:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceServiceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>DeviceServiceVersion:</b>	<i>int32</i>	{read-only}	1.16	open
<b>PhysicalDeviceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>PhysicalDeviceName:</b>	<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>
<b>CapAssociatedHardTotalsDevice:</b>	<i>string</i>	{read-only}	1.16	open
<b>CapChannel:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapRecordingLevel:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapSamplingRate:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapSoundType:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapStorage</b>	<i>int32</i>	{read-only}	1.16	open
<b>Channel:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>ChannelList:</b>	<i>string</i>	{read-only}	1.16	open
<b>RecordingLevel:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>RemainingRecordingTimeInSec:</b>	<i>int32</i>	{read-only}	1.16	open, claim & enable
<b>SamplingRate:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>SamplingRateList:</b>	<i>string</i>	{read-only}	1.16	open
<b>SoundData:</b>	<i>binary</i>	{read-only}	1.16	open
<b>SoundType:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>SoundTypeList:</b>	<i>string</i>	{read-only}	1.16	open
<b>Storage</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable

### Methods(UML operations)

#### Common

<i>Name</i>	<i>Version</i>
<b>open (logicalDeviceName: <i>string</i>):</b> void {raises-exception}	1.16
<b>close ():</b> void {raises-exception, use after open}	1.16
<b>claim (timeout: <i>int32</i>):</b> void {raises-exception, use after open}	1.16
<b>release ():</b> void {raises-exception, use after open, claim}	1.16
<b>checkHealth (level: <i>int32</i>):</b> void {raises-exception, use after open, enable}	1.16
<b>clearInput ():</b> void {raises-exception, use after open, claim}	1.16

[Goto Table77](#)

## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)(continued)

#### Common

<i>Name</i>	<i>Version</i>	
<b>clearInputProperties ():</b> void {raises-exception, use after open, claim}	1.16	<a href="#">Goto Table78</a>
<b>clearOutput ():</b> void { }	<i>Not supported</i>	
<b>compareFirmwareVersion (firmwareFileName: string, out result: int32):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table79</a>
<b>directIO (command: int32, inout data: int32, inout obj: object):</b> void {raises-exception, use after open}	1.16	
<b>resetStatistics (statisticsBuffer: string):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table80</a>
<b>retrieveStatistics (inout statisticsBuffer: string):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table81</a>
<b>updateFirmware (firmwareFileName: string):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table82</a>
<b>updateStatistics (statisticsBuffer: string):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table83</a>

#### Specific

<i>Name</i>	<i>Version</i>	
<b>startRecording (FileName: string, OverWrite: boolean, RecordingTime:int32):</b> void {raises-exception, use after open, claim, enable}	1.16	
<b>stopRecording ():</b> 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>
<b>upos::events::DataEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::DirectIOEvent</b>			1.16
<b>EventNumber:</b>	<i>int32</i>	{read-only}	
<b>Data:</b>	<i>int32</i>	{read-write}	
<b>Obj:</b>	<i>object</i>	{read-write}	
<b>upos::events::ErrorEvent</b>			1.16
<b>ErrorCode:</b>	<i>int32</i>	{read-only}	
<b>ErrorCodeExtended:</b>	<i>int32</i>	{read-only}	
<b>ErrorLocus:</b>	<i>int32</i>	{read-only}	
<b>*pErrorResponse:</b>	<i>int32</i>	{read-write}	
<b>upos::events::OutputCompleteEvent</b>		<i>Not supported</i>	1.16
<b>upos::events::StatusUpdateEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	1.16

## General Information

The Sound Recorder programmatic name is "Sound Recorder".

### Capabilities

The Sound Recorder has the following capability:

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.

### Sound Recorder Class Diagram

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

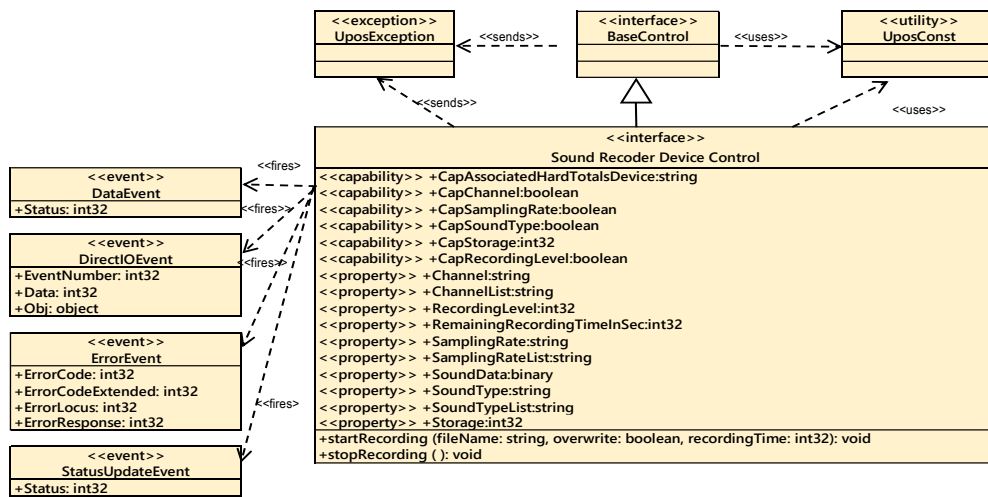


Fig. Chap. 41-1 Sound Recorder Class Diagram

## Model

Sound Recorder Control follows a general “Device Input Model” 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.

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 status **SERC\_SUE\_STOP\_SOUND\_RECORDING** is evoked to notify the application that recording has been stopped.

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.



## UPOS Ver1.16 RCSD Specification

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 **Error Event** can be set back to their default values by calling the **clearInputProperties** 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.

## 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.

## Properties(UML attributes)

### CapAssociatedHardTotalsDevice Property

<b>Syntax</b>	<b>CapAssociatedHardTotalsDevice:</b> <i>string</i> { <b>read-only</b> <del>write</del> , access after open}
<b>Remarks</b>	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 <b>CapStorage</b> is either SREC_CST_ALL or SREC_CST_HARDTOTALS_ONLY. If <b>CapStorage</b> is SREC_CST_HOST_ONLY this property value must be the empty string. <b>This property is initialized by the open method.</b>
<b>Errors</b>	UposException may be thrown when this property is accessed. <a href="#">Goto Table84</a> For further information, see "Errors" on page Intro-20.
<b>See Also</b>	<b>CapStorage</b> Property

### CapChannel Property

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

### CapSamplingRate Property

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

### CapSoundType Property

<b>Syntax</b>	<b>CapSoundType:</b> <i>boolean</i> { <b>read-only</b> , access after open}
<b>Remarks</b>	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 <b>open</b> method.
<b>Errors</b>	UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
<b>See Also</b>	<b>SoundType</b> Property.

### 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.

<b>Value</b>	<b>Meaning</b>
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

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

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	An invalid value was specified.

[Goto Table85](#)

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

## 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.

### 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.
- | <u>Value</u> | <u>Meaning</u>                  |
|--------------|---------------------------------|
| 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

### 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.
- See Also** CapSamplingRate Property, **SamplingRateList** Property **Some possible values of the exception's *ErrorCode* property are:**

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

[Goto Table86](#)

## 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.

### 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**.

### SoundType Property

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

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

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

[Goto Table87](#)

**See Also**      **CapSoundType** Property, **CapSoundTypeList** Property.

### 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.

**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.

<b>Value</b>	<b>Meaning</b>
SREC_ST_HARDTOTALS	The encoded data file is written to the associated Hard Totals device. The property <b>CapAssociatedHardTotalsDevice</b> 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 <b>CapAssociatedHardTotalsDevice</b> 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.

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

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



## UPOS Ver1.16 RCSD Specification

### stopRecording Method

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

**Remarks**      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:

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

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



## 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:

Attribute	Type	Description
Status	int32	Set to 0.

**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

### 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:

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.

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}
```

[Goto Table88](#)

**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.

<b>Attributes</b>	<b>Type</b>	<b>Description</b>
<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:*

<b>Value</b>	<b>Meaning</b>
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.*

<b>Value</b>	<b>Meaning</b>
ESREC_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:*

<b>Value</b>	<b>Meaning</b>
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 <b>DataEvents</b> as directed by the <b>DataEventEnabled</b> property. When all input has been delivered and <b>DataEventEnabled</b> is again set to true, then another <b>ErrorEvent</b> 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](#), “**Device Input Model**” on page Intro-22, “**Error Handling**” on page Intro-23

**UPOS Ver1.16 RCSD Specification**  
**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 Sound Recorder device.*

**Attributes**      This event contains the following attribute:

<b>Attributes</b>	<b>Type</b>	<b>Description</b>
<i>Status</i>	<i>int32</i>	Indicates a change in the power status or a 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.

<b>Value</b>	<b>Meaning</b>
--------------	----------------

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.

# 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>
<b>AutoDisable:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>CapCompareFirmwareVersion:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPowerReporting:</b>	<i>int32</i>	{read-only}	1.16	open
<b>CapStatisticsReporting:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateFirmware:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateStatistics:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CheckHealthText:</b>	<i>string</i>	{read-only}	1.16	open
<b>Claimed:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>DataCount:</b>	<i>int32</i>	{read-only}	1.16	open
<b>DataEventEnabled:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>DeviceEnabled:</b>	<i>boolean</i>	{read-write}	1.16	open & claim
<b>FreezeEvents:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>OutputID:</b>	<i>int32</i>	{read-only}	1.16	<i>Not supported</i>
<b>PowerNotify:</b>	<i>int32</i>	{read-write}	1.16	open
<b>PowerState:</b>	<i>int32</i>	{read-only}	1.16	open
<b>State:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceControlDescription:</b>	<i>string</i>	{read-only}	1.16	--
<b>DeviceControlVersion:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceServiceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>DeviceServiceVersion:</b>	<i>int32</i>	{read-only}	1.16	open
<b>PhysicalDeviceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>PhysicalDeviceName:</b>	<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>
<b>CapLanguage:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>HearingDataPattern:</b>	<i>string</i>	{read-only}	1.16	open, claim & enable
<b>HearingDataWord:</b>	<i>string</i>	{read-only}	1.16	open, claim & enable
<b>HearingDataWordList:</b>	<i>string</i>	{read-only}	1.16	open, claim & enable
<b>HearingResult:</b>	<i>int32</i>	{read-only}	1.16	open, claim & enable
<b>HearingStatus:</b>	<i>int32</i>	{read-only}	1.16	open, claim & enable
<b>LanguageList:</b>	<i>string</i>	{read-only}	1.16	open

### Methods (UML operations)

#### Common

<i>Name</i>	<i>Version</i>	
<b>open (logicalDeviceName: <i>string</i>):</b> void {raises-exception}	1.16	
<b>close ():</b> void {raises-exception, use after open}	1.16	
<b>claim ( timeout: <i>int32</i> ):</b> void {raises-exception, use after open}	1.16	
<b>release ():</b> void {raises-exception, use after open, claim}	1.16	
<b>checkHealth ( level: <i>int32</i> ):</b> void {raises-exception, use after open, <b>claim</b> , <b>enable</b> }	1.16	<a href="#">Goto Table89</a>
<b>clearInput ():</b> void { <b>raises-exception, use after open, claim</b> }	1.16	<a href="#">Goto Table90</a>
<b>clearInputProperties ():</b> void { <b>raises-exception, use after open, claim</b> }	1.16	<a href="#">Goto Table91</a>
<b>clearOutput ():</b> void { }	<i>Not supported</i>	
<b>compareFirmwareVersion (firmwareFileName: <i>string</i>, out result: <i>int32</i>):</b> void {raises-exception, use after open, claim, enable}	1.16	
<b>directIO (command: <i>int32</i>, inout data: <i>int32</i>, inout obj: <i>object</i>):</b> void {raises-exception, use after open}	1.16	
<b>resetStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16	
<b>retrieveStatistics ( inout statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16	

## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)(continued)

#### Common

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

#### Specific

<i>Name</i>	<i>Version</i>
<b>startHearingFree (language: string):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>startHearingSentence (language: string, wordList: string, patternList: string):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>startHearingWord (language: string, wordList: string):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>startHearingYesNo (language: string):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>stopHearing ( ):</b> 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>
<b>upos::events::DataEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::DirectIOEvent</b>			1.16
<b>EventNumber:</b>	<i>int32</i>	{read-only}	
<b>Data:</b>	<i>int32</i>	{read-write}	
<b>Obj:</b>	<i>object</i>	{read-write}	
<b>upos::events::ErrorEvent</b>			1.16
<b>ErrorCode:</b>	<i>int32</i>	{read-only}	
<b>ErrorCodeExtended:</b>	<i>int32</i>	{read-only}	
<b>ErrorLocus:</b>	<i>int32</i>	{read-only}	
<b>ErrorResponse:</b>	<i>int32</i>	{read-write}	
<b>upos::events::OutputCompleteEvent</b>		<i>Not supported</i>	
<b>upos::events::StatusUpdateEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	1.16

## 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.

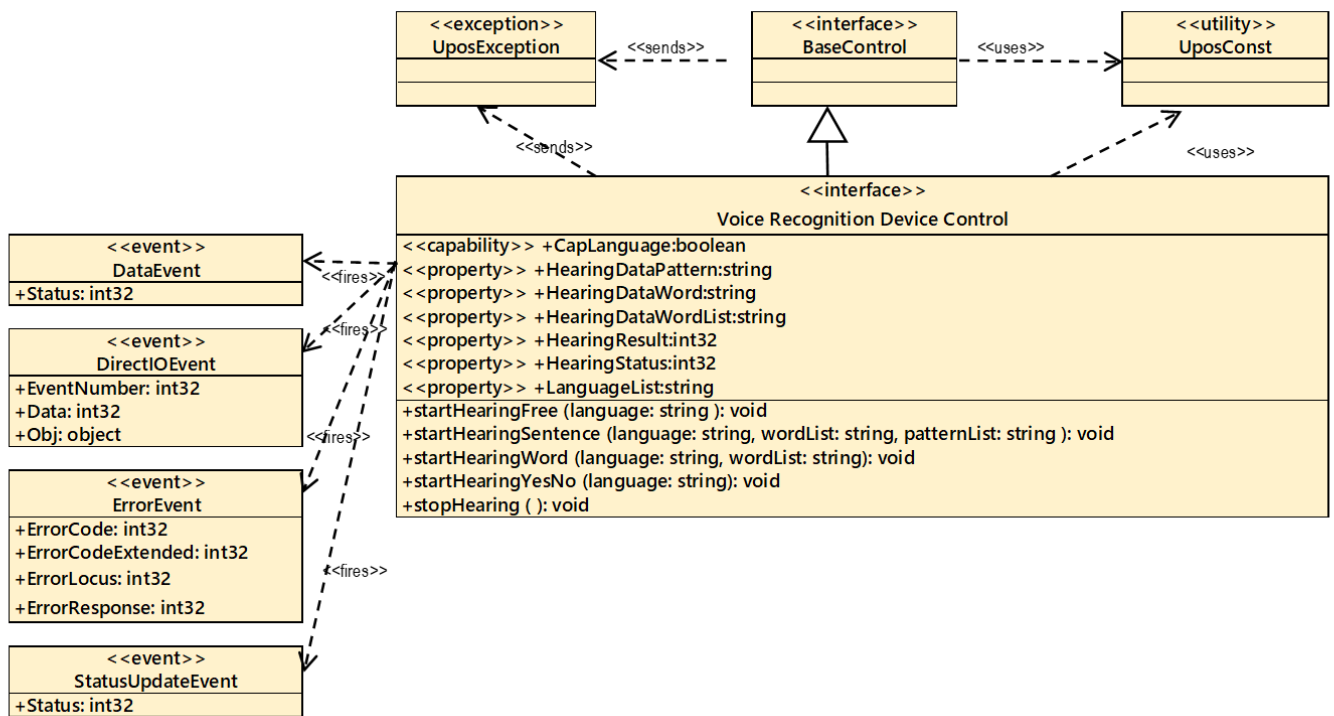


Fig. Chap. 42-1 Voice Recognition Class Diagram



## 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.

## 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.

## 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.

## Properties (UML attributes)

### CapLanguage Property

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

### HearingDataPattern Property

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

### HearingDataWord Property

<b>Syntax</b>	<b>HearingDataWord:</b> <i>string</i> {read-only, access after open-claim-enable}								
<b>Remarks</b>	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 <b>HearingResult</b> property.								
	<table border="1"> <thead> <tr> <th><b>Methods</b></th> <th><b>Meaning</b></th> </tr> </thead> <tbody> <tr> <td><b>startHearingYesNo</b> Method</td> <td>The recognized word is set.</td> </tr> <tr> <td><b>startHearingWord</b> Method</td> <td>Recognized words are set among the word candidates specified by the <b>startHearingWord</b> method.</td> </tr> <tr> <td><b>startHearingFree</b> Method</td> <td>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.</td> </tr> </tbody> </table>	<b>Methods</b>	<b>Meaning</b>	<b>startHearingYesNo</b> Method	The recognized word is set.	<b>startHearingWord</b> Method	Recognized words are set among the word candidates specified by the <b>startHearingWord</b> method.	<b>startHearingFree</b> 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.
<b>Methods</b>	<b>Meaning</b>								
<b>startHearingYesNo</b> Method	The recognized word is set.								
<b>startHearingWord</b> Method	Recognized words are set among the word candidates specified by the <b>startHearingWord</b> method.								
<b>startHearingFree</b> 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 <b>DataEvent</b> is enqueued.								
<b>Errors</b>	A <b>UposException</b> may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.								
<b>See Also</b>	<b>HearingResult</b> Property, <b>startHearingYesNo</b> Method, <b>startHearingWord</b> Method, <b>startHearingFree</b> Method								

## 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 (":").

<b>Parameter</b>	<b>Description</b>
<i>WordGroupID</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, count:a:two:three"

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.

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

**See Also** **startHearingSentence** Method

## 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.

<b>Value</b>	<b>Meaning</b>
VRCG_HRESULT_YESNO_YES	Voice recognition result of <b>StartHearingYesNo</b> methods. Also, Device got an answer that is classified as YES. The recognized content is set in the <b>HearingDataWord</b> property.
VRCG_HRESULT_YESNO_NO	Voice recognition result of <b>startHearingYesNo</b> method. Also, Device got an answer that is classified as NO. The recognition content is set in the <b>HearingDataWord</b> property.
VRCG_HRESULT_YESNO_CANCEL	Voice recognition result of <b>startHearingYesNo</b> method. Also, Device got responses that are classified as CANCEL. The recognition content is set in the <b>HearingDataWord</b> property.
VRCG_HRESULT_WORD	Recognition result of <b>startHearingWord</b> method. The recognition content is set in the <b>HearingDataWord</b> property.
VRCG_HRESULT_SENTENCE	Recognition result of <b>startHearingSentence</b> method. The recognition content is set in the <b>HearingDataWordList</b> property and <b>HearingDataPattern</b> property.
VRCG_HRESULT_FREE	Recognition result of <b>startHearingFree</b> method. The recognition content is set in the <b>HearingDataWord</b> property.

This property is set by the device control just before the **DataEvent** is 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.

## 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.

<u>Value</u>	<u>Meaning</u>
VRCG_HSTATUS_NONE	Voice recognition is not running.
VRCG_HSTATUS_YESNO	Voice recognition by the <b>startHearingYesNo</b> method is in progress.
VRCG_HSTATUS_WORD	Voice recognition by the <b>startHearingWord</b> method is in progress.
VRCG_HSTATUS_SENTENCE	Voice recognition by the <b>startHearingSentence</b> method is in progress.
VRCG_HSTATUS_FREE	Voice recognition by the <b>startHearingFree</b> method is in progress.

This property is initialized by the **open** method. Also, it is set by the device control just before the voice recognition state changes.

**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

### 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.

## UPOS Ver1.16 RCSD Specification Methods (UML operations)

### startHearingFree Method

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

<u>Parameter</u>	<u>Description</u>
<i>Language</i>	Specify the language to recognize. Specify one of the values listed in the <b>LanguageList</b> 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. This method is executed asynchronously. 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:

<u>Value</u>	<u>Meaning</u>
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.

**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 <b>LanguageList</b> 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 <i>wordList</i> to the candidate, enclose the word group ID with "[" and "]". Example: "[word group ID1]" [word group ID2]"

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=VR CG_HSTATUS_SENTENCE;
```

