



AEROSPACE STANDARD

AS6060™**REV. A**

Issued 2010-11
Revised 2021-10
Reaffirmed 2026-05

Superseding AS6060

(R) JAUS Environment Sensing Service Set

RATIONALE

This revision looks to update the JAUS Environment Service Set based on the latest publication of the JAUS Service Interface Definition Language (AS5684B) and JAUS Core Service Set (AS5710A). In addition, updates and fixes are applied based on feedback received from implementations. Finally, additional services related to video streaming, force/torque sensors, and other sensing modalities are added.

AS6060A has been reaffirmed to comply with the SAE Five-Year Review policy.

A reaffirmed status is given to technical reports that have been reviewed by the technical committee and deemed to be current with no need for immediate revision. No changes have been made to this document as a result of this review.

INTRODUCTION

The primary goal of the JAUS Environment Sensing Service Set is logical interoperability between communicating elements in an unmanned system. To this end, each service defines the messages (vocabulary) and protocol (rules) for data exchange. This logical interoperability is independent of the physical transport, and it is expected that a Transport Standard, such as the JAUS/SDP Transport Specification (AS5669A), is used in conjunction with this specification.

Each service in the JAUS Environment Sensing Service Set can be described using the JAUS Service Interface Definition Language (JSIDL) (AS5684B). JSIDL creates a formal schema based on Relax NG Compact [rng] that allows for validation of each service definition described herein. Although knowledge of JSIDL is not required to understand or implement this Specification, it is highly recommended for supporting context. For convenience, the JAUS Environment Sensing Service Set contains both a text based and XML based representation for each service.

This document uses a number of conventions to simplify the text. All names are given in Camel Case. Names start with upper case, while reference names start with a lower case.

The tables and diagrams in this document are hand-transcribed from the JSIDL XML specification in the [Appendix A](#). In case of transcription errors, the XML specification should be considered correct.

SAE Executive Standards Committee Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2026 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, or used for text and data mining, AI training, or similar technologies, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 1-877-606-7323 (U.S. and Canada only)
Tel: 1-724-776-4970 (outside U.S. and Canada)
Fax: 724-776-0790
Email: CustomerService@sae.org
<http://www.sae.org>

SAE WEB ADDRESS:

Full document details: [SAE AS6060A](#)

For more information on this standard, visit
<https://www.sae.org/standards/content/AS6060A/>

Documents

Standard, visit Sharing Hub

TABLE OF CONTENTS

1.	SCOPE.....	7
1.1	Purpose.....	7
1.2	Compliance.....	7
1.3	Document Organization.....	7
2.	REFERENCES.....	8
2.1	Applicable Documents.....	8
2.1.1	SAE Publications.....	8
2.1.2	J AUS Technical References.....	8
2.1.3	Other Publications.....	8
2.2	Definitions.....	9
2.3	List of Acronyms.....	9
3.	COMMON CONVENTIONS.....	10
3.1	Capabilities versus Configuration.....	10
3.2	Streaming Audio and Video.....	10
4.	SERVICE DEFINITIONS.....	10
4.1	AnalogVideo.....	10
4.1.1	Description.....	11
4.1.2	Assumptions.....	11
4.1.3	References.....	11
4.1.4	Vocabulary.....	11
4.1.5	Encoding.....	11
4.1.6	Protocol Behavior.....	14
4.2	DigitalAudioOutput.....	15
4.2.1	Description.....	15
4.2.2	Assumptions.....	15
4.2.3	References.....	15
4.2.4	Vocabulary.....	16
4.2.5	Encoding.....	16
4.2.6	Protocol Behavior.....	23
4.3	DigitalAudioSensor.....	24
4.3.1	Description.....	24
4.3.2	Assumptions.....	24
4.3.3	References.....	24
4.3.4	Vocabulary.....	25
4.3.5	Encoding.....	25
4.3.6	Protocol Behavior.....	34
4.4	DigitalResourceDiscovery.....	35
4.4.1	Description.....	35
4.4.2	Assumptions.....	35
4.4.3	References.....	35
4.4.4	Vocabulary.....	35
4.4.5	Encoding.....	35
4.4.6	Protocol Behavior.....	38
4.5	DigitalVideo.....	39
4.5.1	Description.....	39
4.5.2	Assumptions.....	39
4.5.3	References.....	39
4.5.4	Vocabulary.....	39
4.5.5	Encoding.....	39
4.5.6	Protocol Behavior.....	47
4.6	ForceTorqueSensor.....	48
4.6.1	Description.....	48
4.6.2	Assumptions.....	49
4.6.3	References.....	49

4.6.4	Vocabulary	49
4.6.5	Encoding	49
4.6.6	Protocol Behavior	52
4.7	H264VideoEncoding	52
4.7.1	Description	52
4.7.2	Assumptions.....	52
4.7.3	References.....	52
4.7.4	Vocabulary	53
4.7.5	Encoding	53
4.7.6	Protocol Behavior	65
4.8	PortMapper	66
4.8.1	Description	66
4.8.2	Assumptions.....	66
4.8.3	References.....	66
4.8.4	Vocabulary	66
4.8.5	Encoding	67
4.8.6	Protocol Behavior	69
4.9	RangeSensor	69
4.9.1	Description	69
4.9.2	Assumptions.....	70
4.9.3	References.....	70
4.9.4	Vocabulary	71
4.9.5	Encoding	71
4.9.6	Protocol Behavior	84
4.10	StillImage	85
4.10.1	Description	85
4.10.2	Assumptions.....	85
4.10.3	References.....	85
4.10.4	Vocabulary	86
4.10.5	Encoding	86
4.10.6	Protocol Behavior	94
4.11	VideoIlluminator	95
4.11.1	Description	95
4.11.2	Assumptions.....	95
4.11.3	References.....	95
4.11.4	Vocabulary	95
4.11.5	Encoding	96
4.11.6	Protocol Behavior	103
4.12	VisualSensor	104
4.12.1	Description	104
4.12.2	Assumptions.....	104
4.12.3	References.....	104
4.12.4	Vocabulary	104
4.12.5	Encoding	104
4.12.6	Protocol Behavior	121
5.	NOTES	122
5.1	Revision Indicator.....	122
5.2	RangeSensor	181