People talk to "Give me two cups of coffee"

Speech recognition is performed, properties are set, and an event is notified.

```
HearingResult=VR CG_HRESULT_SENTENCE;
```

```
HearingDataPattern="P1";
```

```
HearingDataWordList="item:coffee,count:two";
```

```
raise DataEvent(0);
```

**Remarks** 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**



## UPOS Ver1.16 RCSD Specification

issuing. This method is executed asynchronously. 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:

<b>Value</b>	<b>Meaning</b>
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

## startHearingWord Method

**Syntax** **startHearingWord** (**language**: *string*, **wordList**: *string*):  
void {raises-exception, use after open-claim-enable}

<b>Parameter</b>	<b>Description</b>
<i>language</i>	Specify the language to recognize. Specify one of the values listed in the <b>LanguageList</b> property.
<i>wordList</i>	Specify word candidates to be waited on in a comma-separated list. Example: "word1, word2, word3"

**Remarks** 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.  
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:

<b>Value</b>	<b>Meaning</b>
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.

### startHearingYesNo Method

**Syntax** `startHearingYesNo (language: string):  
void {raises-exception, use after open-claim-enable}`

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

<i>language</i>	Specify the language to recognize. Specify one of the values listed in the <b>LanguageList</b> property.
-----------------	--

**Remarks** 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. 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:

<u>Value</u>	<u>Meaning</u>
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.

### stopHearing Method

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

**Remarks** Voice Recognition ends when this property called.

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:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified. Or an unsupported language was specified.

## 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:

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

**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** Method, **StartHearingWord** Method, **StartHearingSentence** Method, **StartHearingFree** Method, **directIO** Method.

### 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:

<b>Attribute</b>	<b>Type</b>	<b>Description</b>
<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.

## 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:

<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. See values below. <del>EL_OUTPUT is specified. An error occurred during asynchronous action.</del>
<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.

[Goto Table92](#)

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

<u>Value</u>	<u>Meaning</u>
<del>EL_OUTPUT</del>	<del>Error occurred while processing asynchronous output.</del>
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:

<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 or EL_INPUT_DATA, in which case the input is re-tried, and the error state is exited. <del>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.</del>
ER_CLEAR	Valid for all locus: 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~~

## 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:

<b>Attributes</b>	<b>Type</b>	<b>Description</b>
-------------------	-------------	--------------------

<i>Status</i>	<i>int32</i>	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.

<b>Value</b>	<b>Meaning</b>
--------------	----------------

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.

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

[Goto Table 150](#)

### 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, claim, enable}	1.16
clearInput (): void {raises-exception, use after open, claim }	1.16
clearInputProperties (): void {raises-exception, use after open, claim }	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 Table93](#)

[Goto Table94](#)

[Goto Table95](#)



## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)(continued)

#### Common

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

#### Specific

<i>Name</i>	<i>Version</i>
<b>playSound</b> (fileName: <i>string</i> , loop: <i>boolean</i> ): void {raises-exception, use after open, claim, enable}	1.16
<b>stopSound</b> (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>
<b>upos::events::DataEvent</b>		<i>Not supported</i>	1.16
<b>upos::events::DirectIOEvent</b>			1.16
<b>EventNumber:</b>	<i>int32</i>	{read-only}	
<b>Data:</b>	<i>int32</i>	{read-write}	
<b>Obj:</b>	<i>object</i>	{read-write}	
<b>upos::events::ErrorEvent</b>			1.16
<b>ErrorCode:</b>	<i>int32</i>	{read-only}	
<b>ErrorCodeExtended:</b>	<i>int32</i>	{read-only}	
<b>ErrorLocus:</b>	<i>int32</i>	{read-only}	
<b>ErrorResponse:</b>	<i>int32</i>	{read-write}	
<b>upos::events::OutputCompleteEvent</b>			1.16
<b>OutputID:</b>	<i>int32</i>	{read-only}	
<b>upos::events::StatusUpdateEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	1.16

## 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.

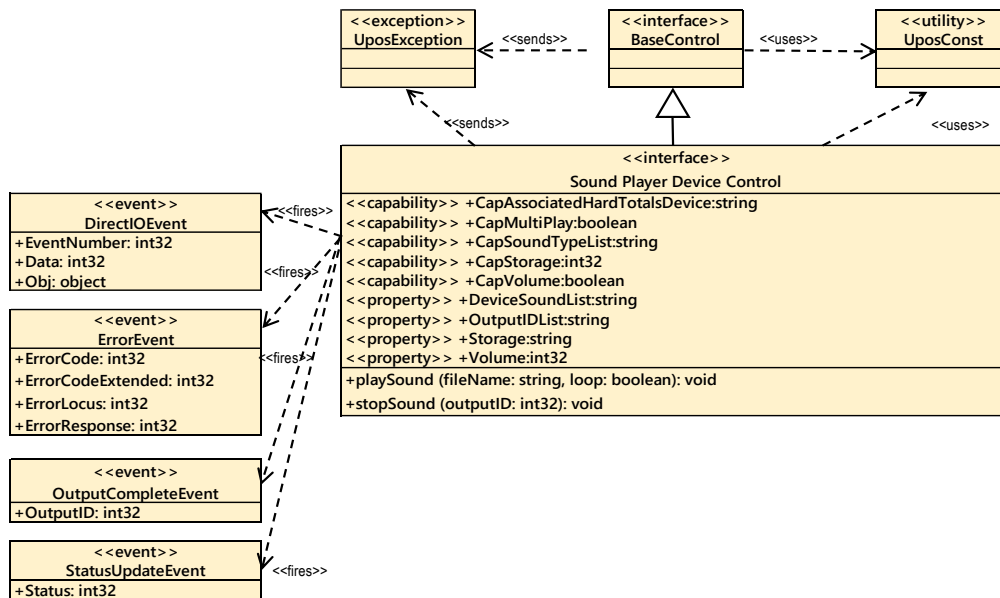


Fig. Chap.43-1 Sound Player Class Diagram

### 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:
- 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; 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.
- 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.

## **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.

## Properties(UML attributes)

### CapAssociatedHardTotalsDevice Property

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

[Goto Table96](#)

**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. **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** CapStorage Property

### 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.

[Goto Table97](#)

Value	Meaning
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.

**See Also** Storage Property, CapAssociatedHardTotalsDevice Property

### CapVolume Property

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

**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.

### 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}

[Goto Table155](#)

**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

## 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 <b>CapAssociatedHardTotalsDevice</b> 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 <b>CapAssociatedHardTotalsDevice</b> holds the open name of the associated Hard Totals device.

[Goto Table98](#)

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

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

**See Also** CapStorage Property

### 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

## 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 <b>DeviceSoundList</b> .
<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 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

### 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



## 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:

<b>Attribute</b>	<b>Type</b>	<b>Description</b>
<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

## ErrorEvent

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

[Goto Table99](#)

**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:

<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. <b>It is indicating that the error occurred while processing asynchronous output. An error occurred during asynchronous action.</b>
<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:

<u>Value</u>	<u>Meaning</u>
ESPLY_NOROOM	The encoded data storage area does not have enough room to store.

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

<u>Value</u>	<u>Meaning</u>
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:

<u>Value</u>	<u>Meaning</u>
ER_RETRY	Retry <del>sending</del> the <b>asynchronous output</b> data. The error state is exited. <del>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.</del> This is the default response <del>when the locus is EL_OUTPUT.</del>
ER_CLEAR	<del>Valid for all loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data <del>(including all asynchronous output).</del> <b>(The effect is the same as when clearOutput method is called.)</b> 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

## 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:

<b>Attribute</b>	<b>Type</b>	<b>Description</b>
<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

## 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:

<b>Attributes</b>	<b>Type</b>	<b>Description</b>
<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.

<b>Value</b>	<b>Meaning</b>
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.

# 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>
<b>AutoDisable:</b>	<i>boolean</i>	{read-write}	1.16	<i>Not supported</i>
<b>CapCompareFirmwareVersion:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPowerReporting:</b>	<i>int32</i>	{read-only}	1.16	open
<b>CapStatisticsReporting:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateFirmware:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapUpdateStatistics:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CheckHealthText:</b>	<i>string</i>	{read-only}	1.16	open
<b>Claimed:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>DataCount:</b>	<i>int32</i>	{read-only}	1.16	<i>Not supported</i>
<b>DataEventEnabled:</b>	<i>boolean</i>	{read-write}	1.16	<i>Not supported</i>
<b>DeviceEnabled:</b>	<i>boolean</i>	{read-write}	1.16	open & claim
<b>FreezeEvents:</b>	<i>boolean</i>	{read-write}	1.16	open
<b>OutputID:</b>	<i>int32</i>	{read-only}	1.16	open
<b>PowerNotify:</b>	<i>int32</i>	{read-write}	1.16	open
<b>PowerState:</b>	<i>int32</i>	{read-only}	1.16	open
<b>State:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceControlDescription:</b>	<i>string</i>	{read-only}	1.16	--
<b>DeviceControlVersion:</b>	<i>int32</i>	{read-only}	1.16	--
<b>DeviceServiceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>DeviceServiceVersion:</b>	<i>int32</i>	{read-only}	1.16	open
<b>PhysicalDeviceDescription:</b>	<i>string</i>	{read-only}	1.16	open
<b>PhysicalDeviceName:</b>	<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>
<b>CapLanguage:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapPitch:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapSpeed:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVoice:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVolume:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>Language:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>LanguageList:</b>	<i>string</i>	{read-only}	1.16	open
<b>OutputIDList:</b>	<i>string</i>	{read-only}	1.16	open, claim & enable
<b>Pitch:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>Speed:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>Voice:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>VoiceList:</b>	<i>string</i>	{read-only}	1.16	open
<b>Volume:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable

### Methods (UML operations)

#### Common

<i>Name</i>	<i>Version</i>	
<b>open (logicalDeviceName: <i>string</i>):</b> void {raises-exception}	1.16	
<b>close ():</b> void {raises-exception, use after open}	1.16	
<b>claim ( timeout: <i>int32</i>):</b> void {raises-exception, use after open}	1.16	
<b>release ():</b> void {raises-exception, use after open, claim}	1.16	
<b>checkHealth (level: <i>int32</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table100</a>
<b>clearInput ():</b> void { <b>raises-exception, use after open, claim</b> }	1.16	<a href="#">Goto Table101</a>
<b>clearInputProperties ():</b> void { <b>raises-exception, use after open, claim</b> }	1.16	<a href="#">Goto Table102</a>

## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)(continued)

**clearOutput ():**  
void {raises-exception, use after open, claim} 1.16

[Goto Table103](#)

#### Common

<i>Name</i>	<i>Version</i>
<b>compareFirmwareVersion (firmwareFileName: <i>string</i>, out result: <i>int32</i>):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>directIO (command: <i>int32</i>, inout data: <i>int32</i>, inout obj: <i>object</i>):</b> void {raises-exception, use after open}	1.16
<b>resetStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>retrieveStatistics (inout statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>updateFirmware (firmwareFileName: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>updateStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16

#### Specific

<i>Name</i>	<i>Version</i>
<b>speak (text: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>speakImmediate (text: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>stopCurrentSpeaking ():</b> void {raises-exception, use after open, claim, enable}	1.16
<b>stopSpeaking (outputID: <i>int32</i>):</b> 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>
<b>upos::events::DataEvent</b>		<i>Not supported</i>	
<b>upos::events::DirectIOEvent</b>			1.16
<b>EventNumber:</b>	<i>int32</i>	{read-only}	
<b>Data:</b>	<i>int32</i>	{read-write}	
<b>Obj:</b>	<i>object</i>	{read-write}	
<b>upos::events::ErrorEvent</b>			1.16
<b>ErrorCode:</b>	<i>int32</i>	{read-only}	
<b>ErrorCodeExtended:</b>	<i>int32</i>	{read-only}	
<b>ErrorLocus:</b>	<i>int32</i>	{read-only}	
<b>*pErrorResponse:</b>	<i>int32</i>	{read-write}	
<b>upos::events::OutputCompleteEvent</b>			1.16
<b>OutputID:</b>	<i>int32</i>	{read-only}	
<b>upos::events::StatusUpdateEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	

## General Information

The Speech Synthesis programmatic name is "Speech Synthesis".

### Capabilities

The Speech Synthesis has the following capability:

- Convert text to speech and read it aloud.

### Speech Synthesis Class Diagram

The following diagram shows the relationships between the Speech Synthesis classes.

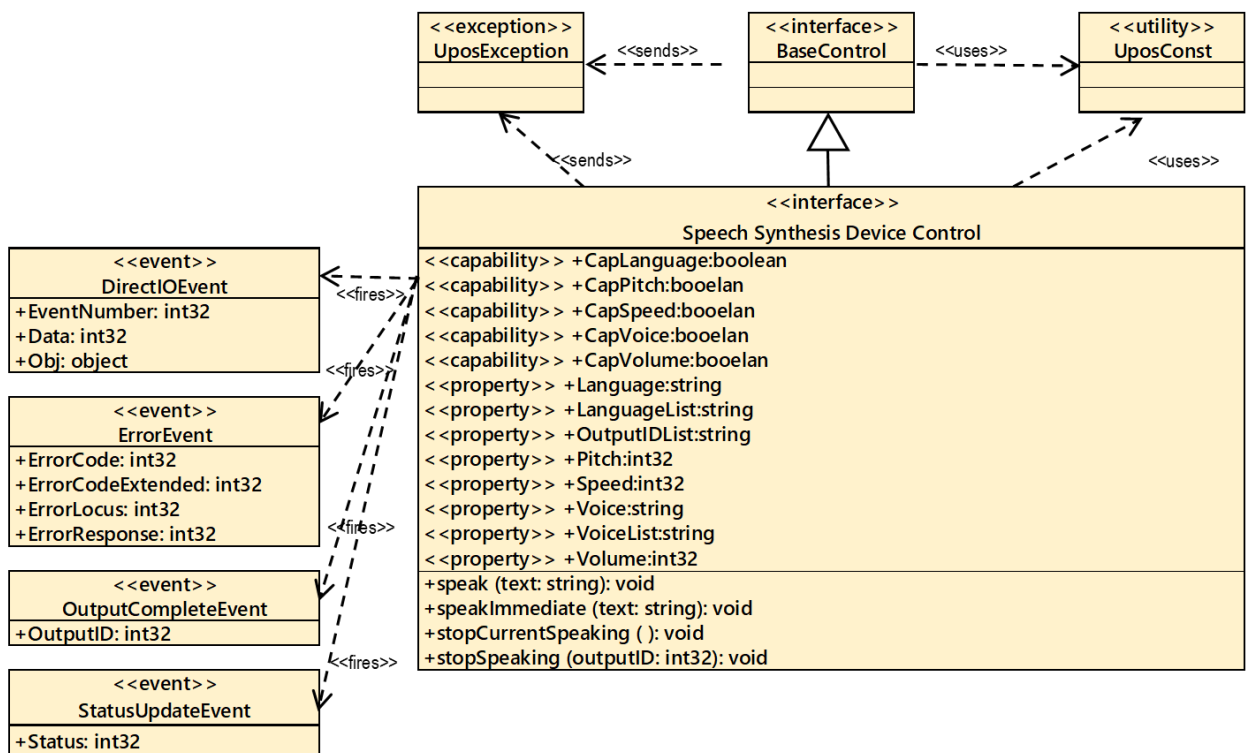


Fig. Chap. 44-1 Speech Synthesis Class Diagram



### Model

The Speech Synthesis follows the general device behavior model for 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.

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 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.

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

## UPOS Ver1.16 RCSD Specification

### Device Sharing

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.

## Properties (UML attributes)

### CapLanguage Property

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

### CapPitch Property

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

### CapSpeed Property

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

### CapVoice Property

<b>Syntax</b>	<b>CapVoice: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, the application can change the voice. If false, the application cannot change the voice.  This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this property is accessed. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>Voice</b> 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~~only, access after open-claim-enable}
- Remarks**      Comma-separated list of **OutputID** property values of audio being played by **speak** method or **speakImmediate** 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

[Goto Table156](#)

## 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 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:
- | <b>Value</b> | <b>Meaning</b>                  |
|--------------|---------------------------------|
| 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}`

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

<i>Text</i>	Specify the text to speak.
-------------	----------------------------

Remarks 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.

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.	-	-

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,volue=80}Today,{reset}It's nice weather.

## UPOS Ver1.16 RCSD Specification

Those utterance defined as follows.

Name	Data	Remarks	
Utterance written by text with the <b>speak</b> 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:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified. The language set in the <b>Language</b> 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



**UPOS Ver1.16 RCSD Specification**  
**SpeakImmediate Method**

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

<b>Parameter</b>	<b>Description</b>
<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:

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	An invalid value was specified. The language set in the <b>Language</b> 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

**stopCurrentSpeaking Method**

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

**Remarks**      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:

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	Speech is not running.

**See Also**      **speak** Method, **speakImmediate** Method

**UPOS Ver1.16 RCSD Specification**  
**stopSpeaking Method**

[Goto Table157](#)

**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 <b>OutputID</b> 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

**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}
```

[Goto Table104](#)

**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:

<b>Attributes</b>	<b>Type</b>	<b>Description</b>
<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. <b>It is indicating that the error occurred while processing asynchronous output. <del>An error occurred during asynchronous action</del></b>
<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:

<b>Value</b>	<b>Meaning</b>
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:

<b>Value</b>	<b>Meaning</b>
ER_RETRY	Retry <del>sending the asynchronous output data</del> . The error state is exited. <del>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.</del> This is the default response <del>when the locus is EL_OUTPUT</del> .
ER_CLEAR	<del>Valid for loci: EL_OUTPUT</del> . Clear all buffered <del>input or</del> output data (including all asynchronous output). <b>(The effect is the same as when clearOutput method is called.)</b> 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.

## 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

## 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.

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

## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)

#### Common

#### 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
CapMotion:	boolean	{read-only}	1.16	open
CapMotionCreation:	boolean	{read-only}	1.16	open
CapPose:	boolean	{read-only}	1.16	open
CapPoseCreation:	boolean	{read-only}	1.16	open
CapStorage:	int32	{read-only}	1.16	open
AutoMode:	string	{read-write}	1.16	open, claim & enable
AutoModeList:	string	{read-only}	1.16	open
JointList:	string	{read-only}	1.16	open
MotionList:	string	{read-only}	1.16	open
PoseCreationMode:	boolean	{read-write}	1.16	open, claim & enable
PoseList:	string	{read-only}	1.16	open
Storage:	int32	{read-write}	1.16	open, claim & enable

#### *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, <b>claim</b> , enable}	1.16	<a href="#">Goto Table105</a>
clearInput (): void {raises-exception, use after open, claim}	1.16	<a href="#">Goto Table106</a>
clearInputProperties (): void {raises-exception, use after open, claim}	1.16	<a href="#">Goto Table107</a>
clearOutput (): void {raises-exception, use after open, claim}	1.16	<a href="#">Goto Table108</a>
compareFirmwareVersion (firmwareFileName: <i>string</i> , out result: <i>int32</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table109</a>
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, <b>claim</b> , enable}	1.16	<a href="#">Goto Table110</a>
retrieveStatistics (inout statisticsBuffer: <i>string</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table111</a>
updateFirmware (firmwareFileName: <i>string</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table112</a>
updateStatistics (statisticsBuffer: <i>string</i> ): void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table113</a>

## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)(continued)

#### Specific

<i>Name</i>	<i>Version</i>
<b>createMotion</b> (fileName: <i>string</i> , poseList: <i>string</i> ): void { raises-exception, use after open, claim, enable }	1.16
<b>createPose</b> (fileName: <i>string</i> , time: <i>int32</i> ): void { raises-exception, use after open, claim, enable }	1.16
<b>getPosition</b> (jointID: <i>string</i> , out position: <i>int32</i> ): void { raises-exception, use after open, claim, enable }	1.16
<b>setPosition</b> (positionList: <i>string</i> , time: <i>int32</i> , absolute: <i>boolean</i> ): void { raises-exception, use after open, claim, enable }	1.16
<b>setSpeed</b> (speedList: <i>string</i> , time: <i>int32</i> ): void { raises-exception, use after open, claim, enable }	1.16
<b>startMotion</b> (fileName: <i>string</i> ): void { raises-exception, use after open, claim, enable }	1.16
<b>startPose</b> (fileName: <i>string</i> ): void { raises-exception, use after open, claim, enable }	1.16
<b>stopControl</b> (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>
<b>upos::events::DataEvent</b>		<i>Not supported</i>	
<b>upos::events::DirectIOEvent</b>			1.16
<b>EventNumber:</b>	<i>int32</i>	{read-only}	
<b>Data:</b>	<i>int32</i>	{read-write}	
<b>Obj:</b>	<i>object</i>	{read-write}	
<b>upos::events::ErrorEvent</b>			1.16
<b>ErrorCode:</b>	<i>int32</i>	{read-only}	
<b>ErrorCodeExtended:</b>	<i>int32</i>	{read-only}	
<b>ErrorLocus:</b>	<i>int32</i>	{read-only}	
<b>ErrorResponse:</b>	<i>int32</i>	{read-write}	
<b>upos::events::OutputCompleteEvent</b>			1.16
<b>OutputID:</b>	<i>int32</i>	{read-only}	
<b>upos::events::StatusUpdateEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	

## 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.

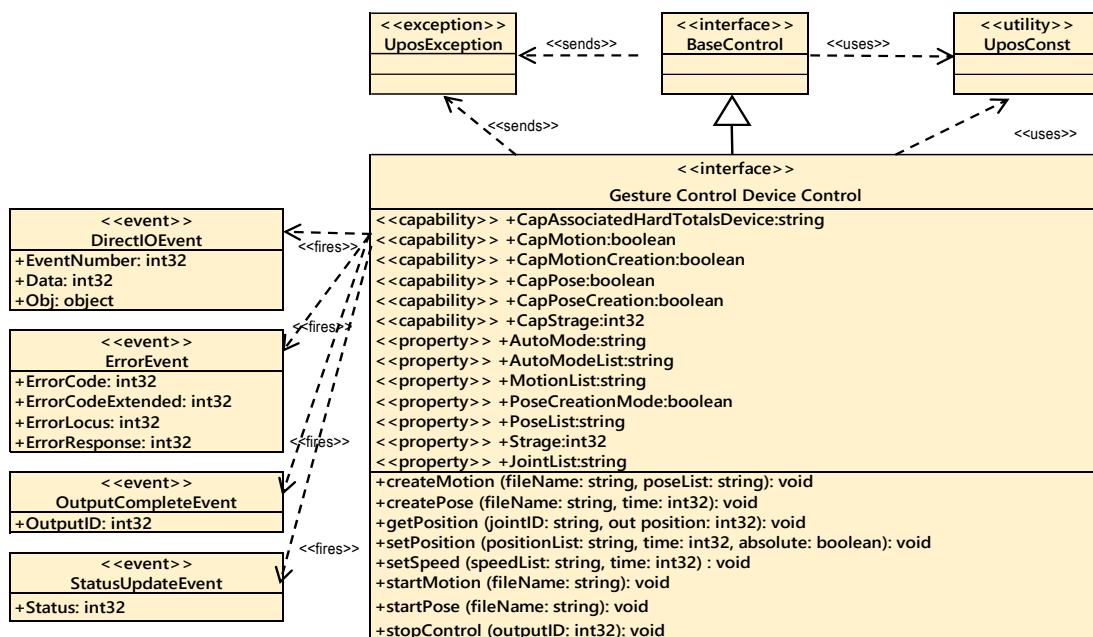


Fig. Chap. 45-1 Gesture Control Class Diagram

## 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.

## 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, application can create a pose file by setting the value 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 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 may store in either the “Hard Totals” 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.

### 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

<b>Syntax</b>	<b>AutoMode:</b> <i>string</i> {read-write, access after open-claim-enable}				
<b>Remarks</b>	Indicates automatic control mode ID. Valid values are the empty string "" or one of the <b>AutoModeList</b> properties listed.  If one of the properties described in the <b>AutoModeList</b> property is set, the automatic control mode will be enabled in the set mode.  Setting the empty character "" disables the automatic control mode.  This property is initialized to the empty string "" by the <b>open</b> method.				
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20.  Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><b>Value</b></th><th><b>Meaning</b></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	<b>Value</b>	<b>Meaning</b>	E_ILLEGAL	An invalid value was specified.
<b>Value</b>	<b>Meaning</b>				
E_ILLEGAL	An invalid value was specified.				
<b>See Also</b>	<b>AutoModeList</b> Property				

### AutoModeList Property

<b>Syntax</b>	<b>AutoModeList:</b> <i>string</i> {read-only, access after open}
<b>Remarks</b>	Comma-separated list of joint automatic control IDs supported by the device.  For example, in conjunction with the camera, if the mode of tracking the face of a person by moving only the joint of Joint01, this is “FaceTrack_Joint01”.  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”.  (Content and order are dependent on the device.)  This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>AutoMode</b> Property.

### CapAssociatedHardTotalsDevice Property

<b>Syntax</b>	<b>CapAssociatedHardTotalsDevice:</b> <i>string</i> {read-only, access after open}
<b>Remarks</b>	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 <b>CapStorage</b> is either GCTL_CST_ALL or GCTL_CST_HARDTOTALS_ONLY. If <b>CapStorage</b> is GCTL_CST_HOST_ONLY this property value must be the empty string. <b>This property is initialized by the open method.</b>
<b>Errors</b>	UposException may be thrown when this property is accessed. <a href="#">Goto Table114</a> For further information, see “Errors” on page Intro-20.
<b>See Also</b>	<b>CapStorage</b> Property

## UPOS Ver1.16 RCSD Specification

### CapMotion Property

<b>Syntax</b>	<b>CapMotion: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, the device supports making the motion function. Otherwise, it is false. When this property is false, <b>startMotion</b> method, <b>createMotion</b> method is not available. This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>startMotion</b> Method, <b>createMotion</b> Method.

### CapMotionCreation Property

<b>Syntax</b>	<b>CapMotionCreation: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, the device supports motion registration function.  If false, the device does not support motion registration function.  If this property is FALSE, the <b>createMotion</b> method is not available.  This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>createMotion</b> Method.

### CapPose Property

<b>Syntax</b>	<b>CapPose: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, the device supports pose function. Otherwise, it is false.  When this property is FALSE, <b>PoseCreationMode</b> property value cannot be changed, in addition, <b>startPose</b> method, and <b>createPose</b> method are not available.  This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>PoseCreationMode</b> Property, <b>startPose</b> Method, <b>createPose</b> Method.

### CapPoseCreation Property

<b>Syntax</b>	<b>CapPoseCreation: <i>boolean</i> {read-only, access after open}</b>
<b>Remarks</b>	If true, the device supports pose registration function.  If false, the device does not support pose registration function.  When this property is FALSE, the <b>createPose</b> method that can change the <b>PoseCreationMode</b> property is not available.  This property is initialized by the <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>PoseCreationMode</b> Property, <b>createPose</b> Method.

### 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.

<b>Value</b>	<b>Meaning</b>
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

### 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 (":").

<b>Parameter</b>	<b>Description</b>
<i>JointID</i>	Indicates a unique ID in the service that identifies the joint. Position range availability: 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, 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.

### 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.

### 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.



## 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.

<b>Value</b>	<b>Meaning</b>
GCTL_ST_HARDTOTALS	The motion and/or pose data file is written to the associated Hard Totals device. The property <b>CapAssociatedHardTotalsDevice</b> 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 <b>CapAssociatedHardTotalsDevice</b> 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.

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

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

**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

## 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 recorded as motion.
<i>poseList</i>	Specify the comma-separated list of pose information to be registered.

**Remarks** 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 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.

### 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, it needs to set the **PoseCreationMode** property to TRUE and to make enabling pose registration mode.

The place where the motion file is recorded is the area 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.

## UPOS Ver1.16 RCSD Specification

### getPosition Method

**Syntax**      **getPosition (jointID: *string*, out position: *int32*):**  
                              **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. And specified JointList property should be the position range present one.
<i>position</i>	Store the specified value as the position associated with jointID.

**Remarks**      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.

**UPOS Ver1.16 RCSD Specification**  
**setPosition Method**

**Syntax**      **setPosition (positionList: *string*, time: *int32*, absolute: *boolean*): void {raises-exception, use after open-claim-enable}**

<b>Parameter</b>	<b>Description</b>
<i>positionList</i>	Specify the position information in a comma-separated list.
<i>time</i>	Specify the time of device 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 (":").

<b>Parameter</b>	<b>Description</b>
<i>jointID</i>	Specify the joint ID. Specify one of the values listed in the <b>JointList</b> property. However, it must be an ID whose position range is present.
<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:

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	An invalid value was specified.

**See Also**      **JointList** Property, **stopControl** Method.

## UPOS Ver1.16 RCSD Specification setSpeed Method

**Syntax**     **setSpeed** (*speedList: string, time: int32*):  
                  **void** {raises-exception, use after open-claim-enable}

<b>Parameter</b>	<b>Description</b>
<i>speedList</i>	Specify speed information in a comma-separated list.
<i>time</i>	Specify the time to device control in seconds. If the value of FOREVER(-1) is specified, it will continue to operate until you call the <b>stopControl</b> 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 (":").

<b>Parameter</b>	<b>Description</b>
<i>jointID</i>	Specify the joint ID. Specify one of the values listed in the <b>JointList</b> 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:

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	An invalid value was specified.

**See Also**     **JointList** Property, **stopControl** Method.

**startMotion Method**

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

<b>Parameter</b>	<b>Description</b>
<i>fileName</i>	Prior to start this method, need to specify the name of the motion file or the motion ID value that is listed in the <b>MotionList</b> property.

**Remarks**      Start the motion defined by fileName or motion defined by the device. 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:

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	An invalid value was specified.
E_NOEXIST	File does not exist.

**See Also**      **MotionList** Property.

**startPose Method**

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

Parameter	Description
<i>fileName</i>	Specify the name of the pose file to start. Or one of the pose ID lists listed in the <b>PoseList</b> property.

**Remarks** Start the pose defined by the 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 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:

Value	Meaning
E_ILLEGAL	An invalid value was specified.
E_NOEXISTS	File does not exist.

**See Also** **PoseList** Property, **stopControl** Method.

**stopControl Method**

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

Parameter	Description
<i>outputID</i>	Specify the value of the <b>OutputID</b> property to be 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:

Value	Meaning
E_ILLEGAL	An invalid value was specified.

**See Also** **setPosition** Method, **setSpeed** Method, **startPose** Method, **startMotion** Method.



## 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:

<b>Attribute</b>	<b>Type</b>	<b>Description</b>
<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

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}
```

[Goto Table115](#)

**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 <b>it is indicating that error occurred while processing asynchronous output.</b> <del>An error occurred during asynchronous action.</del>
<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 <i>ErrorResponse</i> attribute to one of the following values:

Value	Meaning
ER_RETRY	Retry <del>sending</del> the <b>asynchronous output data.</b> The error state is exited. <del>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.</del> This is the default response <del>when the locus is EL_OUTPUT.</del>
ER_CLEAR	<del>Valid for all loci: EL_OUTPUT.</del> Clear all buffered input or output data <del>(including all asynchronous output).</del> <b>(The effect is the same as when clearOutput method is called.)</b> 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

## 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

## 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.

# Device Monitor

This Chapter defines the Device Monitor device category.