APPENDIX A	XML FOR SERVICE DEFINITIONS	123
------------	-----------------------------------	-----

Figure 1	AnalogVideo service protocol behavior.....	14
Figure 2	DigitalAudioOutput service protocol behavior.....	23
Figure 3	DigitalAudioSensor service protocol behavior	34
Figure 4	DigitalResourceDiscovery service protocol behavior.....	38
Figure 5	DigitalVideo service protocol behavior.....	47
Figure 6	ForceTorqueSensor service protocol behavior.....	52
Figure 7	H264VideoEncoding service protocol behavior	65
Figure 8	PortMapper service protocol behavior	69



Figure 9	Bearing and inclination definition	70
Figure 10	RangeSensor service protocol behavior	84
Figure 11	StillImage service protocol behavior	94
Figure 12	Videollluminator service protocol behavior	103
Figure 13	VisualSensor service protocol behavior	121
Table 1	AnalogVideo service message vocabulary	11
Table 2	SetAnalogVideoSensorConfiguration message encoding	12
Table 3	QueryAnalogVideoSensorConfiguration message encoding	12
Table 4	QueryAnalogVideoSensorCapabilities message encoding	13
Table 5	ReportAnalogVideoSensorConfiguration message encoding	13
Table 6	ReportAnalogVideoSensorCapabilities message encoding	14
Table 7	AnalogVideo service state transitions	15
Table 8	AnalogVideo service conditions	15
Table 9	AnalogVideo service transition actions	15
Table 10	DigitalAudioOutput service message vocabulary	16
Table 11	SetDigitalAudioOutputSource message encoding	16
Table 12	QueryDigitalAudioOutputSource message encoding	17
Table 13	QueryDigitalAudioOutputCapabilities message encoding	17
Table 14	ConfirmSensorConfiguration message encoding	18
Table 15	ReportDigitalAudioOutputSource message encoding	21
Table 16	ReportDigitalAudioOutputCapabilities message encoding	21
Table 17	DigitalAudioOutput service state transitions	24
Table 18	DigitalAudioOutput service conditions	24
Table 19	DigitalAudioOutput service transition actions	24
Table 20	DigitalAudioSensor service message vocabulary	25
Table 21	SetDigitalAudioSensorConfiguration message encoding	25
Table 22	QueryDigitalAudioSensorCapabilities message encoding	26
Table 23	QueryDigitalAudioSensorConfiguration message encoding	27
Table 24	ConfirmSensorConfiguration message encoding	27
Table 25	ReportDigitalAudioSensorCapabilities message encoding	30
Table 26	ReportDigitalAudioSensorConfiguration message encoding	33
Table 27	DigitalAudioSensor service state transitions	34
Table 28	DigitalAudioSensor service conditions	34
Table 29	DigitalAudioSensor service transition actions	34
Table 30	DigitalResourceDiscovery service message vocabulary	35
Table 31	QueryDigitalResourceEndpoint message encoding	35
Table 32	RegisterDigitalResourceEndpoint message encoding	36
Table 33	RemoveDigitalResourceEndpoint message encoding	37
Table 34	ReportDigitalResourceEndpoint message encoding	37
Table 35	ConfirmDigitalResourceEndpoint message encoding	38
Table 36	DigitalResourceDiscovery service state transitions	38
Table 37	DigitalResourceDiscovery service transition actions	38
Table 38	DigitalVideo service message vocabulary	39
Table 39	ControlDigitalVideoSensorStream message encoding	40
Table 40	ControlDigitalVideoSensorRecording message encoding	40
Table 41	SetDigitalVideoSensorConfiguration message encoding	41
Table 42	QueryDigitalVideoSensorConfiguration message encoding	42
Table 43	QueryDigitalVideoSensorCapabilitiesExt message encoding	42
Table 44	QueryDigitalVideoSensorStatus message encoding	43
Table 45	ReportDigitalVideoSensorCapabilitiesExt message encoding	43
Table 46	ReportDigitalVideoSensorConfiguration message encoding	46
Table 47	ReportDigitalVideoSensorStatus message encoding	47
Table 48	DigitalVideo service state transitions	48
Table 49	DigitalVideo service conditions	48
Table 50	DigitalVideo service transition actions	48
Table 51	ForceTorqueSensor service message vocabulary	49
Table 52	QueryForceTorqueCapabilities message encoding	49
Table 53	QueryForceTorque message encoding	49
Table 54	ReportForceTorqueCapabilities message encoding	50

Table 55	ReportForceTorque message encoding	51
Table 56	ForceTorqueSensor service state transitions	52
Table 57	ForceTorqueSensor service transition actions	52
Table 58	H264VideoEncoding service message vocabulary.....	53
Table 59	SetH264VideoEncodingConfiguration message encoding	53
Table 60	QueryH264VideoEncodingCapabilities message encoding	55
Table 61	QueryH264VideoEncodingConfiguration message encoding.....	56
Table 62	ConfirmSensorConfiguration message encoding	57
Table 63	ReportH264VideoEncodingCapabilities message encoding	60
Table 64	ReportH264VideoEncodingConfiguration message encoding.....	63
Table 65	H264VideoEncoding service state transitions	65
Table 66	H264VideoEncoding service conditions.....	65
Table 67	H264VideoEncoding service transition actions.....	66
Table 68	PortMapper service message vocabulary.....	66
Table 69	CancelPortMapping message encoding	67
Table 70	RequestPortMapping message encoding	67
Table 71	QueryPortMappings message encoding.....	67
Table 72	GrantPortMapping message encoding	68
Table 73	ReportPortMappings message encoding.....	68
Table 74	PortMapper service state transitions.....	69
Table 75	PortMapper service transition actions.....	69
Table 76	RangeSensor service message vocabulary.....	71
Table 77	SetRangeSensorConfiguration message encoding.....	71
Table 78	QuerySensorGeometricProperties message encoding	72
Table 79	QueryRangeSensorCompressedData message encoding.....	73
Table 80	QueryRangeSensorData message encoding	73
Table 81	QueryRangeSensorCapabilities message encoding	74
Table 82	QueryRangeSensorConfiguration message encoding	74
Table 83	ConfirmSensorConfiguration message encoding	75
Table 84	ReportRangeSensorCapabilities message encoding	78
Table 85	ReportRangeSensorConfiguration message encoding	79
Table 86	ReportRangeSensorData message encoding	80
Table 87	ReportRangeSensorCompressedData message encoding.....	81
Table 88	ReportSensorGeometricProperties message encoding	83
Table 89	RangeSensor service state transitions	84
Table 90	RangeSensor service conditions	84
Table 91	RangeSensor service transition actions.....	85
Table 92	StillImage service message vocabulary.....	86
Table 93	SetStillImageSensorConfiguration message encoding.....	86
Table 94	QueryStillImageDataExt message encoding	88
Table 95	QueryStillImageSensorConfiguration message encoding	88
Table 96	QueryStillImageSensorCapabilities message encoding	89
Table 97	ReportStillImageDataExt message encoding	89
Table 98	ReportStillImageSensorConfiguration message encoding	90
Table 99	ReportStillImageSensorCapabilities message encoding	91
Table 100	StillImage service state transitions.....	94
Table 101	StillImage service conditions.....	94
Table 102	StillImage service transition actions.....	95
Table 103	VideoIlluminator service message vocabulary.....	95
Table 104	SetVideoIlluminatorConfiguration message encoding	96
Table 105	QueryVideoIlluminatorConfiguration message encoding.....	97
Table 106	QueryVideoIlluminatorCapabilities message encoding	97
Table 107	ConfirmSensorConfiguration message encoding	98
Table 108	ReportVideoIlluminatorConfiguration message encoding.....	101
Table 109	ReportVideoIlluminatorCapabilities message encoding	101
Table 110	VideoIlluminator service state transitions.....	103
Table 111	VideoIlluminator service conditions.....	103
Table 112	VideoIlluminator service transition actions.....	103
Table 113	VisualSensor service message vocabulary	104
Table 114	QuerySensorGeometricProperties message encoding	105