## Summary

### Properties (UML attributes)

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

### Methods (UML operations)

#### Common

<i>Name</i>	<i>Version</i>	
<b>open (logicalDeviceName: <i>string</i>):</b> void {raises-exception}	1.16	
<b>close ():</b> void {raises-exception, use after open}	1.16	
<b>claim (timeout: <i>int32</i>):</b> void {raises-exception, use after open}	1.16	
<b>release ():</b> void {raises-exception, use after open, claim}	1.16	
<b>checkHealth (level: <i>int32</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table116</a>
<b>clearInput ():</b> void { <b>raises-exception, use after open, claim</b> }	1.16	<a href="#">Goto Table117</a>
<b>clearInputProperties ():</b> void { <b>raises-exception, use after open, claim</b> }	1.16	<a href="#">Goto Table118</a>
<b>clearOutput ():</b> void { }	<i>Not supported</i>	
<b>compareFirmwareVersion (firmwareFileName: <i>string</i>, out result: <i>int32</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table119</a>
<b>directIO (command: <i>int32</i>, inout data: <i>int32</i>, inout obj: <i>object</i>):</b> void {raises-exception, use after open}	1.16	
<b>resetStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table120</a>
<b>retrieveStatistics (inout statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table121</a>
<b>updateFirmware (firmwareFileName: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table122</a>
<b>updateStatistics (statisticsBuffer: <i>string</i>):</b> void {raises-exception, use after open, <b>claim</b> , enable}	1.16	<a href="#">Goto Table123</a>

#### Specific

<b>addMonitoringDevice (deviceID: <i>string</i>, monitoringMode: <i>int32</i>, boundary: <i>int32</i>, subBoundary: <i>int32</i>, intervalTime: <i>int32</i>):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>clearMonitoringDevices ():</b> void {raises-exception, use after open, claim, enable}	1.16
<b>deleteMonitoringDevice (deviceID: <i>string</i>):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>getDeviceValue (deviceID: <i>string</i>, pValue: <i>int32</i>):</b> void {raises-exception, use after open}	1.16

**UPOS Ver1.16 RCSD Specification****Events (UML interfaces)**

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

## 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.

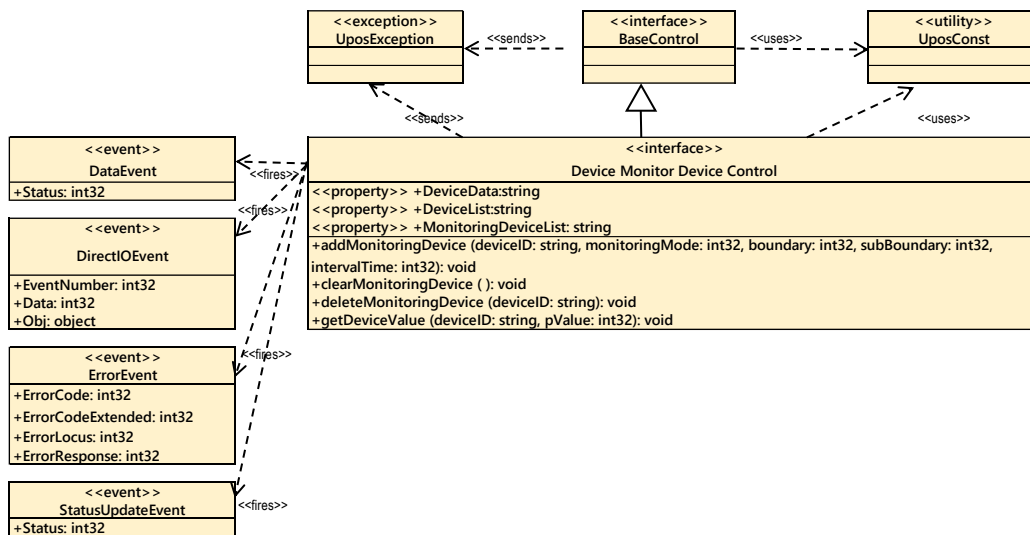


Fig. Chap. 46-1 Device Monitor Class Diagram

## 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 are managed most of cases with the int32 type integers, but some are decimals.
- In that case, the decimals are implicit, and the actual value can be calculated by dividing the measured value by the coefficient of each device that can be 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.

## 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.



## UPOS Ver1.16 RCSD Specification

# 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 (":").

<u>Parameter</u>	<u>Description</u>
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 <b>DeviceList</b> property.  For example, "Device01:365"  Its value is set prior to a <b>DataEvent</b> 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 (":").

<b>Parameter</b>	<b>Description</b>
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 <b>DeviceData</b> property and the measured value of the device that can be obtained with the <b>GetDeviceValue</b> 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.

## 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 <b>DeviceList</b> 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, the event will be notified in each time. When it is set to this mode, the value of the argument subBoundary is ignored.

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, the event will be notified in each time.

When it is set to this mode, the value of the argument subBoundary is ignored.

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, the event will be notified in each time.

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**.

### 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}`

<u>Parameter</u>	<u>Description</u>
<code>deviceID</code>	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:

<u>Value</u>	<u>Description</u>
<code>E_ILLEGAL</code>	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, pValue: *int32):`  
   `void {raises-exception, use after open}`

<u>Parameter</u>	<u>Description</u>
<code>deviceID</code>	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 <code>DeviceList</code> property.
<code>pValue</code>	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:

<u>Value</u>	<u>Description</u>
<code>E_ILLEGAL</code>	An invalid value was specified, or unsupported operation with the Device.

**See Also**     `DeviceList` Property.

## 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:

<b>Attribute</b>	<b>Type</b>	<b>Description</b>
<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.

#### 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:

<b>Attribute</b>	<b>Type</b>	<b>Description</b>
<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

## 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:

<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.
<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:

<u>Value</u>	<u>Meaning</u>
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:

<u>Value</u>	<u>Meaning</u>
<del>ER_RETRY</del>	<del>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.</del>

[Goto Table 124](#)

ER\_CLEAR Valid for all locus: 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,

## 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.

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

**Properties (Continued)**

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
<b>CapAssociatedHardTotalsDevice:</b>	<i>string</i>	{read-only}	1.16	open
<b>CapBrightness:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapImageType:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapStorage:</b>	<i>int32</i>	{read-only}	1.16	open
<b>CapURLBack:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapURLForward:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVideoType:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>CapVolume:</b>	<i>boolean</i>	{read-only}	1.16	open
<b>Brightness:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>DisplayMode:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>ImageType:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>ImageTypeList:</b>	<i>string</i>	{read-only}	1.16	open
<b>LoadStatus:</b>	<i>int32</i>	{read-only}	1.16	open
<b>Storage:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable
<b>URL:</b>	<i>string</i>	{read-only}	1.16	open
<b>VideoType:</b>	<i>string</i>	{read-write}	1.16	open, claim & enable
<b>VideoTypeList:</b>	<i>string</i>	{read-only}	1.16	open
<b>Volume:</b>	<i>int32</i>	{read-write}	1.16	open, claim & enable

**Methods (UML operations)**

**Common**

<i>Name</i>	<i>Version</i>	
<b>open (logicalDeviceName: <i>string</i>):</b> void {raises-exception}	1.16	
<b>close ():</b> void {raises-exception, use after open}	1.16	
<b>claim (timeout: <i>int32</i>):</b> void {raises-exception, use after open}	1.16	
<b>release ():</b> void {raises-exception, use after open, claim}	1.16	
<b>checkHealth (level: <i>int32</i>):</b> void {raises-exception, use after open, <b>claim</b> , <b>enable</b> }	1.16	<a href="#">Goto Table125</a>
<b>clearInput ():</b> void {raises-exception, use after open, <b>claim</b> }	1.16	<a href="#">Goto Table126</a>
<b>clearInputProperties ():</b> void {raises-exception, use after open, <b>claim</b> }	1.16	<a href="#">Goto Table127</a>

## UPOS Ver1.16 RCSD Specification

### Methods (UML operations)(Continued)

<b>clearOutput ():</b> void {raises-exception, use after open, claim}	1.16	<a href="#">Goto Table128</a>
<b>compareFirmwareVersion (firmwareFileName: string, out result: int32):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table129</a>
<b>directIO (command: int32, inout data: int32, inout obj: object):</b> void {raises-exception, use after open}	1.16	
<b>resetStatistics (statisticsBuffer: string):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table130</a>
<b>retrieveStatistics (inout statisticsBuffer: string):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table131</a>
<b>updateFirmware (firmwareFileName: string):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table132</a>
<b>updateStatistics (statisticsBuffer: string):</b> void {raises-exception, use after open, claim, enable}	1.16	<a href="#">Goto Table133</a>

### Specific

<i>Name</i>	<i>Version</i>
<b>cancelURLLoading ():</b> void {raises-exception, use after open, claim, enable}	1.16
<b>goURLBack ():</b> void {raises-exception, use after open, claim, enable}	1.16
<b>goURLForward ():</b> void {raises-exception, use after open, claim, enable}	1.16
<b>loadImage (fileName: string):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>loadURL (uRL: string):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>playVideo (fileName: string, loop: boolean):</b> void {raises-exception, use after open, claim, enable}	1.16
<b>stopVideo ():</b> void {raises-exception, use after open, claim, enable}	1.16
<b>updateURLPage ():</b> 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>
<b>upos::events::DataEvent</b>			
<b>Status:</b>		<i>Not supported</i>	
<b>upos::events::DirectIOEvent</b>			1.16
<b>EventNumber:</b>	<i>int32</i>	{read-only}	
<b>Data:</b>	<i>int32</i>	{read-write}	
<b>Obj:</b>	<i>object</i>	{read-write}	
<b>upos::events::ErrorEvent</b>			1.16
<b>ErrorCode:</b>	<i>int32</i>	{read-only}	
<b>ErrorCodeExtended:</b>	<i>int32</i>	{read-only}	
<b>ErrorLocus:</b>	<i>int32</i>	{read-only}	
<b>ErrorResponse</b>	<i>int32</i>	{read-write}	
<b>upos::events::OutputCompleteEvent</b>			1.16
<b>OutputID:</b>	<i>int32</i>	{read-only}	
<b>upos::events::StatusUpdateEvent</b>			1.16
<b>Status:</b>	<i>int32</i>	{read-only}	
<b>upos::events::TransitionEvent</b>		<i>Not supported</i>	

## General Information

The Graphic Display programmatic name is “Graphic Display”.

### Capabilities

The Graphic Display has the following capability:

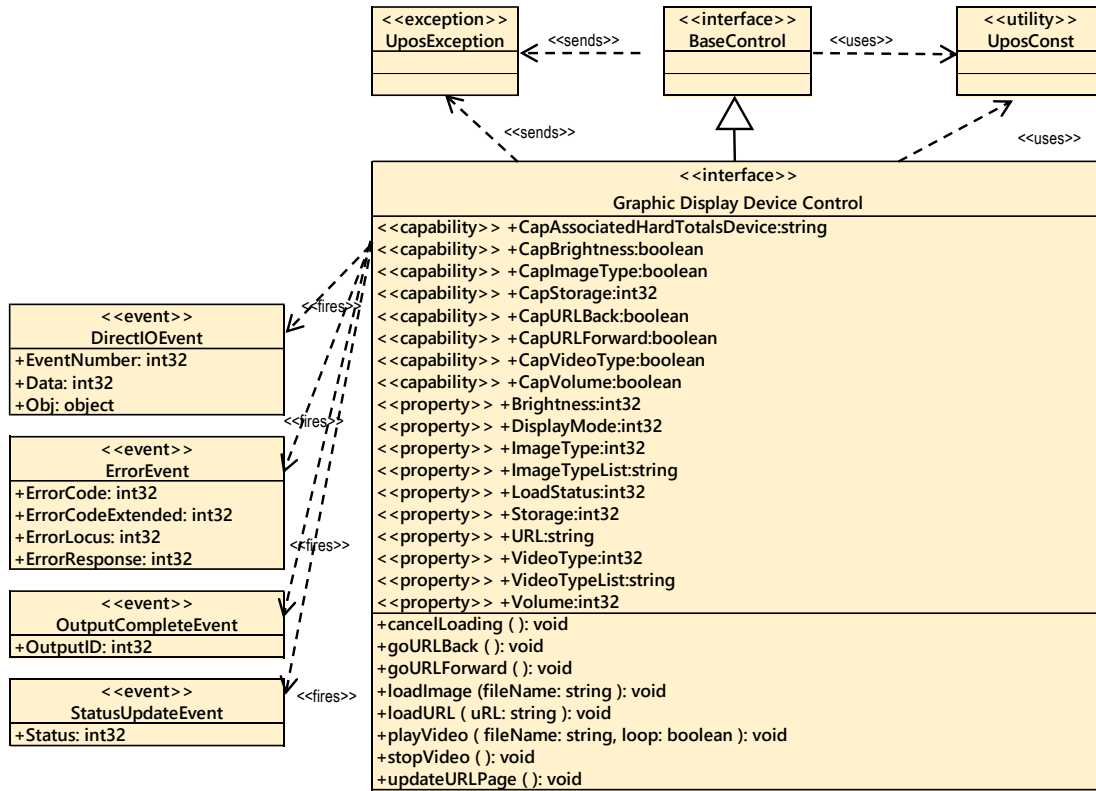
- Displays the specified image files.

- Play the specified video.

- Display the specified web page.

- Notify the application of changes in the load status of the web page.

## Graphics Display Class Diagram



The following diagram shows the relationships between the Graphic Display classes.

Fig. Chap. 47-1 Graphic Display Class Diagram

## Model

The following display modes exist in the graphics control, and the model differs depending on the display mode:

- Image display mode
- Video display mode.
- Web display mode.

The application can change the display mode by changing the value of the **DisplayMode** property.

### 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 **ImageTypeList** 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.~~

[Goto Table134](#)

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.

### 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.

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 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.



## UPOS Ver1.16 RCSD Specification

- 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 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.

[Goto Table135](#)

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 video files that the device can display are listed in the **VideoTypeList** property. 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.

### Web Display Mode

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 **cancelURLLoading** method is called during the loading process, current accessed URL information will be stored in the **URL** property.

[Goto Table136](#)

The graphics control web display mode follows an asynchronous output model.

## 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.

## Properties (UML attributes)

### Brightness Property

- Syntax**      **Brightness: *int32* {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 *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

**See Also**      **CapBrightness** Property.

### CapAssociatedHardTotalsDevice Property

- Syntax**      **CapAssociatedHardTotalsDevice: *string* {read-only~~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. **This property is initialized by the open method.**
- Errors**          UposException may be thrown when this property is accessed. [Goto Table137](#)  
For further information, see “Errors” on page Intro-20.

**See Also**      **CapStorage** Property

### CapBrightness Property

- Syntax**      **CapBrightness: *boolean* {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: *boolean* {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

## 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

## 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.

## 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.

## CapVideoType Property

## UPOS Ver1.16 RCSD Specification

<b>Syntax</b>	<b>CapVideoType:</b> <i>boolean</i> {read-only, access after open}
<b>Remarks</b>	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 <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>VideoType</b> Property, <b>VideoTypeList</b> Property

## CapVolume Property

<b>Syntax</b>	<b>CapVolume:</b> <i>boolean</i> {read-only, access after open}
<b>Remarks</b>	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 <b>open</b> method.
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20.
<b>See Also</b>	<b>Volume</b> Property.

## DisplayMode Property

<b>Syntax</b>	<b>DisplayMode:</b> <i>int32</i> {read-write, access after open-claim-enable}
---------------	---

## UPOS Ver1.16 RCSD Specification

**Remarks** Holds the image and/or video displaying mode.

<u>Value</u>	<u>Meaning</u>
GDSP_DMODE_HIDDEN	It is a mode to hide images and/or video
GDSP_DMODE_IMAGE_FIT	It is a mode to display images. The displayed image is enlarged / reduced to the size that maintains the aspect and fits on the screen.
GDSP_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.
GDSP_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.
GDSP_DMODE_VIDEO_NORMAL	It is a mode to display video. The displayed video will be displayed in the center of the screen without resizing.
GDSP_DMODE_VIDEO_FULL	It is a mode to display video. The displayed video will be displayed in full screen.
GDSP_DMODE_WEB	Display the web screen.

If application hide other modes and screens while displaying images, videos, or web, all displayed contents will be cleared. The video will be stopped while the video is playing.

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 Table138](#)

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** [loadImage Method](#), [playVideo Method](#), [CapCaptureColorSpaceList Property](#), [VideoCaptureMode Property](#) (They are Video Capture Device Properties)

## ImageType Property

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

## UPOS Ver1.16 RCSD Specification

**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. [Goto Table139](#)  
**Some possible values of the exception's ErrorCode property are:**

<b>Value</b>	<b>Meaning</b>
E_ILLEGAL	An invalid value was specified.

**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.

### ImageTypeList Property

**Syntax** **ImageTypeList: 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.

### 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.

[Goto Table140](#)

<b>Value</b>	<b>Meaning</b>
GDSP_LSTATUS_START	Start loading the web page.
GDSP_LSTATUS_FINISH	It has finished loading the web page.
GDSP_LSTATUS_CANCEL	It has canceled loading the web page

Its value is set prior to a **StatusUpdateEvent** 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.

### Storage Property

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

## UPOS Ver1.16 RCSD Specification

**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.

<u>Value</u>	<u>Meaning</u>
GDSP_ST_HARDTOTALS	The image data file is written to the associated Hard Totals device. The property <b>CapAssociatedHardTotalsDevice</b> 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 <b>CapAssociatedHardTotalsDevice</b> 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 GDSP\_CST\_ALL, it is initialized to GDSP\_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

## URL Property

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

**Remarks** When the **LoadStatus** property is GDSP\_LSTATUS\_START, the URL of the Web page that starts loading is set.

When the **LoadStatus** property is GDSP\_LSTATUS\_FINISH, the URL of the loaded Web page is set.

When the **LoadStatus** property is GDSP\_LSTATUS\_CANCEL, the URL of the canceled Web page is set.

Its value is set prior to a **StatusUpdateEvent** 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.



## 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. [Goto Table141](#)
- 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:**
- | <b>Value</b> | <b>Meaning</b>                  |
|--------------|---------------------------------|
| E_ILLEGAL    | An invalid value was specified. |
- See Also**      **CapVideoType** Property, **VideoTypeList** Property, **playVideo** Method.

## VideoTypeList Property

- Syntax**      **VideoTypeList:** *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.

## 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:
- | <b>Value</b> | <b>Meaning</b>                  |
|--------------|---------------------------------|
| E_ILLEGAL    | An invalid value was specified. |
- See Also**      **CapVolume** Property, **playVideo** Method.

## Methods (UML operations)

### cancelURLLoading Method

<b>Syntax</b>	<b>cancelURLLoading ( ):</b> <b>void {raises-exception, use after open-claim-enable}</b>
<b>Remarks</b>	Cancel loading web page. This method is executed asynchronously. The load status is reported by <b>StatusUpdateEvent</b> and <b>OutputCompleteEvent</b> or <b>ErrorEvent</b> .
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are:

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

### goURLBack Method

<b>Syntax</b>	<b>goURLBack ( ):</b> <b>void {raises-exception, use after open-claim-enable}</b>
<b>Remarks</b>	It returns to the previous page of browsing history. This method is executed asynchronously. The load status is reported by <b>StatusUpdateEvent</b> and <b>OutputCompleteEvent</b> or <b>ErrorEvent</b> .
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	There is no previous page in the browsing history.

**See Also** CapURLBack Property.

### goURLForward Method

<b>Syntax</b>	<b>goURLForward ( ):</b> <b>void {raises-exception, use after open-claim-enable}</b>
<b>Remarks</b>	Go to the next page of browsing history. This method is executed asynchronously. The load status is reported by <b>StatusUpdateEvent</b> and <b>OutputCompleteEvent</b> or <b>ErrorEvent</b> .
<b>Errors</b>	A UposException may be thrown when this method is invoked. For further information, see “ <b>Errors</b> ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	There is no next page in the browsing history.

**See Also** CapURLForward Property.

**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 GDSP\_DMODE\_IMAGE\_FIT, GDSP\_DMODE\_IMAGE\_FILL, or GDSP\_DMODE\_IMAGE\_CENTER.

Image files are located in the area 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.

**loadURL Method**

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

<u>Parameter</u>	<u>Description</u>
<i>uURL</i>	Specify the uRL of the web page to load.

**Remarks**      Load the web page with the specified URL.

This method is executed asynchronously. The load 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.

## playVideo Method

**Syntax**     **playVideo** (**fileName**: *string*, **loop**: *boolean*):  
                  **void** {raises-exception, use after open-claim-enable}

Parameter	Description
<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 **type file content** that **is specified using VideoType property** ~~are loaded in the storage area by the loadImage method~~. All of ~~loaded~~ the video file **values images** are listed in the **ImageVideoTypeList** property, **if CapVideoType property is true.**

[Goto Table142](#)

If the value of the **DisplayMode** property is not set to GD~~I~~SP\_DMODE\_VIDEO\_NORMAL, GD~~I~~SP\_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 located in the area 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:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or an unsupported video file was specified.
E_NOEXIST	File does not exist.

**See Also**     **DisplayMode** Property.

## stopVideo Method

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

[Goto Table143](#)

**Remarks**     Stop the video being displayed.

This method is executed asynchronously. **VideoImage** 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:

Value	Meaning
E_ILLEGAL	The Video is not playing.

**See Also**     ~~start~~**playVideo** Method.

## 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 **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	Web page loading.

## 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

### 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.

[Goto Table144](#)

**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, <b>it is indicating that the error occurred while processing asynchronous output-action. An error occurred during asynchronous action.</b>
<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
EGDSP_NOROOM	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
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 <del>sending</del> the asynchronous output <del>data</del> . The error state is exited. <del>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.</del> This is the default response <del>when the locus is EL_OUTPUT.</del>
ER_CLEAR	<del>Valid for loci: EL_OUTPUT.</del> Clear all buffered <del>input or</del> output data (including all asynchronous output). <b>(The effect is the same as when clearOutput method is called.)</b> 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.

## 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
<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

## 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:

Attributes	Type	Description
<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.

Value	Meaning
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.



## 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

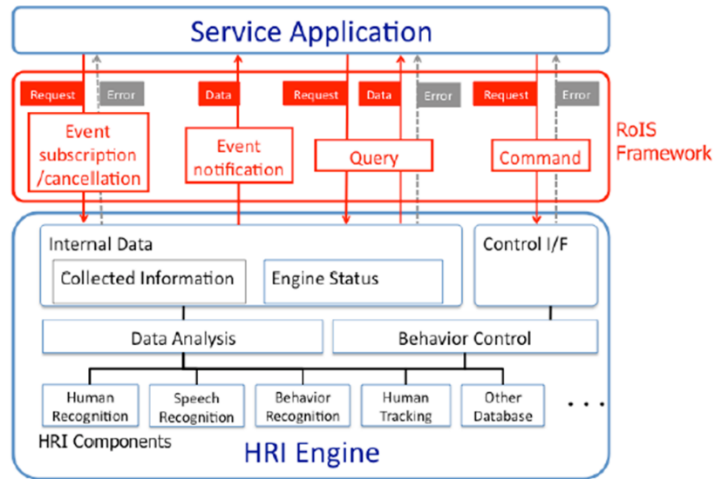


Fig.5: Example of RoIS Framework

## 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

## Interoperability between UPOS RCSD and Rols

### Rleationsihp 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

**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
Sound & Voice Recognition	gesture recognition	Recognizes person's gesture
	sound detection	Detects number of sound sources
	sound localization	Detects position of sound sources
Speech Synthesis	speech recognition	Recognizes person's speech
	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		

specification.

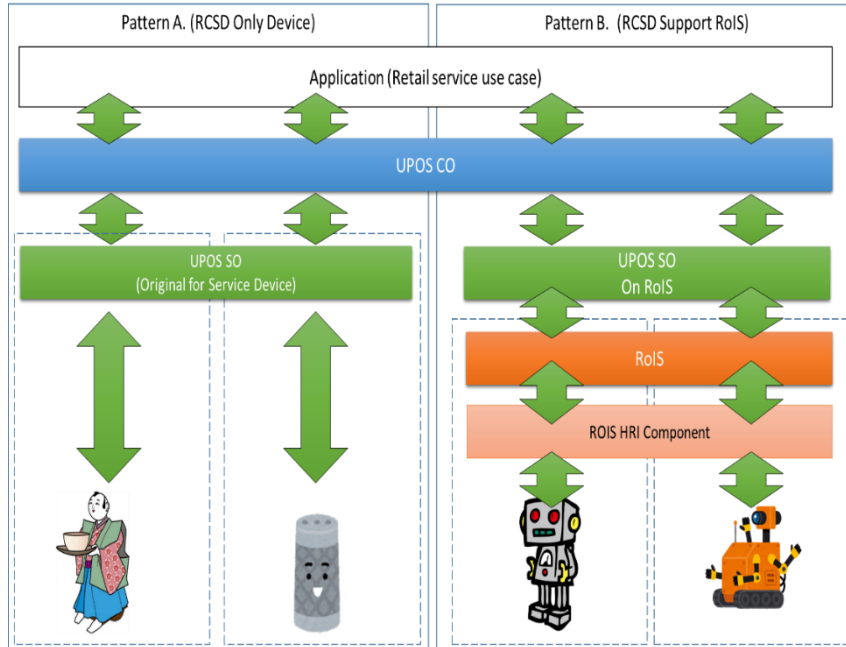
## UPOS Ver1.16 RCSD Specification

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.



## 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
1.4	2021-08-10		Issues, Updates are added version from the Review

## Glossary

Term	Definition
<b>EVRW</b>	Electronic Value Reader Writer
<b>CAT</b>	Credit Authorization Terminal

## UPOS 1.16.1 RCSD Issues Reference Table

No.	Chapter	Section	Item	Company /Name	Issues	Editing Result	Status	Remarks
Issue1 <a href="#">Table1</a> <a href="#">Table14</a> <a href="#">Table89</a> <a href="#">Table93</a> <a href="#">Table100</a> <a href="#">Table105</a> <a href="#">Table116</a> <a href="#">Table125</a>	21/Lights 39/Video Capture 42/Voice Recog. 43/Sound Player 44/Speech Synthe. 45/Gesture Control 46/Device Monitor 47/Graphic Display	Summary <b>checkHealth</b> Method	Method behavior condition description was incorrect. Since this method can be use after <b>open, claim, enable</b> . However <b>claim</b> was missing	OPOS-J / Tad Furuhata	Current description was as follows. This method can be use after <b>open, enable</b> .	It corrected as follows. This method can be use after <b>open, claim, enable</b> .	Resolved	
Issue2 <a href="#">Table2</a> <a href="#">Table16</a> <a href="#">Table70</a> <a href="#">Table79</a> <a href="#">Table109</a> <a href="#">Table119</a> <a href="#">Table129</a>	21/Lights 39/Video Capture 40/Ind. Recognition 41/Sound Recorder 45/Gesture Control 46/Device Monitor 47/Graphic Display	Summary <b>compareFirmwar</b> <b>eVersion</b> Method	Method behavior condition description was incorrect. Since this method can be use after <b>open, claim, enable</b> . However <b>claim</b> was missing	OPOS-J / Tad Furuhata	Current description was as follows. This method can be use after <b>open, enable</b> .	It corrected as follows. This method can be use after <b>open, claim, enable</b> .	Resolved	
Issue3 <a href="#">Table3</a> <a href="#">Table17</a> <a href="#">Table71</a> <a href="#">Table80</a> <a href="#">Table110</a> <a href="#">Table120</a> <a href="#">Table130</a>	21/Lights 39/Video Capture 40/Ind. Recognition 41/Sound Recorder 45/Gesture Control 46/Device Monitor 47/Graphic Display	Summary <b>resetStatistics</b> Method	Method behavior condition description was incorrect. Since this method can be use after <b>open, claim, enable</b> . However <b>claim</b> was missing	OPOS-J / Tad Furuhata	Current description was as follows. This method can be use after <b>open, enable</b> .	It corrected as follows. This method can be use after <b>open, claim, enable</b> .	Resolved	
Issue4 <a href="#">Table4</a> <a href="#">Table18</a> <a href="#">Table72</a> <a href="#">Table81</a> <a href="#">Table111</a> <a href="#">Table121</a> <a href="#">Table131</a>	21/Lights 39/Video Capture 41/Sound Recorder 40/Individual Recog. 45/Gesture Control 46/Device Monitor 47/Graphic Display	Summary <b>retrieveStatistics</b> Method	Method behavior condition description was incorrect. Since this method can be use after <b>open, claim, enable</b> . However <b>claim</b> was missing	OPOS-J / Tad Furuhata	Current description was as follows. This method can be use after <b>open, enable</b> .	It corrected as follows. This method can be use after <b>open, claim, enable</b> .	Resolved	
Issue5 <a href="#">Table5</a> <a href="#">Table19</a> <a href="#">Table73</a> <a href="#">Table82</a> <a href="#">Table112</a> <a href="#">Table122</a> <a href="#">Table132</a>	21/Lights 39/Video Capture 40/Individual Recog. 41/Sound Recorder 45/Gesture Control 46/Device Monitor 47/Graphic Display	Summary <b>updateFirmware</b> Method	Method behavior condition description was incorrect. Since this method can be use after <b>open, claim, enable</b> . However <b>claim</b> was missing	OPOS-J / Tad Furuhata	Current description was as follows. This method can be use after <b>open, enable</b> .	It corrected as follows. This method can be use after <b>open, claim, enable</b> .	Resolved	
Issue6 <a href="#">Table6</a> <a href="#">Table20</a> <a href="#">Table74</a> <a href="#">Table83</a> <a href="#">Table113</a> <a href="#">Table123</a> <a href="#">Table133</a>	21/Lights 39/Video Capture 40/Individual Recog. 41/Sound Recorder 45/Gesture Control 46/Device Monitor 47/Graphic Display	Summary <b>updateStatistics</b> Method	Method behavior condition description was incorrect. Since this method can be use after <b>open, claim, enable</b> . However <b>claim</b> was missing	OPOS-J / Tad Furuhata	Current description was as follows. This method can be use after <b>open, enable</b> .	It corrected as follows. This method can be use after <b>open, claim, enable</b> .	Resolved	

## UPOS Ver1.16 RCSD Specification

Issue7 <a href="#">Table7</a>	29/POS Power	Summary <b>UPSChargeState</b> Property	Property behavior condition description was incorrect. This property may use after open & enable. However, mistakenly claim was added.	OPOS-J / Tad Furuhata	Current description was as follows. This property may use after <b>open, claim &amp; enable.</b>	It corrected as follows. This property may use after <b>open &amp; enable.</b>	Resolved	
Issue8 <a href="#">Table8</a>	29/POS Power	<b>BatteryCritically LowThreshold</b> Property	The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	This property is able to <b>read</b> and <b>write</b> . In that case there should be the E_ILLEGAL error Code.	E_ILLEGAL related description was added.  Some possible values of the exception's ErrorCode property are: <b>Value      Meaning</b> ----- E_ILLEGAL    An invalid value was specified. Or it does not support this function.	Resolved	
Issue9 <a href="#">Table9</a>	29/POS Power	<b>BatteryCritically LowThresholdIn Seconds</b> Property	The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	This property is able to <b>read</b> and <b>write</b> . In that case there should be the E_ILLEGAL error Code.	E_ILLEGAL related description was added.  Some possible values of the exception's ErrorCode property are: <b>Value      Meaning</b> ----- E_ILLEGAL    An invalid value was specified. Or it does not support this function.	Resolved	
Issue10 <a href="#">Table10</a>	29/POS Power	<b>BatteryLowThres hold</b> Property	The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	This property is able to read and write. In that case there should be the E_ILLEGAL error Code.	E_ILLEGAL related description was added.  Some possible values of the exception's ErrorCode property are: <b>Value      Meaning</b> ----- E_ILLEGAL    An invalid value was specified. Or it does not support this function.	Resolved	
Issue11 <a href="#">Table11</a>	29/POS Power	<b>BatteryLowThres holdInSeconds</b> Property	The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	This property is able to read and write. In that case there should be the E_ILLEGAL error Code.	E_ILLEGAL related description was added.  Some possible values of the exception's ErrorCode property are: <b>Value      Meaning</b> ----- E_ILLEGAL    An invalid value was specified. Or it does not support this function.	Resolved	
Issue12 <a href="#">Table12</a>	29/POS Power	<b>EnforcedShutdo wnDelayTime</b> Property	The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	This property is able to read and write. In that case there should be the error Code.	E_ILLEGAL related description was added.  Some possible values of the exception's ErrorCode property are: <b>Value      Meaning</b> ----- E_ILLEGAL    An invalid value was specified. Or it does not support this function.	Resolved	
Issue13 <a href="#">Table13</a>	39/Video Capture	Summary <b>VideoCaptureMode</b> Property	This property's mutability is <b>read-write</b> . However description was <b>read-only</b> and this is corrected.	OPOS-J / Tad Furuhata	Currently, this property's mutability described as <b>read-only</b> .	This property's mutability corrected as <b>read-write</b> .	Resolved	
Issue15 <a href="#">Table15</a> <a href="#">Table68</a> <a href="#">Table77</a> <a href="#">Table90</a> <a href="#">Table94</a> <a href="#">Table101</a> <a href="#">Table106</a>	39/Video Capture 40/Indi. Recognition 41/Sound Recorder 42/Voice Recog. 43/Sound Player 44/Speech Synthe. 45/Gesture Control 46/Device Monitor	Summary <b>clearInput</b> Method	Missing the method behavior, therefore they were added.	OPOS-J / Tad Furuhata	Current description was as follows. Method name <b>void: {}</b>	Method behavior description was changed as follows. Method name <b>void: {raises-exception, use after open, claim}</b>	Resolved	