Table 115	QueryVisualSensorConfiguration message encoding	105
Table 116	QueryVisualSensorCapabilities message encoding	106
Table 117	QueryVisualSensorCapabilitiesExt message encoding	106
Table 118	SetVisualSensorConfiguration message encoding	107
Table 119	ConfirmSensorConfiguration message encoding	109
Table 120	ReportSensorGeometricProperties message encoding	112
Table 121	ReportVisualSensorConfiguration message encoding	113
Table 122	ReportVisualSensorCapabilities message encoding	115
Table 123	ReportVisualSensorCapabilitiesExt message encoding	118
Table 124	VisualSensor service state transitions	121
Table 125	VisualSensor service conditions	122
Table 126	VisualSensor service transition actions	122

1. SCOPE

This document defines a set of standard application layer interfaces called JAUS Environment Sensing Services. JAUS Services provide the means for software entities in an unmanned system or system of unmanned systems to communicate and coordinate their activities. The Environment Sensing Services represent typical environment sensing capabilities commonly found across all domains and types of unmanned systems in a platform-independent manner. At present, twelve services are defined in this document:

- Range Sensor: Determine the proximity of objects in the platform's environment
- Visual Sensor: Provides common configuration and setup for different types of imaging systems
- Digital Video: A type of Visual Sensor that manages digital video
- Analog Video: A type of Visual Sensor that manages analog video
- Still Image: A type of Visual Sensor that manages and encodes individual digital images
- Digital Audio Sensor: Provides common configuration and setup for different types of audio streams
- Digital Audio Output: Send streaming audio to a destination, frequently a speaker on a remote platform
- Digital Resource Discovery: Support run-time discovery of RTSP-based streaming audio and video
- Force Torque Sensor: Report the forces and torques measured in six degrees of freedom
- H264 Video Encoding: Configure the parameters of an H.264 based video encoded stream
- Port Mapper: Provides a bridge between internal and external networks, similar to Network Address Translation
- Video Illuminator: Control one or more lights associated with one or more cameras

Each service is described by a JAUS Service Definition (JSD) which specifies the message set and protocol required for compliance. Each JSD is fully compliant with the JAUS Service Interface Definition Language (AS5684).

1.1 Purpose

The purpose of this document is to facilitate interoperation of unmanned vehicle systems, subsystems, and payloads by standardization of the message set and associated protocol.

1.2 Compliance

The JAUS Environment Sensing Service Set must support compliance assessment. To do so, this specification must be sufficiently precise to enable the “compliant”/“not compliant” distinction to be made independently of the underlying transport mechanism. It is important to note that implementations are considered compliant to individual Service Definitions within this Specification; it is not necessary that a single entity realize each Service to be considered compliant.

1.3 Document Organization

The layout of this document is as follows. Section [2](#) lists external references used throughout the specification. Section [3](#) specifies the JAUS Service Definition for each of the services, with particular emphasis on the description, assumptions, message set, and protocol behavior. Section [4](#) describes the message encoding for each message set. Finally, [Appendix A](#) contains the complete JSIDL representation for each service and their associated message set.

2. REFERENCES

2.1 Applicable Documents

The following publications form a part of this document to the extent specified herein. The latest issue of SAE publications shall apply. The applicable issue of other publications shall be the issue in effect on the date of the purchase order. In the event of conflict between the text of this document and references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

AIR5665B Architecture Framework for Unmanned Systems

AS5669A JAUS/SDP Transport Specification

AS5684B JAUS Service Interface Definition Language

AS5710A JAUS Core Service Set

AS6009A JAUS Mobility Service Set

2.1.2 JAUS Technical References

RA33P1 JAUS Reference Architecture Specification, Volume II, Part 1, Architecture Framework, Version 3.3, June 22, 2007

RA33P2 JAUS Reference Architecture Specification, Volume II, Part 2, Message Definition, Version 3.3, June 22, 2007

RA33P3 JAUS Reference Architecture Specification, Volume II, Part 3, Message Set, Version 3.3, June 22, 2007

2.1.3 Other Publications

[bnt] Barry N. Taylor. The International System of Units (SI), National Institute of Standards and Technology Special Publication 330, 1991 Edition [<http://physics.nist.gov/Document/sp330.pdf>]

[crane] Kinematic Analysis of Robot Manipulators, Carl D. Crane III and Joseph Duffy, Cambridge University Press, 1998

[jpeg] ITU-T81 [<http://www.w3.org/Graphics/JPEG/itu-t81.pdf>]

[mpeg-2] ISO/IEC 13818-1:2007

[mpeg4avc] ISO/IEC 14496-10:2008

[mpeg4visual] ISO/IEC 14496-2:2004

[nitr] National Imagery Transmission Format Version 2.1, March 1, 2001
[<http://www.gwg.nga.mil/ntb/baseline/docs/2500c/2500C.pdf>]

[rng] Relax NG [<http://www.oasis-open.org/committees/relax-ng/spec-20011203.html>] Standard lightweight XML schema language.