## UPOS Ver1.16 RCSD Specification

<a href="#">Table117</a> <a href="#">Table126</a>	47/Graphic Display							
Issue21 <a href="#">Table21</a> <a href="#">Table26</a> <a href="#">Table28</a> <a href="#">Table32</a>	39/Video Capture	Device Behavior of Photo Mode	VCAP_CST_HARDTTOTALS_ONLY was a typo. It should be VCAP_CST_HARDTTOTALS_ONLY	OPOS-J / Tad Furuhata	Current description was VCAP_CST_HARDTTOTALS_ONLY	It was corrected as follows. VCAP_CST_HARDTTOTALS_ONLY	Resolved	
Issue22 <a href="#">Table22</a>	39/Video Capture	Device Behavior of Video Mode	The <b>clearInput</b> method related description was not good enough. The word of "input" should be changed to "data".	OPOS-J / Tad Furuhata	Current description was as follows. "All enqueued input may be deleted by calling <b>clearInput</b> method. See the <b>clearInput</b> method description for more details.	It was changed as follows. "All enqueued data may be deleted by calling <b>clearInput</b> method. See the <b>clearInput</b> method description for more details."	Resolved	
Issue23 <a href="#">Table23</a> <a href="#">Table29</a>	39/Video Capture	Device Behavior of Photo Mode Device Behavior of Video Mode	In the device behavior description ER_CONTINUEINPUT was expressed. Since this is device input behavior during the storage, this should be replaced by ER_RETRY.	OPOS-J / Tad Furuhata	Current description was, "If <b>ErrorEvent</b> response is ER_CONTINUEINPUT, the process of input processing continues."	This was corrected as follows. "If <b>ErrorEvent</b> response is ER_RETRY, the process of recorded data storing was retried."	Resolved	
Issue24 <a href="#">Table24</a> <a href="#">Table30</a>	39/Video Capture	Device Behavior of Photo Mode Device Behavior of Video Mode	ER_CLEAR description was changed. Since this is buffered data clearing and related to the <b>takePhoto</b> method.	OPOS-J / Tad Furuhata	Current description was, "If <b>ErrorEvent</b> is ER_CLEAR, the input processing process is terminated and the taking photo is discarded.	It was changed as follows, "If <b>ErrorEvent</b> is ER_CLEAR, all of the device buffered data is cleared and the <b>takePhoto</b> method is discarded."	Resolved	
Issue25 <a href="#">Table25</a>	39/Video Capture	Device Behavior of Video Mode	The <b>startVideo</b> method behavior was not good enough. The <b>startVideo</b> method detailed behavior with the relation of <b>StatusUpdateEvent</b> and <b>RemainingRecordingTimeInSec</b> property were added.	OPOS-J / Tad Furuhata	Current description was only, "It starts video image capturing by executing the <b>startVideo</b> method."	To explain the startVideo method behavior this description was added. "This method is executed synchronously. During video image capturing, recorded data storing is processed asynchronously and when the start and the end device state is changed, <b>StatusUpdateEvents</b> are delivered to the application. In addition, remaining device recording time is updated in the <b>RemainingRecordingTimeInSec</b> property."	Resolved	
Issue27 <a href="#">Table27</a>	39/Video Capture	Device Behavior of Video Mode	To explain, the <b>stopVideo</b> method and <b>clearInput</b> method related behavior, added the <b>startVideo</b> method related information.	OPOS-J / Tad Furuhata	Current description was as follows. "The video capturing ends after the specified time has elapsed or when <b>stopVideo</b> method is called or when <b>clearInput</b> method is called"	Added the startVideo method related information and changed as follows. "The video capturing ends after the specified time has elapsed or when <b>stopVideo</b> method is called or when <b>clearInput</b> method is called even <b>startVideo</b> method is called."	Resolved	
Issue31 <a href="#">Table31</a>	39/Video Capture	Device Behavior of Video Mode	<b>ErrorEvent</b> description was corrected since this device handling the data not a simple input.	OPOS-J / Tad Furuhata	Current <b>ErrorEvent</b> description was as follows. "An <b>ErrorEvent</b> event (or events) is enqueued if an error occurs while gathering or processing input."	They were changed as follows. "An <b>ErrorEvent</b> event (or events) is enqueued if an error occurs while gathering or processing the data."	Resolved	
Issue33 <a href="#">Table33</a>	39/Video Capture	Device Behavior of Video Mode	The <b>clearInput</b> method related description was corrected since this device handling the data not a simple input.	OPOS-J / Tad Furuhata	Current <b>clearInput</b> method related description was as follows. "All enqueued input may be deleted by calling <b>clearInput</b> method. See the <b>clearInput</b> method description for more details."	They were changed as follows. "All enqueued data may be deleted by calling <b>clearInput</b> method. See the <b>clearInput</b> method description for more details."	Resolved	
Issue34 <a href="#">Table34</a> <a href="#">Table35</a> <a href="#">Table36</a> <a href="#">Table37</a> <a href="#">Table38</a>	39/Video Capture	<b>AutoExposure</b> Property <b>AutoFocus</b> Property <b>AutoGain</b> Property	This property needs to be used under the <b>open</b> , <b>claim</b> , <b>enable</b> . However, description missing the <b>claim</b> and <b>enable</b> .	OPOS-J / Tad Furuhata	In the access after section it was <b>open</b> only.	In the access after section it was changed <b>open-claim-enable</b> .	Resolved	

## UPOS Ver1.16 RCSD Specification

		AutoWhiteBalance Property Brightness property						
Issue39 <a href="#">Table39</a>	39/Video Capture	CapAssociatedHardTotalsDevice Property	Capability needs to be used under the <b>read-only</b> . However description was <b>read-write</b> and this is incorrect. In <b>Remarks</b> section to explain the data handling by using the CapStorage property, several descriptions were added.	Diebold/Dennis	Capability description was <b>read-write</b> . <b>Remarks</b> section description was as follows. “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 <b>CapStorage</b> is either VCAP_CST_ALL or VCAP_CST_HARDTOTALS_ONLY. If <b>CapStorage</b> is VCAP_CST_HOST_ONLY this property value must be the empty string.”	Capability description to be used as the capability was corrected <b>read-only</b> . <b>Remarks</b> section description was corrected as follows. “Indicate that the device is able to store the recorded data into the Associated Hard Totals device and holds its open name, 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 <b>CapStorage</b> is VCAP_CST_HOST_ONLY, the device is not able to store the data into the Associated Hard Totals device and this property value must be the empty string. This property is initialized by the <b>open</b> method.”	Resolved	
Issue40 <a href="#">Table40</a>	39/Video Capture	CapPhotoType Property	This property description’s <b>See Also</b> section missing the reference property name of <b>PhotoType</b> property.	OPOS-J / Tad Furuhashi	Current <b>See Also</b> section did not describe the <b>PhotoType</b> property as the referencing property.	Added the referencing property name in the <b>See Also</b> section. <b>See Also</b> : <b>PhotoType</b> Property.	Resolved	
Issue41 <a href="#">Table41</a>	39/Video Capture	CapStorage Property	This property description’s <b>Remarks</b> did have the incorrect description. Current description was sound data and this was replaced as video or photo data since this device handling video or photo.	OPOS-J / Tad Furuhashi	Current description was as follows. “This is an enumeration and announces where the device is able to write the recorded sound data file to.”	It was changed as follows. “This is an enumeration and announces where the device is able to write the recorded video or photo data file to.”	Resolved	
Issue42 <a href="#">Table42</a> <a href="#">Table43</a> <a href="#">Table44</a> <a href="#">Table45</a> <a href="#">Table46</a>	39/Video Capture	Contrast Property Exposure Property Gain Property HorizontalFlip Property Hue Property	This property needs to be used under the <b>open, claim enable</b> . However it was described <b>open</b> only.	OPOS-J / Tad Furuhashi	Current description was as follows. In the access after section description was <b>open</b> only.	In thee access after section it was corrected as <b>open-claim-enable</b> .	Resolved	
Issue47 <a href="#">Table47</a>	39/Video Capture	PhotoColorSpace Property	This property needs to be used under the <b>open, claim enable</b> . However it was described <b>open</b> only. The <b>Remarks</b> section was incorrect and need to change. Need the explanation with the relation of <b>takePhoto</b> method, <b>VideoCaptureMode</b> property and <b>CapPhoto</b> property values. Also, in the <b>See also</b> section several properties and methods are missing. They are <b>takePhoto</b> method, <b>VideoCaptureMode</b> property, <b>CapPhoto</b> property.	OPOS-J / Tad Furuhashi	Current description was as follows. In the access after section description was <b>open</b> only. Currently in the <b>Remarks</b> section description was as follows, “Indicates the photo color space ID of the frame data to be acquired by the Video Capture Device, if <b>CapPhotoColorSpace</b> property is true. Valid values are one of the values listed in the <b>CapPhotoColorSpaceList</b> property. This property is initialized by the open method.” In the <b>See also</b> section, <b>takePhoto</b> method, <b>CapPhoto</b> property were missing.	In thee access after section it was corrected as <b>open-claim-enable</b> . <b>Remarks</b> is changed as follows. “Indicates the photo color space ID of the frame data to be acquired by the Video Capture Device, if <b>CapPhotoColorSpace</b> property is true and it is used <b>takePhoto</b> method. Valid values are one of the values listed in the <b>CapPhotoColorSpaceList</b> property. This property is referred to when <b>VideoCaptureMode</b> property value is VCAP_VCMODE_PHOTO and <b>CapPhoto</b> is true. This property is initialized by the <b>open</b> method.” In the <b>See also</b> section added the missing property and method were added. They are, <b>CapPhoto</b> property, <b>takePhoto</b> method.	Resolved	

## UPOS Ver1.16 RCSD Specification

Issue48 <a href="#">Table48</a>	39/Video Capture	<b>PhotoColorSpace List</b> Property	In <b>Remarks</b> , there was typo of Photo Color space, this should be Photo color space. In the Color space ID parameter description was changed. To make clear this parameter, added the RGB Depth of 16 bit, 32 bit examples. In the <b>See also</b> section, <b>CapPhotoColorSpace</b> Property was missing.	OPOS-J / Tad Furu-hata	In <b>Remarks</b> , Photo Color space this is the typo of Photo color space. In the Color space ID parameter, RGB depth description was currently, "ID for identifying the color space of RGB, YUV422, etc. And they are indicating like "RGB;YUV422;....." In <b>See also</b> section, <b>CapPhotoColorSpace</b> property was missing.	In <b>Remarks</b> , Photo Color space type was changed to Photo color space. Color space ID parameter description was changed as follows. "ID for identifying the color space of RGB, YUV422, etc. Then if RGB Depth was 16 bits, YUV422 Depth was 32 bits, they are indicating like "RGB:16, YUV422:32, ....." <b>See also</b> section added missing property, <b>CapPhotoColorSpace</b> property.	Resolved	
Issue49 <a href="#">Table49</a>	39/Video Capture	<b>PhotoFrameRate</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec <b>claim and enable</b> were missing. In <b>Remarks</b> , the device behavior description needs to change. Need the explanation with the relation of <b>takePhoto</b> method, <b>VideoCaptureMode</b> property and <b>CapPhoto</b> property values. In the <b>See also</b> section, <b>CapPhoto</b> Property and <b>takePhoto</b> method were missing.	OPOS-J / Tad Furu-hata	In the access after section it was <b>open</b> only. In <b>Remarks</b> , device behavior section was as follows. "Indicates the frame rate of frame data to be acquired by the Video Capture Device. Valid values range from 1 to <b>PhotoMaxFrameRate</b> property." In the <b>See also</b> section, <b>CapPhoto</b> Property and <b>takePhoto</b> method were missing.	In the access after section was changed as <b>open-claim-enable</b> . In <b>Remarks</b> , Photo Color space wording was changed as follows. "Indicates the frame rate of frame data recorded by the Video Capture Device and the photo image capturing and recorded with the <b>takePhoto</b> method. This property is only applied when VCAP_VCMODE_PHOTO is set in <b>VideoCaptureMode</b> property. Valid values range from 1 to <b>PhotoMaxFrameRate</b> property and <b>CapPhoto</b> property is true." In <b>See also</b> section added the <b>CapPhoto</b> property and <b>takePhoto</b> method.	Resolved	
Issue50 <a href="#">Table50</a>	39/Video Capture	<b>PhotoResolution</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec <b>claim and enable</b> were missing. In <b>Remarks</b> , <b>VideoCaptureMode</b> property and <b>CapPhoto</b> property value related descriptions were missing. In the <b>See also</b> section, <b>CapPhoto</b> Property was missing.	OPOS-J / Tad Furu-hata	In the access after section it was <b>open</b> only. In <b>Remarks</b> there was a description as follows. "It shows the resolution of the frame data acquired by the Video Capture Device and the photo taken and recorded with the <b>takePhoto</b> method. Valid values are one of those listed in <b>PhotoResolutionList</b> property. This property is initialized by the <b>open</b> method." In the <b>See also</b> section, <b>CapPhoto</b> Property was missing.	In the access after section was changed as <b>open-claim-enable</b> . In <b>Remarks</b> it was changed as follows. "It shows the resolution of the frame data acquired by the Video Capture Device and the photo taken and recorded with the <b>takePhoto</b> method. Valid values are one of those listed in <b>PhotoResolutionList</b> property. This property is only applied when <b>VideoCaptureMode</b> property is set to VCAP_VCMODE_PHOTO and if <b>CapPhoto</b> is true. This property is initialized by the <b>open</b> method." In <b>See also</b> section added the <b>CapPhoto</b> property.	Resolved	
Issue51 <a href="#">Table51</a>	39/Video Capture	<b>PhotoResolution List</b> Property	In the <b>See also</b> section, <b>CapPhotoResolution</b> Property was missing.	OPOS-J / Tad Furu-hata	In the <b>See also</b> section, <b>CapPhotoResolution</b> Property was missing.	In <b>See also</b> section added the <b>CapPhotoResolution</b> property.	Resolved	
Issue52 <a href="#">Table52</a>	39/Video Capture	<b>PhotoType</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec <b>claim and enable</b> were missing. In <b>Remarks</b> , there was a name of <b>CapPhotoTypeList</b> property however it is not existing and need change to <b>PhotoTypeList</b> property. In the <b>See also</b> section, <b>CapPhoto</b> Property was missing. And <b>CapPhotoTypeList</b> property	OPOS-J / Tad Furu-hata	In the access after section it was <b>open</b> only. Currently in <b>Remarks</b> description was as follows. "Indicates the data format of photo taken with the <b>takePhoto</b> method. Valid values are one of the values listed in the <b>CapPhotoTypeList</b> property. This property is initialized by the <b>open</b> method." In the <b>See also</b> section, <b>CapPhoto</b> Property was missing. Also, <b>CapPhotoTypeList</b> property need to be corrected as <b>PhotoTypeList</b> property.	In the access after section was changed as <b>open-claim-enable</b> . In <b>Remarks</b> it was changed as follows. "Indicates the data format of photo taken with the <b>takePhoto</b> method. Valid values are one of the values listed in the <b>PhotoTypeList</b> property. This property is initialized by the <b>open</b> method." , <b>CapPhotoTypeList</b> property was corrected as <b>PhotoTypeList</b> property. In <b>See also</b> section added the <b>CapPhoto</b> property and correct the <b>CapPhotoTypeList</b> property to <b>PhotoTypeList</b> property.	Resolved	

## UPOS Ver1.16 RCSD Specification

			need to be corrected to <b>PhotoTypeList</b> property.					
Issue53 <a href="#">Table53</a>	39/Video Capture	<b>Saturation</b> Property	This property needs to be used under the <b>open, claim enable</b> . However it was described <b>open</b> only.	OPOS-J / Tad Furuhata	Current description was as follows. In the access after section description was <b>open</b> only.	In thee access after section it was corrected as <b>open-claim-enable</b> .	Resolved	
Issue54 <a href="#">Table54</a>	39/Video Capture	<b>Storage</b> Property	The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	This property is able to read and write. In that case there should be the E_ILLEGAL error Code.	E_ILLEGAL related description was added.  Some possible values of the exception's ErrorCode property are: <b>Value      Meaning</b> E_ILLEGAL    An invalid value was specified or recording is ongoing.	Resolved	
Issue55 <a href="#">Table55</a>	39/Video Capture	<b>VerticalFlip</b> Property	This property needs to be used under the <b>open, claim enable</b> . However it was described <b>open</b> only.	OPOS-J / Tad Furuhata	Current description was as follows. In the access after section description was <b>open</b> only.	In thee access after section it was corrected as <b>open-claim-enable</b> .	Resolved	
Issue56 <a href="#">Table56</a>	39/Video Capture	<b>VideoCaptureMode</b> Property	This property needs to be used under the <b>open, claim enable</b> . However it was described <b>open</b> only. The E_ILLEGAL error Code was missing. In <b>See also</b> section there was a <b>VideoCaptureMode</b> property wording, since this is the <b>VideoCaptureMode</b> property section therefore it is eliminated.	OPOS-J / Tad Furuhata	Current description was as follows. In the access after section description was <b>open</b> only. This property is able to read and write. In that case there should be the E_ILLEGAL error Code. In the <b>See also</b> section there is the word of <b>VideoCaptureMode</b> property.	In thee access after section it was corrected as <b>open-claim-enable</b> . E_ILLEGAL related description was added.  Some possible values of the exception's ErrorCode property are: <b>Value      Meaning</b> E_ILLEGAL    An invalid value was specified.  In the <b>See also</b> section the word of <b>VideoCaptureMode</b> property was eliminated.	Resolved	
Issue57 <a href="#">Table57</a>	39/Video Capture	<b>VideoColorSpace</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec <b>claim</b> and <b>enable</b> were missing. In <b>Remarks</b> , color space ID description is not good enough need to change. Need to explain frame data is acquired by Video Capture Device and should be used under the <b>CapVideoColorSpace</b> property value is true. In the <b>See also</b> section, <b>CapVideoColorSpace</b> Property was missing.	OPOS-J / Tad Furuhata	In the access after section it was <b>open</b> only. In <b>Remarks</b> , current color space ID related description was as follows. "Indicates the video color space ID of the frame data to be used by <b>startVideo</b> method." In the <b>See also</b> section, <b>CapVideoColorSpace</b> property was missing.	In the access after section was changed as <b>open-claim-enable</b> . In <b>Remarks</b> , color space ID related description was corrected as follows. "Indicates the video color space ID of the frame data to be acquired by the Video Capture Device, if <b>CapVideoColorSpace</b> property is true and it is used by <b>startVideo</b> method." In <b>See also</b> section added the <b>CapVideoColorSpace</b> property.	Resolved	
Issue58 <a href="#">Table58</a>	39/Video Capture	<b>VideoColorSpace List</b> Property	In <b>Remarks</b> , Color space ID parameter related RGB depth description need to change. In the Color space ID parameter description was changed. To make clear this parameter added the RGB Depth of 16 bit, 32 bit examples. In the <b>See also</b> section, <b>VideoCaptureMode</b> property	OPOS-J / Tad Furuhata	In <b>Remarks</b> , Color space ID related RGB depth description was as follows. "ID for identifying the color space of RGB, YUV422, etc. And they are indicating like "RGB;YUV422;....." In the <b>See also</b> section, <b>VideoCaptureMode</b> property was missing.	In <b>Remarks</b> , Color space ID related RGB depth description was changed as follows. "ID for identifying the color space of RGB, YUV422, etc. Then if RGB Depth was 16 bits, YUV422 Depth was 32 bits, they are indicating like "RGB:16, YUV422:32, ....." <b>See also</b> section added missing <b>VideoCaptureMode</b> property.	Resolved	

## UPOS Ver1.16 RCSD Specification

			was missing,					
Issue59 <a href="#">Table59</a>	39/Video Capture	<b>VideoFrameRate</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec <b>claim</b> and <b>enable</b> were missing. In the <b>See also</b> section, <b>CapVideoFrameRate</b> Property was missing.	OPOS-J / Tad Furuhata	In the access after section it was <b>open</b> only. In the <b>See also</b> section, <b>CapVideoFrameRate</b> Property was missing.	In the access after section was changed as <b>open-claim-enable</b> . In <b>See also</b> section added the missing <b>CapVideoFrameRate</b> property.	Resolved	
Issue60 <a href="#">Table60</a>	39/Video Capture	<b>VideoMaxFrameRate</b> Property	In the <b>See also</b> section, <b>VideoCaptureMode</b> Property was missing.	OPOS-J / Tad Furuhata	In the <b>See also</b> section, <b>VideoCaptureMode</b> property was missing.	In <b>See also</b> section added the missing <b>VideoCaptureMode</b> property.	Resolved	
Issue61 <a href="#">Table61</a> <a href="#">Table62</a>	39/Video Capture	<b>VideoResolution</b> Property <b>VideoType</b> Property	This property needs to be used under the <b>open, claim enable</b> , however in the spec <b>claim</b> and <b>enable</b> were missing.	OPOS-J / Tad Furuhata	In the access after section it was <b>open</b> only.	In the access after section was changed as <b>open-claim-enable</b> .	Resolved	
Issue63 <a href="#">Table63</a>	39/Video Capture	<b>VideoTypeList</b> Property	In <b>Remarks</b> section list of image wording missing the video wording therefore should be corrected from “list of image format” to “list of video image format”. In the <b>See also</b> section, <b>VideoCaptureMode</b> Property was missing.	OPOS-J / Tad Furuhata	In <b>Remarks</b> current wording was “list of image format”. Currently it was described as follows. “A comma-separated list of video image format values that can be set for the <b>VideoType</b> property.” In the <b>See also</b> section, <b>VideoCaptureMode</b> Property was missing.	In <b>Remarks</b> was changed as follows. “A comma-separated list of video image format values that can be set for the <b>VideoType</b> property.” In <b>See also</b> section, added the missing <b>VideoCaptureMode</b> property.	Resolved	
Issue64 <a href="#">Table64</a>	39/Video Capture	<b>startVideo</b> Method	In the <b>Remarks</b> , <b>StatusUpdateEvent</b> related description was not precise therefore, it was changed. Since current one explaining the power state however this time it will deliver the start and stop of the video state. In the <b>Errors</b> section, in the E_ILLEGAL related description missing the <b>CapVideo</b> property value condition.	Diebold/ Dennis	In the <b>Remarks</b> , <b>StatusUpdateEvent</b> related description was mentioned power state of the device, but in this device <b>StatusUpdateEvent</b> will not notify the power state but will do the start and stop state of the video. Therefore this description was eliminated. “ <b>StatusUpdateEvent</b> will notify the application that there is a change in the power status or a state change during video capturing and recording.” In the <b>Errors</b> section, in the E_ILLEGAL related description missing the <b>CapVideo</b> property condition.	In the <b>Remarks</b> , <b>StatusUpdateEvent</b> related description as follows was added instead of eliminated description. “During the video image capturing, the recorded data storing is processed asynchronously and when the start and stop state are changed, <b>StatusUpdateEvents</b> are delivered to the application.” In the <b>Errors</b> section, in the E_ILLEGAL related missing <b>CapVideo</b> property wording was added after description of “VCAP_VCMODE_VIDEO” and it is “and <b>CapVideo</b> is not true.”	Resolved	
Issue65 <a href="#">Table65</a>	39/Video Capture	<b>stopVideo</b> Method	In the <b>Remarks</b> , <b>StatusUpdateEvent</b> related description was changed. Since it is notifying the start or stop state of video capturing or recording.	Diebold/ Dennis	In the <b>Remarks</b> , current description was as follows. “ <b>StatusUpdateEvent</b> will notify the application that there is a change in the power status or a state change during taking video and recording.”	In the <b>Remarks</b> , changed the description as follows. “ <b>StatusUpdateEvent</b> is delivered to notify the application that the device video capturing and recording were stopped.”	Resolved	
Issue66 <a href="#">Table66</a>	39/Video Capture	<b>takePhoto</b> Method	In the <b>Remarks</b> , <b>StatusUpdateEvent</b> related description was missing the function of its notifying the start and stop state. Therefore this related description was added.	Diebold/ Dennis	In the <b>Remarks</b> , <b>StatusUpdateEvent</b> related description was as follows. “Take photo and record with setting contents of <b>PhotoColorSpace</b> property, <b>PhotoResolution</b> property, <b>PhotoFrameRate</b> Property and <b>PhotoType</b> property. Before calling this method, it needs to set the <b>VideoCaptureMode</b> property to VCAP_VCMODE_PHOTO and this method can be executed if <b>CapPhoto</b> property is true.	In the <b>Remarks</b> , added the <b>StatusUpdateEvent</b> related description and changed as follows. “Take photo and record with setting contents of <b>PhotoColorSpace</b> property, <b>PhotoResolution</b> property, <b>PhotoFrameRate</b> Property and <b>PhotoType</b> property. Before calling this method, it needs to set the <b>VideoCaptureMode</b> property to VCAP_VCMODE_PHOTO and this method can be executed if <b>CapPhoto</b> property is true. This method	Resolved	



UPOS Ver1.16 RCSD Specification

					This method is executed synchronously. The location where photo files are recorded is controlled through the <b>Storage</b> Property. The timeout specifies the number of milliseconds”	is executed synchronously. The process of recorded data storing is performed asynchronously. <b>StatusUpdateEvents</b> are delivered to the application when the start and the end states were changed. The location where photo files are recorded is controlled through the <b>Storage</b> Property. The timeout specifies the number of milliseconds”		
Issue67 <a href="#">Table67</a>	39/Video Capture	<b>ErrorEvent</b>	<p>In the <i>ErrorLocus</i> section EL_OUTPUT should be corrected since cancelling method is <b>clearInput</b> method.</p> <p>In the <i>ErrorLocus</i> attribute EL_OUTPUT should be corrected to EL_INPUT, since this is device input behavior. And ER_RETRY and ER_CLEAR description was corrected since there is no EL_OUTPUT since this is device input behavior.</p> <p>In the <b>See also</b> section, “<b>Device Output Models</b>” on page Intro-25 was eliminated. Instead “<b>Device information Reporting Model</b>” on Page Intro-30 was added since this device behavior is along with this model.</p>	Diebold/ Dennis	<p>In the <i>Error Locus</i> section, there was a description as follows. <u>Attributes Type Description</u> ErrorLocus int32 Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.</p> <p>In the ErrorLocus attribute, EL_OUTPUT was changed to EL_INPUT since this is device input behavior.</p> <p>And current description was as follows. The ErrorLocus attribute has one of the following values: <u>Value Meaning</u> EL_OUTPUT Error occurred while processing asynchronous output.</p> <p>And current ErrorResponse description was as follows. <u>Value Meaning</u> 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 retried, and the error state is exited. This is the default response when the locus is EL_OUTPUT.</p> <p>ER_CLEAR Valid for loci: EL_OUTPUT. Clear all buffered input or output data (including all asynchronous output). The error state is exited</p> <p>In the <b>See also</b> section, “<b>Device Output Models</b>” on PageIntro-25 was there. This is incorrect.</p>	<p>In the <i>Error Locus</i> section, EL_OUTPUT related description was changed to EL_INPUT and changed as follows. . <u>Attributes Type Description</u> ErrorLocus int32 Location of the error. If EL_INPUT is specified. An error occurred during asynchronous input.</p> <p>In the ErrorLocus attribute, EL_OUTPUT was changed to EL_INPUT since this is device input behavior.</p> <p>And description changed as follows. The ErrorLocus attribute has the following value: <u>Value Meaning</u> EL_INPUT Error occurred while processing asynchronous input.</p> <p>And current ErrorResponse description was changed based on replacing the EL_OUTPUT to the EL_INPUT and described as follows. <u>Value Meaning</u> ER_RETRY Retry recorded data storing. The error state is exited. Typically, valid for asynchronous recorded data storing when the locus is EL_INPUT, in which case the asynchronous recorded data storing is retried, and the error state is exited. This is the default response.</p> <p>ER_CLEAR Clear all buffered captured or stored data. The error state is exited</p> <p>In the <b>See also</b> section, “<b>Device Output Models</b>” on Page Intro-25 was eliminated and “<b>Device information Reporting Model</b> “ on Page Intro-30” was added since this device behavior is along with this model.</p>	Resolved	
Issue69 <a href="#">Table69</a> <a href="#">Table78</a>	40/Indi. Recognition 41/Sound Recorder 42/Voice Recog.	Summary <b>clearInputProper</b> <b>ties Method</b>	This method description was missing the method behavior therefore it is added.	OPOS-J / Tad Furuhata	Current description was as follows. Method name <b>void: {}</b>	Method behavior description was changed as follows. Method name <b>void: {raises-exception, use after</b>	Resolved	

**UPOS Ver1.16 RCSD Specification**

<a href="#">Table91</a> <a href="#">Table95</a> <a href="#">Table102</a> <a href="#">Table107</a> <a href="#">Table118</a> <a href="#">Table127</a>	43/Sound Player 44/Speech Synthe. 45/Gesture Control 46/Device Monitor 47/Graphic Display					open, claim}																										
Issue75 <a href="#">Table75</a>	40/Indi. Recognition	<b>IndividualRecognitionFilter</b> Property	This property needs to be used under the <b>open, claim enable</b> . However it was described <b>open</b> only. The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	Current description was as follows. In the access after section description was <b>open</b> only. This property is able to read and write. In that case there should be the E_ILLEGAL error Code and its related description.	In thee access after section it was corrected as <b>open-claim-enable</b> . E_ILLEGAL related description was added.  Some possible values of the exception’s ErrorCode property are: <table border="1"> <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.	Resolved																					
Value	Meaning																															
E_ILLEGAL	An invalid value was specified.																															
Issue76 <a href="#">Table76</a>	40/Indi. Recognition	<b>ErrorEvent</b>	In the <i>ErrorLocus</i> section EL_OUTPUT should be eliminated since this is input device. Also added the description in there “See values below.” to refer to the ErrorLocus attribute. In the <i>ErrorLocus</i> attribute EL_OUTPUT is eliminated since this is input device. And ER_RETRY was deleted because of there is no RETRY. ER_CLEAR description was corrected since there is no EL_OUTPUT. In the <b>See also</b> section, “ <b>Device Output Models</b> ” on page Intro-25 was eliminated since this is input device.	Diebold/ Dennis	In the <i>Error Locus</i> section, there was a description as follows. <table border="1"> <thead> <tr> <th>Attributes</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ErrorLocus</td> <td>int32</td> <td>Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.</td> </tr> </tbody> </table> In the ErrorLocus attribute, EL_OUTPUT was existing. However this must be eliminated since this is input device.  And current description was as follows. The ErrorLocus attribute has one of the following values: <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>EL_OUTPUT</td> <td>Error occurred while processing asynchronous output.</td> </tr> </tbody> </table> And current ErrorResponse description was as follows. <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>ER_RETRY</td> <td>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.</td> </tr> </tbody> </table>	Attributes	Type	Description	ErrorLocus	int32	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.	Value	Meaning	EL_OUTPUT	Error occurred while processing asynchronous output.	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.	In the <i>Error Locus</i> section, EL_OUTPUT related description was eliminated and described as follows. . <table border="1"> <thead> <tr> <th>Attributes</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ErrorLocus</td> <td>int32</td> <td>Location of the error. See values below.</td> </tr> </tbody> </table> In the ErrorLocus attribute, EL_OUTPUT was eliminated completely.  And current ErrorResponse description was changed as follows. ER_RETRY was eliminated since there is no RETRY. ER_CLEAR was changed as follows. <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>ER_CLEAR</td> <td>Valid for all locus: EL_INPUT, 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.</td> </tr> </tbody> </table> In the <b>See also</b> section, “ <b>Device Output Models</b> ” on Page Intro-25 was eliminated since this is input device.	Attributes	Type	Description	ErrorLocus	int32	Location of the error. See values below.	Value	Meaning	ER_CLEAR	Valid for all locus: EL_INPUT, 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.	Resolved	
Attributes	Type	Description																														
ErrorLocus	int32	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.																														
Value	Meaning																															
EL_OUTPUT	Error occurred while processing asynchronous output.																															
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.																															
Attributes	Type	Description																														
ErrorLocus	int32	Location of the error. See values below.																														
Value	Meaning																															
ER_CLEAR	Valid for all locus: EL_INPUT, 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.																															

**UPOS Ver1.16 RCSD Specification**

					<p>ER_CLEAR Valid for 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.</p> <p>In the <b>See also</b> section, “<b>Device Output Models</b>” on PageIntro-25 was there. This is incorrect.</p>			
<p>Issue84 <a href="#">Table84</a> <a href="#">Table96</a> <a href="#">Table137</a></p>	<p>41/Sound Recorder 43/Sound Player 47/Graphic Display</p>	<p><b>CapAssociatedHardTotalsDevice</b> Property</p>	<p>Capability needs to be used under the <b>read-only</b>. However description was <b>read-write</b> and this is incorrect. In addition, <b>Remarks</b> section initialization description was missing.</p>	<p>OPOS-J / Tad Furuhata</p>	<p>Capability description was <b>read-write</b>. In <b>Remarks</b> section initialization description was missing.</p>	<p>Capability description was corrected <b>read-only</b>. In <b>Remarks</b> section initialization description was added as follows. “This property is initialized by the <b>open</b> method.”</p>	<p>Resolved</p>	
<p>Issue85 <a href="#">Table85</a></p>	<p>41/Sound Recorder</p>	<p><b>Channel</b> Property</p>	<p>The E_ILLEGAL error Code was missing.</p>	<p>OPOS-J / Tad Furuhata</p>	<p>This property is able to read and write. In that case there should be the E_ILLEGAL error Code.</p>	<p>E_ILLEGAL related description was added. Some possible values of the exception’s ErrorCode property are: <u><b>Value</b>    <b>Meaning</b></u> E_ILLEGAL    An invalid value was specified.</p>	<p>Resolved</p>	
<p>Issue86 <a href="#">Table86</a></p>	<p>41/Sound Recorder</p>	<p><b>SamplingRate</b> Property</p>	<p>The E_ILLEGAL error Code was missing.</p>	<p>OPOS-J / Tad Furuhata</p>	<p>This property is able to read and write. In that case there should be the E_ILLEGAL error Code.</p>	<p>E_ILLEGAL related description was added. Some possible values of the exception’s ErrorCode property are: <u><b>Value</b>    <b>Meaning</b></u> E_ILLEGAL    An invalid value was specified.</p>	<p>Resolved</p>	
<p>Issue87 <a href="#">Table87</a></p>	<p>41/Sound Recorder</p>	<p><b>SoundType</b> Property</p>	<p>This property needs to be used under the <b>open, claim enable</b>. However it was missing the enable. The E_ILLEGAL error Code was missing.</p>	<p>OPOS-J / Tad Furuhata</p>	<p>Current description was as follows. In the access after section description was <b>open-claim</b>. This property is able to read and write. In that case there should be the E_ILLEGAL error Code.</p>	<p>In thee access after section it was corrected as <b>open-claim-enable</b>. E_ILLEGAL related description was added. Some possible values of the exception’s ErrorCode property are: <u><b>Value</b>    <b>Meaning</b></u> E_ILLEGAL    An invalid value was specified.</p>	<p>Resolved</p>	
<p>Issue88 <a href="#">Table88</a></p>	<p>41/Sound Recorder</p>	<p><b>ErrorEvent</b></p>	<p>In the <b>See also</b> section, <b>Status, Error code, State model</b> description was not needed, since they were not referenced. Then “<b>Device Input Model</b>” on page Intro-22” and “<b>Error Handling</b>” on page Intro-23 were added instead.</p>	<p>OPOS-J / Tad Furuhata</p>	<p>In the <b>See also</b> section, <b>Status, Error code, State model</b> are not needed, since they are not referenced. “<b>Device Input Model</b>” on page Intro-22” and “<b>Error Handling</b>” on page Intro-23 were missing.</p>	<p>In the <b>See also</b> section, <b>Status, Error code, State model</b> are eliminated. “<b>Device Input Model</b>” on page Intro-22” and “<b>Error Handling</b>” on page Intro-23 were added.</p>	<p>Resolved.</p>	
<p>Issue92 <a href="#">Table92</a></p>	<p>42/Voice Recog.</p>	<p><b>ErrorEvent</b></p>	<p>In the <i>ErrorLocus</i> section EL_OUTPUT should be eliminated since this is input device. Also added the description in there “See values below.” to refer to the ErrorLocus attribute. In the <i>ErrorLocus</i> attribute</p>	<p>Diebold/ Dennis</p>	<p>In the <i>Error Locus</i> section, there was a description as follows. <u><b>Attributes Type Description</b></u> ErrorLocus int32 Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.</p>	<p>In the <i>Error Locus</i> section, EL_OUTPUT related description was eliminated and described as follows. . <u><b>Attributes Type Description</b></u> ErrorLocus int32 Location of the error. See values below.</p>	<p>Resolved</p>	