[tiff] TIFF, Revision 6.0, Final June 3, 1992 [<http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>]

[rfc1951]	DEFLATE Compressed Data Format Specification version 1.3 [http://tools.ietf.org/html/rfc1951]
[bzip2]	bzip2 library for data compression [http://www.bzip.org/1.0.5/bzip2-manual-1.0.5.html]
[lzma]	LZMA data compression algorithm SDK [http://www.7-zip.org/sdk.html]
[url]	Uniform Resource Identifier (URI): Generic Syntax [https://datatracker.ietf.org/doc/html/rfc3986]

2.2 Definitions

CAPABILITIES: This document denotes a sensor's capabilities as the entire range of parameters and values supported by the sensor. These are intrinsic to the make, model, and type of sensor, and cannot be modified at run-time by the client.

CONFIGURATION: While capabilities represents the entire range of supported parameters, the configuration represents the current or active settings. Configurations can be changed at run-time by the controlling client.

DATA POINT: A data point describes a specific object within a given space that consists of neither volume, area, length, nor any other higher dimensional analogue. Thus, a data point is a zero-dimensional object which is defined by bearing, inclination, and range.

QUATERNION: This specification uses quaternion-based notation to reflect three dimensional rotation of a coordinate system as measured with respect to a base coordinate system. This rotation is defined by a unit quaternion ($d; a, b, c$) where the quaternion q is expressed as $q = d + ai + bj + ck$ and the vectors (i, j, k) are unit vectors along the (x, y, z) axes of the base coordinate frame, respectively. [crane]

2.3 List of Acronyms

ANSI	American National Standards Institute
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ID	Identifier
JAUS	Joint Architecture for Unmanned Systems
JSD	JAUS Service Definition
JSIDL	JAUS Service (Interface) Definition Language
LZMA	Lempel-Ziv-Markov Chain-Algorithm
MPEG	Moving Picture Experts Group
MJPEG	Motion Joint Photographic Experts Group
NIST	National Institute of Standards and Technology
RA	(JAUS) Reference Architecture
SDK	Software Development Kit
UML	Unified Modeling Language
URL	Uniform Resource Locator
URN	Uniform Resource Name

URI	Uniform Resource Identifier
UUID	Universally Unique Identifier
XML	Extensible Markup Language

3. COMMON CONVENTIONS

3.1 Capabilities versus Configuration

The Environmental Sensing Service Set makes extensive use of capabilities and configuration messages. Generally, capabilities describe all possible values that an implementation supports, e.g., it describes all possible configurations. A specific configuration then reflects the current values of the sensor properties. For example, a still image camera may support VGA, SVGA, and XGA resolutions (three values reported in the capabilities message), but at any given time can only be configured to give images in one resolution (one value reported in the configuration message).

In cases where an implementation only supports a single value, that value should still be reported in the capabilities message and in the configuration report (when requested via the presence vector). Note that this value cannot be changed by a set configuration message, as only a single value is allowed, but it may still be crucial for a client to understand the fixed value.

In cases where an implementation does not know the supported values, or the listed enumeration types do not apply, implementations may make use of the optionality flag for those specific fields, and they may be absent from the capabilities and configuration messages completely.

3.2 Streaming Audio and Video

Many of the services defined herein focus on configuration of audio and video sensors, such as cameras and microphones. To access the streaming media itself, the Environmental Sensing Service Set relies on encoded data streams using the Real Time Streaming Protocol (RTSP). Generally, RTSP endpoints are described using a URN of the form “rtsp://<ip-address>/end/point/name”.

To support run-time discovery of available media streams, this Service Set includes a Digital Resource Discovery Service (DRDS). Like the Discovery Service within the AS5710 JAUS Core Service Set, the DRDS provides a means for cameras, microphone, and other stream media generators to register one or more available RTSP endpoints at run-time. Clients can then query the DRDS for those endpoint URNs, along with the JAUS and Sensor IDs of the associate services for configuring the stream.

However, many systems use a tiered network, such that an internal subnet is used on-board a platform while an external facing subnet is used for connecting to operator control units and other clients. As a result, internal addresses, such as 192.168.X.X, are not valid on the external network, causing issues when media streams register RTSP endpoints. Consequently, the Service Set also includes the Port Mapper Service, capable of mapping network traffic from an external facing subnet to an internal one, similar in behavior to Network Address Translation. In the nominal use case, a media stream provider that offers a stream on an internal network can also request an external facing network address and port from the Port Mapper. The stream provider can then register the available stream with the DRDS using the internal address, the external address, or both.

4. SERVICE DEFINITIONS

The following subsections provide a textual definition for each Service Definition in the JAUS Environment Sensing Service Set. Corresponding JSIDL definitions are offered in the Appendix. Additional information on interpreting the service definition elements may be found in [\[JSIDL\]](#).

4.1 AnalogVideo

name=AnalogVideo
version=2.0
id=urn:jaus:jss:environmentSensing:AnalogVideo

4.1.1 Description

This service provides access to the capabilities and configuration of the analog visual sensor, allowing the controlling component to set the visual sensor to a particular operational profile. The actual transmission of the video stream is outside the scope of this service.

4.1.2 Assumptions

Messages may be delayed, lost, or reordered.

4.1.3 References

Inherits-from=visualSensor

version=2.0

id=urn:jaus:jss:environmentSensing:VisualSensor

4.1.4 Vocabulary

Table 1 - AnalogVideo service message vocabulary

Message ID (hex)	Name	Command
Input Set		
0806	SetAnalogVideoSensorConfiguration	true
2811	QueryAnalogVideoSensorConfiguration	false
2810	QueryAnalogVideoSensorCapabilities	false
Output Set		
4811	ReportAnalogVideoSensorConfiguration	false
4810	ReportAnalogVideoSensorCapabilities	false

4.1.5 Encoding

4.1.5.1 Input Set

4.1.5.1.1 ID 0806: SetAnalogVideoSensorConfiguration

This message is used to set the configuration of the analog video sensor associated with the service. Configuration is based on each sensor's capabilities as described in the Report Analog Video Sensor Capabilities message. This message shall cause the receiving service to reply to the sender with a Confirm Sensor Configuration message. If the configuration specified is invalid for a given sensor ID, the confirm message shall contain an Analog Video Error Record for the given Sensor ID however other, valid, configurations specified shall be set (if they exist).

Table 2 - SetAnalogVideoSensorConfiguration message encoding

<pre> body └─ sequence name = AnalogVideoSensorConfigurationSequence └─ record name = RequestIdRec └─ list name = AnalogVideoSensorList (count_field = unsigned short integer) └─ record name = AnalogVideoSensorConfigurationRec </pre>					
Record Name = RequestIdRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestID	unsigned byte	one	false	
Record Name = AnalogVideoSensorConfigurationRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> AnalogFormat	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= NTSC-M 1= NTSC-J 2= PAL-N 3= PAL-M 4= SECAM-L 5= SECAM-B/G

4.1.5.1.2 ID 2811: QueryAnalogVideoSensorConfiguration

This message shall cause the receiving service to reply to the requestor with a Report Analog Video Sensor Configuration message.

Table 3 - QueryAnalogVideoSensorConfiguration message encoding

<pre> body └─ list name = SensorIDList (count_field = unsigned short integer) └─ record name = SensorIDQueryRec </pre>					
Record Name = SensorIDQueryRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried

4.1.5.1.3 ID 2810: QueryAnalogVideoSensorCapabilities

This message shall cause the receiving service to reply to the requestor with a Report Analog Video Sensor Capabilities message.

Table 4 - QueryAnalogVideoSensorCapabilities message encoding

body └─ list name = SensorIDList (count_field = unsigned short integer) └─ record name = SensorIDQueryRec					
Record Name = SensorIDQueryRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried

4.1.5.2 Output Set

4.1.5.2.1 ID 4811: ReportAnalogVideoSensorConfiguration

This message is sent in response to a Query Analog Video Sensor Configuration message. It is populated with the current sensor configuration (per sensor ID) as defined in the table below.

Table 5 - ReportAnalogVideoSensorConfiguration message encoding

body └─ list name = AnalogVideoSensorConfigurationList (count_field = unsigned short integer) └─ record name = AnalogVideoSensorConfigurationRec					
Record Name = AnalogVideoSensorConfigurationRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> AnalogFormat	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= NTSC-M 1= NTSC-J 2= PAL-N 3= PAL-M 4= SECAM-L 5= SECAM-B/G

4.1.5.2.2 ID 4810: ReportAnalogVideoSensorCapabilities

This message is used to report the sensors' capabilities upon receipt of a Query Analog Video Sensor Capabilities message. Capabilities include sensor properties, values and ranges which can be modified by the Set Analog Video Sensor Configuration message.

Table 6 - ReportAnalogVideoSensorCapabilities message encoding

body list name = AnalogVideoSensorCapabilitiesList (count_field = unsigned short integer) record name = AnalogVideoSensorCapabilitiesRec					
Record Name = AnalogVideoSensorCapabilitiesRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<bit_field> SupportedAnalogFormats	unsigned byte	bit_field	false	Bits 0..0, NTSCM: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, NTSCJ: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, PALN: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, PALM: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, SECAML: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, SECAMBG: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported

4.1.6 Protocol Behavior

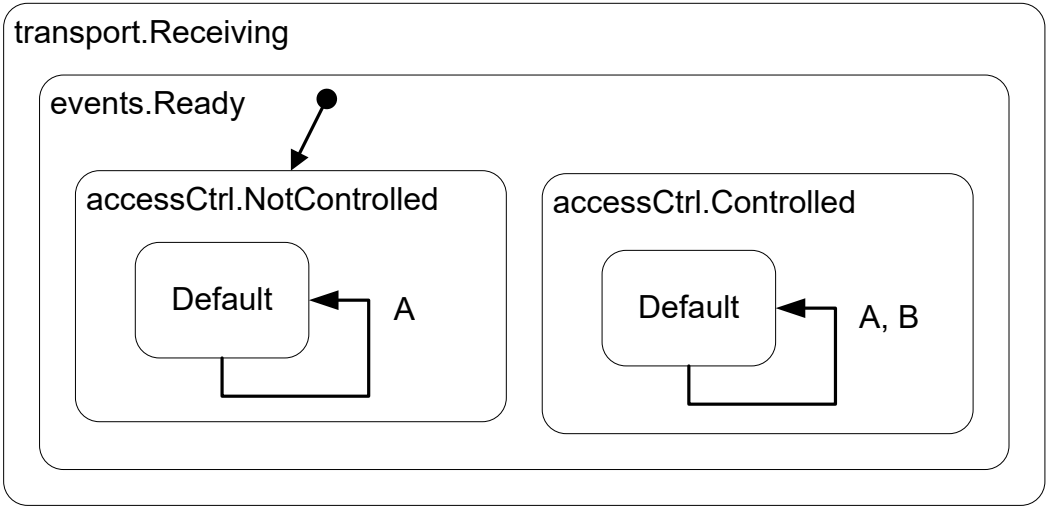


Figure 1 - AnalogVideo service protocol behavior

Table 7 - AnalogVideo service state transitions

Start State	Trigger	Conditions	Actions
A	QueryAnalogVideoSensorCapabilities		sendReportAnalogVideoSensorCapabilities (msg,transportData)
A	QueryAnalogVideoSensorConfiguration		sendReportAnalogVideoSensorConfiguration (msg,transportData)
B	SetAnalogVideoSensorConfiguration	isControllingClient(transportData)	sendConfirmSensorConfiguration (msg, transportData) , updateAnalogVideoSensorConfiguration (msg)

Table 8 - AnalogVideo service conditions

Condition	Interpretation
isControllingClient(transportData)	True if the message that triggered the transition is received from the client that is in control of this service.

Table 9 - AnalogVideo service transition actions

Action	Interpretation
sendReportAnalogVideoSensorCapabilities	Send a ReportAnalogVideoSensorCapabilities message
sendReportAnalogVideoSensorConfiguration	Send a ReportAnalogVideoSensorConfiguration message
sendConfirmSensorConfiguration	Send sendConfirmSensorConfiguration message
updateAnalogVideoSensorConfiguration	Update the sensor user controllable configuration parameters according to the ones specified.

4.2 DigitalAudioOutput

name=DigitalAudioOutput

version=2.0

id=urn:jaus:jss:environmentSensing:DigitalAudioOutput

4.2.1 Description

The Digital Audio Output Service allows a client to specify an RTSP stream as an audio source. The service is expected to connect, decode, and play the specified stream, presumably over one or more speakers. This can be used to send live audio to a speaker or annunciator on the platform, or playback a prerecorded message. The Capabilities message can be used to determine what codecs the implementation supports.

4.2.2 Assumptions

Messages may be delayed, lost, or reordered.

4.2.3 References

Inherits-from=accessControl

version=1.1

id=urn:jaus:jss:core:AccessControl

4.2.4 Vocabulary

Table 10 - DigitalAudioOutput service message vocabulary

Message ID (hex)	Name	Command
Input Set		
0818	SetDigitalAudioOutputSource	true
2818	QueryDigitalAudioOutputSource	false
2819	QueryDigitalAudioOutputCapabilities	false
Output Set		
0801	ConfirmSensorConfiguration	false
4818	ReportDigitalAudioOutputSource	false
4819	ReportDigitalAudioOutputCapabilities	false

4.2.5 Encoding

4.2.5.1 Input Set

4.2.5.1.1 ID 0818: SetDigitalAudioOutputSource

This message is used to specify the stream source to play. Note that the specified stream must not require a DNS lookup to resolve.