**UPOS Ver1.16 RCSD Specification**

			<p>EL_OUTPUT is eliminated since this is input device.          And ER_RETRY was deleted because of there is no RETRY.          ER_CLEAR description was corrected since there is no EL_OUTPUT.          In the See also section, “<b>Device Output Models</b>” on page Intro-25 was eliminated since this is input device.</p>		<p>In the ErrorLocus attribute, EL_OUTPUT was existing. However this must be eliminated since this is input device.          And current description was as follows.          The ErrorLocus attribute has one of the following values:</p> <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>EL_OUTPUT</td> <td>Error occurred while processing asynchronous output.</td> </tr> </tbody> </table> <p>And current ErrorResponse description was as follows.</p> <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>ER_RETRY</td> <td>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.</td> </tr> <tr> <td>ER_CLEAR</td> <td>Valid for 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.</td> </tr> </tbody> </table> <p>In the <b>See also</b> section, “<b>Device Output Models</b>” on PageIntro-25 was there. This is incorrect.</p>	<u>Value</u>	<u>Meaning</u>	EL_OUTPUT	Error occurred while processing asynchronous output.	<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 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 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.	<p>In the ErrorLocus attribute, EL_OUTPUT was eliminated completely.          And current ErrorResponse description was changed as follows. ER_RETRY was eliminated since there is no RETRY. ER_CLEAR was changed as follows.</p> <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>ER_CLEAR</td> <td>Valid for all locus: EL_INPUT, 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.</td> </tr> </tbody> </table> <p>In the <b>See also</b> section, “<b>Device Output Models</b>” on Page Intro-25 was eliminated since this is input device.</p>	<u>Value</u>	<u>Meaning</u>	ER_CLEAR	Valid for all locus: EL_INPUT, 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.		
<u>Value</u>	<u>Meaning</u>																					
EL_OUTPUT	Error occurred while processing asynchronous output.																					
<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 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 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.																					
<u>Value</u>	<u>Meaning</u>																					
ER_CLEAR	Valid for all locus: EL_INPUT, 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.																					
<p>Issue97  <a href="#">Table97</a></p>	43/Sound Player	<b>CapStorage</b> Property	<p>In <b>See also</b> section, referenced properties were missing. They are <b>Storage</b> property and <b>CapAssociatedHardTotalsDevice</b> property.</p>	OPOS-J / Tad Furuata	<p>In <b>See also</b> section several referencing properties were missing.</p>	<p>Added the missing properties in the <b>See also</b> section. They are <b>Storage</b> property, <b>CapAssociatedHardTotalsDevice</b> property.</p>	Resolved															
<p>Issue98  <a href="#">Table98</a></p>	43/Sound Player	<b>Storage</b> Property	<p>The E_ILLEGAL error Code was missing.</p>	OPOS-J / Tad Furuata	<p>This property is able to read and write. In that case there should be the E_ILLEGAL error Code.</p>	<p>E_ILLEGAL related description was added.          Some possible values of the exception’s ErrorCode property are:</p>	Resolved															

**UPOS Ver1.16 RCSD Specification**

						<p><b>Value      Meaning</b></p> <p>E_ILLEGAL    An invalid value was specified or recording is ongoing.</p>		
<p>Issue99  <a href="#">Table99</a>  <a href="#">Table104</a>  <a href="#">Table115</a>  <a href="#">Table144</a></p>	<p>43/Sound Player  44/Speech Synthe.  45/Gesture Control  47/Graphic Display</p>	<p><b>ErrorEvent</b></p>	<p>In the <i>ErrorLocus</i> section EL_OUTPUT related description was changed since this is output device.  In the <i>ErrorLocus</i> attribute EL_OUTPUT related description was changed to fit with the output device behavior.  And ER_RETRY and ER_CLEAR need changes to fit with the output device behavior.</p>	<p>Diebold/  Dennis</p>	<p>In the <i>Error Locus</i> section, there was a description as follows.  <b>Attributes Type Description</b>  ErrorLocus int32 Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.</p> <p>In the ErrorLocus attribute the starting description and following ER_RETRY and ER_CLEAR descriptions were as follows.  The ErrorLocus attribute has one of the following values:  <b>Value      Meaning</b>  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.</p> <p>ER_CLEAR    Valid for loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT. Clear all buffered input or output data (including all asynchronous output).</p>	<p>In the <i>Error Locus</i> section, they are changed as follows since this is output device.  <b>Attributes Type Description</b>  ErrorLocus int32 Location of the error. If EL_OUTPUT is specified. It is indicating that the error occurred while processing asynchronous output.</p> <p>In the ErrorLocus attribute, the starting description and following ER_RETRY and ER_CLEAR descriptions were changed as follows.</p> <p>The ErrorLocus attribute has the following value:  <b>Value      Meaning</b>  ER_RETRY    Retry the asynchronous output. The error state is exited.                    This is the default response.</p> <p>ER_CLEAR    Clear all buffered output data including all asynchronous output. (The effect is the same as when clearOutput method is called.) The error state is exited.</p>	<p>Resolved</p>	
<p>Issue103  <a href="#">Table103</a>  <a href="#">Table108</a>  <a href="#">Table128</a></p>	<p>44/Speech Synthe.  45/Gesture Control  47/Graphic Display</p>	<p>Summary  <b>clearOutput</b>  Method</p>	<p>This method description was missing the method behavior condition. Therefore it is added.</p>	<p>OPOS-J /  Tad  Furuhata</p>	<p>Current description was as follows.  Method name <b>void: {}</b></p>	<p>Method behavior description was changed as follows.  Method name <b>void: {raises-exception, use after open, claim}</b></p>	<p>Resolved</p>	
<p>Issue114  <a href="#">Table114</a></p>	<p>45/Gesture Control</p>	<p><b>CapAssociatedHardTotalsDevice</b>  Property</p>	<p>In <b>Remarks</b> section initialization description was missing.</p>	<p>OPOS-J /  Tad  Furuhata</p>	<p>In <b>Remarks</b> section initialization description was missing.</p>	<p>In <b>Remarks</b> section initialization description was added as follows.  “This property is initialized by the <b>open</b> method.”</p>	<p>Resolved</p>	
<p>Issue124  <a href="#">Table124</a></p>	<p>46/Device Monitor</p>	<p><b>ErrorEvent</b></p>	<p>In ErrorResponse attribute value was incorrect. ER_RETRY value should be eliminated since this is the input device. Also, its value of ER_CLEAR needs the change to</p>	<p>Diebold/  Dennis</p>	<p>The current ErrorLocus attribute values were as follows.  <b>Value      Meaning</b>  ER_RETRY    Retry sending the data. The error state is exited.</p>	<p>The ErrorLocus attribute values will be changed as follows.  ER_RETRY values are eliminated since there is not ER_RETRY. And ER_CLEAR value will be changed as follows.</p>	<p>Resolved</p>	

## UPOS Ver1.16 RCSD Specification

			fit with the input device behavior.		<p>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.</p> <p>ER_CLEAR Valid for 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.</p>	<p><b>Value Meaning</b></p> <p>ER_CLEAR Valid for locus: EL_INPUT, EL_INPUT_DATA. Clear all buffered input data. The error state is exited. This is the default response when the locus is EL_INPUT.</p>		
Issue134 <a href="#">Table134</a>	47/Graphic Display	Model Section Image Display Mode	There was a description duplicated the previous one. Therefore it is eliminated.	OPOS-J / Tad Furuhata	There was a description as follows duplicated previous one. “The <b>ImageTypeList</b> property lists image files that the device can display.”	The duplicated description was eliminated.	Resolved	
Issue135 <a href="#">Table135</a>	47/Graphic Display	Model Section Video Display Mode	There was a description regarding the <b>CapMultiDisplay</b> property. However this property is not existing, therefore its description was eliminated.	OPOS-J / Tad Furuhata	There was a <b>CapMultiDisplay</b> property related description. “To check if the device supports simultaneous playback, check the <b>CapMultiDisplay</b> property”	The <b>CapMultiDisplay</b> property related description was eliminated since it is not existing property.	Resolved	
Issue136 <a href="#">Table136</a>	47/Graphic Display	Model Section Web Display Mode	The <b>cancelLoading</b> method name was incorrect. This should be <b>cancelURLLoading</b> method.	OPOS-J / Tad Furuhata	The <b>cancelLoading</b> method name was existing. However it is incorrect.	Corrected the method name. It is <b>cancelURLLoading</b> method.	Resolved.	
Issue138 <a href="#">Table138</a>	47/Graphic Display	<b>DisplayMode</b> Property	In <b>See also</b> section referencing method and property names are incorrect. <b>loadImage</b> method and <b>playVideo</b> method were missing. <b>CapCaptureColorSpaceList</b> Property, <b>VideoCaptureMode</b> Property were eliminated since they are not referenced.	OPOS-J / Tad Furuhata	In <b>See also</b> section, <b>loadImage</b> method and <b>playVideo</b> method were missing. Not referenced <b>CapCaptureColorSpaceList</b> property and <b>VideoCaptureMode</b> property were existing.	In <b>See also</b> section, <b>loadImage</b> method and <b>playVideo</b> method were added. <b>CapCaptureColorSpaceList</b> property and <b>VideoCaptureMode</b> property were eliminated.	Resolved.	
Issue139 <a href="#">Table139</a>	47/Graphic Display	<b>ImageType</b> Property	The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	This property is able to read and write. In that case there should be the E_ILLEGAL error Code.	E_ILLEGAL related description was added.  Some possible values of the exception’s ErrorCode property are: <b>Value Meaning</b> E_ILLEGAL An invalid value was specified.	Resolved.	
Issue140 <a href="#">Table140</a> -	47/Graphic Display	<b>LoadStatus</b> Property	This property can be accessed after <b>open</b> only.	OPOS-J / Tad Furuhata	As the property access condition, it was described can be access after <b>open, claim, enable</b> .	As the property access condition, it was corrected <b>open</b> only and <b>claim, enable</b> were eliminated.	Resolved.	
Issue141 <a href="#">Table141</a>	47/Graphic Display	<b>VideoType</b> Property	The E_ILLEGAL error Code was missing.	OPOS-J / Tad Furuhata	This property is able to read and write. In that case there should be the E_ILLEGAL error Code.	E_ILLEGAL related description was added.  Some possible values of the exception’s ErrorCode property are: <b>Value Meaning</b> E_ILLEGAL An invalid value was specified.	Resolved.	

## UPOS Ver1.16 RCSD Specification

Issue142 <a href="#">Table142</a>	47/Graphic Display	<b>playVideo</b> Method	In <b>Remarks</b> there was a description of image. However this is the video function method. Therefore, image related information was corrected to the video related information. Regarding <b>DisplayMode</b> property value, there was a description of DGISP. This was typo and corrected as DGSP.	OPOS-J / Tad Furuhata	Current <b>Remarks</b> description was as follows. “Play the specified video that are loaded in the storage area by the <b>loadImage</b> method. All of loaded file images are listed in the <b>ImageTypeList</b> property.” And <b>DisplayMode</b> property values were described with typo as follows. GDISP_DMODE_VIDEO_NORMAL, GDISP_DMODE_VIDEO_FULL.	Remarks description was corrected as follows. “Play the video type file content that is specified using <b>VideoType</b> property. All of the video file values are listed in the <b>VideoTypeList</b> property, if <b>CapVideoType</b> property is true.” And <b>DisplayMode</b> property values were described with the elimination of typo as follows. GDSP_DMODE_VIDEO_NORMAL, GDSP_DMODE_VIDEO_FULL	Resolved.	
Issue143 <a href="#">Table143</a>	47/Graphic Display	<b>stopVideo</b> Method	In <b>Remarks</b> there was a word of image file. However this is the video function handling method. Therefore, image file wording was changed to video file. In <b>See also</b> section, referencing method name was <b>startVideo</b> method was incorrect. It should be <b>playVideo</b> method.	OPOS-J / Tad Furuhata	In <b>Remarks</b> there was a word of Image file loading status. This was corrected to Video file loading status. In <b>See also</b> section <b>startVideo</b> method wording was existing.	In <b>Remarks</b> wording was corrected and Image file loading status was changed to Video file loading status. In <b>See also</b> section <b>startVideo</b> method wording was changed to <b>playVideo</b> method.	Resolved.	
Issue145 <a href="#">Table145</a>	Cover Page	Specification Ver. Submission Date	In the cover page there was an old description. This specification proposing the UPOS 1.16.1 and it edited the UPOS1.16. Also, documentation submission date is Sept. 2021.	OPOS-J / Tad Furuhata	Current description was as follows. Date: April 2020 Unified POS RCSD, v1.16 FTF Beta 1 This specification adds to and extends the UPOS 1.15 specification.	Corrected description was as follows. Date: Nov. 2021 Unified POS RCSD, v1.16.1 This specification adds to and extends the UPOS 1.16 specification.	Resolved.	
Issue146 <a href="#">Table146</a>	39/Video Capture	<b>StatusUpdateEvent</b> value in Video Mode	In the Video Mode description, there was a wording of <b>StatusUpdateEvent</b> value. It was described as VCAP_SUE_END_VIDEO. However it is not correct. And correct value is VCAP_SUE_STOP_VIDEO.	OPOS-J / Tad Furuhata	Current description was described as <b>StatusUpdateEvent</b> value was VCAP_SUE_END_VIDEO.	In this section, <b>StatusUpdateEvent</b> value was corrected as VCAP_SUE_STOP_VIDEO.	Resolved	
Issue147 <a href="#">Table147</a>	40/Individual Recog.	<b>StatusUpdateEvent</b> value	The StatusUpdateEvent values are incorrect. Not required RECORDING was included.	OPOS-J / Tad Furuhata	Currently, they were as follows. <u><b>Value Meaning</b></u> VCAP_SUE_START_VIDEO_RECORDING It will be notified when video recording starts. VCAP_SUE_STOP_VIDEO_RECORDINGIt will be notified when video recording stops.	They were corrected as follows. <u><b>Value Meaning</b></u> VCAP_SUE_START_VIDEO It will be notified when video Recording starts. VCAP_SUE_STOP_VIDEO It will be notified when video recording stops.	Resolved	
Issue148 <a href="#">Table148</a>	40/Individual Recog.	<b>DataEvent</b> Description	In <b>Remarks DataEvent</b> behavior was incorrect and corrected since this <b>DataEvent</b> behavior is not related to the <b>readValue</b> method. In <b>See also</b> , not referencing <b>VideoCaptureMode</b> property <b>directIO</b> method were existing. They need to be eliminated.	OPOS-J / Tad Furuhata	In <b>Remarks</b> current description was as follows. <b>Remarks</b> Before this event is delivered, the individual recognition information is enqueued into the area that is indicated by the <b>readValue</b> 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.  In <b>See also</b> current description was as follows. <b>See Also VideoCaptureMode</b> Property, “Events” on page Intro-19, <b>directIO</b>	They were corrected as follows. <b>Remarks</b> Before this event is delivered, the data is copied into corresponding properties. .  <b>See Also</b> “Events” on page Intro-19	Resolved.	

## UPOS Ver1.16 RCSD Specification

					method.			
Issue149 <a href="#">Table149</a>	41/Sound Recorder	<b>startRecording</b> Method <i>recordingTime</i> parameter value	This method's <i>recordingTime</i> value was incorrect. Since in the value there was a description of OPOS_FOREVER(-1) Since OPOS is a name of POS application architecture therefore it is eliminated.	OPOS-J / Tad Furuhata	Current <i>recordingTime</i> value description was OPOS_FOREVER(-1)	The <i>recordingTime</i> value description was corrected as follows. FOREVER(-1)	Resolved.	
Issue150 <a href="#">Table150</a>	43/Sound Player	Summary <b>CapAssociatedHardTotalsDevice</b> Property	Capability needs to be used under the <b>read-only</b> . However description was <b>read-write</b> and this is incorrect.	OPOS-J / Tad Furuhata	Current property function description was <b>read-write</b> .	Capability description was corrected as <b>read-only</b> .	Resolved	
Issue151 <a href="#">Table151</a> <a href="#">Table152</a> <a href="#">Table153</a> <a href="#">Table154</a>	29/POS Power	<b>restartPOS</b> Method <b>shutdownPOS</b> Method <b>standbyPOS</b> Method <b>suspendPOS</b> Method	Method use condition description had a typo. To keep the consistency with other description <b>use after open, enable</b> was corrected as <b>use after open-enable</b>	OPOS-J / Tad Furuhata	Current description was <b>use after open, enable</b>	It is corrected as follows. <b>Use after open-enable</b>	Resolved	
Issue155 <a href="#">Table155</a> <a href="#">Table156</a>	43/Sound Player 44/Speech Synthe.	<b>OutputIDList</b> Property	Property use condition description had a typo. To keep the consistency with other description <b>use after open, claim, enable</b> was corrected as <b>use after open-claim-enable</b>	OPOS-J / Tad Furuhata	Current property use condition description was <b>use after open, claim, enable</b>	It is corrected as follows. <b>use after open-claim-enable</b>	Resolved	
Issue157 <a href="#">Table157</a>	44/Speech Synthe.	<b>stopSpeaking</b> Method	Method use condition description had a typo. To keep the consistency with other description <b>use after open, claim, enable</b> was corrected as <b>use after open-claim-enable</b>	OPOS-J / Tad Furuhata	Current description was <b>use after open, claim, enable</b>	It is corrected as follows. <b>Use after open-claim, enable</b>	Resolved	