Table 11 - SetDigitalAudioOutputSource message encoding

<pre> body ├── sequence name = DigitalAudioOutputSeq │ ├── record name = RequestIdRec │ │ └── list name = DigitalAudioOutputSourceList │ │ (count_field = unsigned short integer) │ │ └── record name = DigitalAudioOutputSourceRec </pre>					
Record Name = RequestIdRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestId	unsigned byte	one	false	
Record Name = DigitalAudioOutputSourceRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<variable_length_string> StreamURL	variable length string (byte[])	N/A	false	URL of the source of the stream. This URL should not require a DNS to resolve; hence, an IP address should be substituted for a host name. (Length min..max = 0..255)
3	<fixed_field> Repeat	unsigned byte	one	false	Specifies behavior for RTSP streams that have a finite play time. Value set, offset=false, ranges/enums: 0= PlayOnce 1= RepeatUntilNewStreamSpecified
4	<fixed_field> Volume	unsigned short integer	percent	false	The gain or volume applied to the input as a percent. A value of 0% indicates that no output signal is generated, while a value of 100% indicates that maximum amplification is applied to the input. (scaled range = [0,100] round)

4.2.5.1.2 ID 2818: QueryDigitalAudioOutputSource

This message is used to query the stream source currently being played for each sensor.

Table 12 - QueryDigitalAudioOutputSource message encoding

body └ record name = QueryDigitalAudioRec					
Record Name = QueryDigitalAudioRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried

4.2.5.1.3 ID 2819: QueryDigitalAudioOutputCapabilities

This message is used to query the full set of capabilities (supported performance levels) for one or more sensors. A SensorID of zero may be used to query all sensors supported by the service in a single message.

Table 13 - QueryDigitalAudioOutputCapabilities message encoding

body └ record name = QueryDigitalAudioRec					
Record Name = QueryDigitalAudioRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried

4.2.5.2 Output Set

4.2.5.2.1 ID 0801: ConfirmSensorConfiguration

This message is used to notify a client component that the configuration has been received with the values specified in the corresponding set message with Request ID matching the value of field 1 of this message. If the specified configuration request is deemed valid, the SensorIDRec is returned with the matching SensorID (or illuminatorID) of the sensor (or illuminator) for which the configuration was successfully set. If the requested configuration is invalid, one of the ErrorRec types shall be returned (depending on the source message) with an error code and description of the configuration setting which was deemed invalid.

Table 14 - ConfirmSensorConfiguration message encoding

<pre> body └─ sequence name = ConfirmSensorConfigurationSequence └─ record name = RequestIdRec └─ list name = ConfirmSensorList (count_field = unsigned short integer) └─ variant name = ConfirmSensorConfigurationVariant (vtag_field = unsigned byte) └─ record name = SensorIdRec └─ record name = RangeSensorErrorRec └─ record name = VisualSensorErrorRec └─ record name = DigitalVideoSensorErrorRec └─ record name = AnalogVideoSensorErrorRec └─ record name = StillImageSensorErrorRec └─ record name = H264VideoEncodingErrorRec └─ record name = DigitalAudioSensorErrorRec └─ record name = DigitalAudioOutputErrorRec └─ record name = VideoIlluminatorErrorRec </pre>					
Record Name = RequestIdRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestId	unsigned byte	one	false	
Record Name = SensorIdRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
Record Name = RangeSensorErrorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> RangeSensorErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Horizontal Field of View 2= Invalid Vertical Field of View 3= Invalid Update Rate 4= Invalid Sensor Range 5= Invalid Sensor State 6= Multiple Invalid Parameters 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
Record Name = VisualSensorErrorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]

2	<fixed_field> VisualSensorErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Sensor State 2= Invalid Zoom Mode 3= Invalid Zoom Value 4= Invalid Focus Mode 5= Invalid Focus Value 6= Invalid White Balance 7= Invalid Imaging Mode 8= Invalid Exposure Mode 9= Invalid Metering Mode 10= Invalid Shutter Speed 11= Invalid Aperture Value 12= Invalid Light Sensitivity 13= Invalid Image Stabilization 14= Invalid Horizontal FOV 15= Invalid Vertical FOV 16= Multiple Invalid Parameters 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = DigitalVideoSensorErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> DigitalVideoErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Minimum Bit Rate 2= Invalid Maximum Bit Rate 3= Requested Frame Rate Too Low 4= Requested Frame Rate Too High 5= Invalid Frame Size 6= Invalid Format 7= Multiple Invalid Parameters 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = AnalogVideoSensorErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> AnalogVideoErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Format 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = StillImageSensorErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> StillImageErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Frame Size 2= Invalid Format 3= Multiple Invalid Parameters 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = H264VideoEncodingErrorRec

Field #	Name	Type	Units	Optional	Interpretation
---------	------	------	-------	----------	----------------

1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> H264VideoEncodingErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Profile 2= Invalid Preset 3= Invalid GroupOfPictures 4= Invalid GDR 5= Invalid RegionOfInterest 6= Invalid IntraMacroRefresh 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = DigitalAudioSensorErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> DigitalAudioSensorErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid BitRate 2= Invalid Format 3= Invalid SampleRate 4= Invalid BitDepth 5= Invalid EncodingQuality 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = DigitalAudioOutputErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> DigitalAudioOutputErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Stream Not Found 2= Stream Not Supported 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = VideoIlluminatorErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> IlluminatorID	unsigned short integer	one	false	
2	<fixed_field> VideoIlluminatorErrorRecCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Unsupported Mode 2= Unsupported Beam Width 3= Unsupported Beam Height 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

4.2.5.2.2 ID 4818: ReportDigitalAudioOutputSource

This message is used to report the stream currently being played for each queried sensor. An empty StreamURL implies that no stream is being played.

Table 15 - ReportDigitalAudioOutputSource message encoding

<pre> body └─ list name = DigitalAudioOutputSourceList └─ (count_field = unsigned short integer) └─ record name = DigitalAudioOutputSourceRec </pre>					
Record Name = DigitalAudioOutputSourceRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<variable_length_string> StreamURL	variable length string (byte[])	N/A	false	URL of the source of the stream. This URL should not require a DNS to resolve; hence, an IP address should be substituted for a host name. (Length min..max = 0..255)
3	<fixed_field> Repeat	unsigned byte	one	false	Specifies behavior for RTSP streams that have a finite play time. Value set, offset=false, ranges/enums: 0= PlayOnce 1= RepeatUntilNewStreamSpecified
4	<fixed_field> Volume	unsigned short integer	percent	false	The gain or volume applied to the input as a percent. A value of 0% indicates that no output signal is generated, while a value of 100% indicates that maximum amplification is applied to the input. (scaled range = [0,100], round)

4.2.5.2.3 ID 4819: ReportDigitalAudioOutputCapabilities

This message is used to report full set of capabilities (supported performance levels) for one or more sensors.

Table 16 - ReportDigitalAudioOutputCapabilities message encoding

<pre> body └─ list name = DigitalAudioOutputCapabilitiesList └─ (count_field = unsigned short integer) └─ record name = DigitalAudioOutputCapabilitiesRec </pre>					
Record Name = DigitalAudioOutputCapabilitiesRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
3	<fixed_field> MinimumAllowedBitRate	unsigned short integer	one	true	kilobits per second
4	<fixed_field> MaximumAllowedBitRate	unsigned short integer	one	true	kilobits per second

5	<bit_field> SupportedDigitalFormats	unsigned integer	bit_field	true	Bits 0..0, LPCM_PCM: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 1..1, AIFF: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 2..2, WAV: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 3..3, ALAC: Value set, offset=false, ranges/enums: 1= SUPPORTED 0= NOT_SUPPORTED Bits 4..4, FLAC: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 5..5, RealAudio: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 6..6, WMA9_Lossless: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 7..7, TrueAudio: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 8..8, DolbyDigital: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 9..9, DTS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 10..10, MP2: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 11..11, MP3: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 12..12, AAC_MPEG2: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 13..13, AAC_MPEG4: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 14..14, VORBIS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 15..15, WMA: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 16..16, Speex: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED
6	<fixed_field> MinSampleRate	unsigned integer	one	true	Samples per second
7	<fixed_field> MaxSampleRate	unsigned integer	one	true	Samples per second



8	<bit_field> SupportedBitDepths	unsigned short integer	bit_field	true	Bits 0..0, EIGHT_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 1..1, TEN_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 2..2, TWELVE_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 3..3, FOURTEEN_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 4..4, SIXTEEN_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 5..5, TWENTY_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 6..6, TWENTY_FOUR_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 7..7, THIRTY_TWO_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED
---	-----------------------------------	------------------------	-----------	------	--

4.2.6 Protocol Behavior

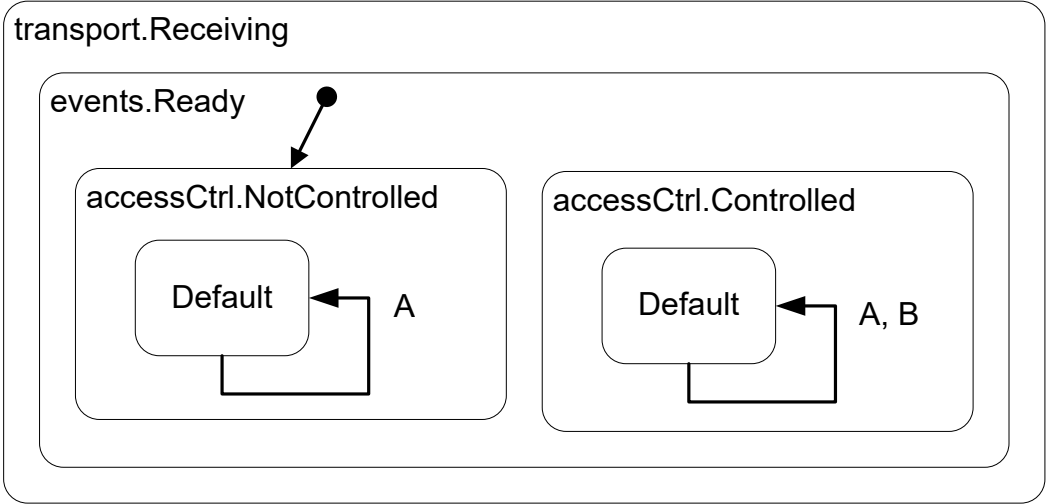


Figure 2 - DigitalAudioOutput service protocol behavior

Table 17 - DigitalAudioOutput service state transitions

Start State	Trigger	Conditions	Actions
A	QueryDigitalAudioOutputSource		sendReportDigitalAudioOutputSource (msg, transportData)
A	QueryDigitalAudioOutputCapabilities		sendReportDigitalAudioOutputCapabilities (msg, transportData)
B	SetDigitalAudioOutputSource	isControllingClient(transportData)	playStream (msg) , sendConfirmSensorConfiguration (msg, transportData)

Table 18 - DigitalAudioOutput service conditions

Condition	Interpretation
isControllingClient(transportData)	True if the command message was received from the client currently controlling this component.

Table 19 - DigitalAudioOutput service transition actions

Action	Interpretation
sendReportDigitalAudioOutputSource	Send a ReportDigitalAudioOutputSource message to querying client
sendReportDigitalAudioOutputCapabilities	Send a ReportDigitalAudioOutputCapabilities message to querying client
playStream	Begin playback for any sensor IDs with valid and supported streams specified in the message with the given repeat behavior and volume.
sendConfirmSensorConfiguration	Send sendConfirmSensorConfiguration message with confirmation or error code for each specified sensor ID

4.3 DigitalAudioSensor

name=DigitalAudioSensor

version=2.0

id=urn:jaus:jss:environmentSensing:DigitalAudioSensor

4.3.1 Description

The Digital Audio Sensor Service provides a means of configuring a digital audio stream, often from a microphone or other source. Note that the transport of the digitized audio stream itself is not covered by this service, and may use existing audio networking standards such as RTSP.

4.3.2 Assumptions

Messages may be delayed, lost, or reordered.

4.3.3 References

Inherits-from=AccessControl

version=1.1

id=urn:jaus:jss:core:AccessControl

4.3.4 Vocabulary

Table 20 - DigitalAudioSensor service message vocabulary

Message ID (hex)	Name	Command
Input Set		
080F	SetDigitalAudioSensorConfiguration	true
2817	QueryDigitalAudioSensorCapabilities	false
280F	QueryDigitalAudioSensorConfiguration	false
Output Set		
0801	ConfirmSensorConfiguration	false
4817	ReportDigitalAudioSensorCapabilities	false
480F	ReportDigitalAudioSensorConfiguration	false

4.3.5 Encoding

4.3.5.1 Input Set

4.3.5.1.1 ID 080F: SetDigitalAudioSensorConfiguration

This message is used to set the current configuration for one or more audio sensors. Each Set message contains a local request ID; this ID is returned by the corresponding SetDigitalAudioSensorConfigurationResponse message and may be used by the client to coordinate requests and responses.

Table 21 - SetDigitalAudioSensorConfiguration message encoding

<pre> body ├── sequence name = DigitalAudioSensorConfigurationSeq │ ├── record name = RequestIdRec │ │ └── list name = DigitalAudioSensorConfigurationList │ │ (count_field = unsigned short integer) │ │ └── record name = DigitalAudioSensorConfigurationRec </pre>					
Record Name = RequestIdRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestId	unsigned byte	one	false	
Record Name = DigitalAudioSensorConfigurationRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
3	<fixed_field> Sensitivity	unsigned short integer	one	true	The gain or volume applied to the input as a percent. A value of 0% indicates that no output signal is generated, while a value of 100% indicates that maximum amplification is applied to the input. (scaled range = [0,100], round)
4	<fixed_field> MinimumBitRate	unsigned short integer	one	true	kilobits per second. Different min and max bitrates may result in a Variable Bit Rate (VBR) stream, if supported.
5	<fixed_field> MaximumBitRate	unsigned short integer	one	true	kilobits per second. Different min and max bitrates may result in a Variable Bit Rate (VBR) stream, if supported.

6	<fixed_field> DigitalFormat	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= LPCM_PCM 1= AIFF 2= WAV 3= ALAC 4= FLAC 5= RealAudio 6= WMA9_LOSSLESS 7= TrueAudio 8= DolbyDigital 9= DTS 10= MP2 11= MP3 12= AAC_MPEG2 13= AAC_MPEG4 14= VORBIS 15= WMA 16= Speex
7	<fixed_field> SampleRate	unsigned integer	one	true	Samples per second
8	<fixed_field> BitDepth	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= EIGHT_BITS 1= TEN_BITS 2= TWELVE_BITS 3= FOURTEEN_BITS 4= SIXTEEN_BITS 5= TWENTY_BITS 6= TWENTY_FOUR_BITS 7= THIRTY_TWO_BITS
9	<fixed_field> EncodingQuality	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Best 1= Better 2= Good 3= Average 4= Adequate 5= Poor 6= Worst

4.3.5.1.2 ID 2817: QueryDigitalAudioSensorCapabilities

This message is used to query the full set of capabilities (supported performance levels) for one or more sensors. A SensorID of zero may be used to query all sensors supported by the service in a single message.

Table 22 - QueryDigitalAudioSensorCapabilities message encoding

body └ record name = QueryDigitalAudioSensorRec					
Record Name = QueryDigitalAudioSensorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried

4.3.5.1.3 ID 280F: QueryDigitalAudioSensorConfiguration

This message is used to query the current configuration (active performance level) for one or more sensors. A SensorID of zero may be used to query all sensors supported by the service in a single message.

Table 23 - QueryDigitalAudioSensorConfiguration message encoding

body └ record name = QueryDigitalAudioSensorRec					
Record Name = QueryDigitalAudioSensorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried

4.3.5.2 Output Set

4.3.5.2.1 ID 0801: ConfirmSensorConfiguration

This message is used to notify a client component that the configuration has been received with the values specified in the corresponding set message with Request ID matching the value of field 1 of this message. If the specified configuration request is deemed valid, the SensorIdRec is returned with the matching SensorID (or illuminatorID) of the sensor (or illuminator) for which the configuration was successfully set. If the requested configuration is invalid, one of the ErrorRec types shall be returned (depending on the source message) with an error code and description of the configuration setting which was deemed invalid.

Table 24 - ConfirmSensorConfiguration message encoding

body └ sequence name = ConfirmSensorConfigurationSequence └ record name = RequestIdRec └ list name = ConfirmSensorList (count_field = unsigned short integer) └ variant name = ConfirmSensorConfigurationVariant (vtag_field = unsigned byte) ├ record name = SensorIdRec ├ record name = RangeSensorErrorRec ├ record name = VisualSensorErrorRec ├ record name = DigitalVideoSensorErrorRec ├ record name = AnalogVideoSensorErrorRec ├ record name = StillImageSensorErrorRec ├ record name = H264VideoEncodingErrorRec ├ record name = DigitalAudioSensorErrorRec ├ record name = DigitalAudioOutputErrorRec └ record name = VideoIlluminatorErrorRec					
Record Name = RequestIdRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestID	unsigned byte	one	false	
Record Name = SensorIdRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
Record Name = RangeSensorErrorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]

2	<fixed_field> RangeSensorErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Horizontal Field of View 2= Invalid Vertical Field of View 3= Invalid Update Rate 4= Invalid Sensor Range 5= Invalid Sensor State 6= Multiple Invalid Parameters 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = VisualSensorErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> VisualSensorErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Sensor State 2= Invalid Zoom Mode 3= Invalid Zoom Value 4= Invalid Focus Mode 5= Invalid Focus Value 6= Invalid White Balance 7= Invalid Imaging Mode 8= Invalid Exposure Mode 9= Invalid Metering Mode 10= Invalid Shutter Speed 11= Invalid Aperture Value 12= Invalid Light Sensitivity 13= Invalid Image Stabilization 14= Invalid Horizontal FOV 15= Invalid Vertical FOV 16= Multiple Invalid Parameters 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = DigitalVideoSensorErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> DigitalVideoErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Minimum Bit Rate 2= Invalid Maximum Bit Rate 3= Requested Frame Rate Too Low 4= Requested Frame Rate Too High 5= Invalid Frame Size 6= Invalid Format 7= Multiple Invalid Parameters 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = AnalogVideoSensorErrorRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> AnalogVideoErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Format 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)



Record Name = StillImageSensorErrorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> StillImageErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Frame Size 2= Invalid Format 3= Multiple Invalid Parameters 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
Record Name = H264VideoEncodingErrorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> H264VideoEncodingErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid Profile 2= Invalid Preset 3= Invalid GroupOfPictures 4= Invalid GDR 5= Invalid RegionOfInterest 6= Invalid IntraMacroRefresh 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
Record Name = DigitalAudioSensorErrorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> DigitalAudioSensorErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Invalid BitRate 2= Invalid Format 3= Invalid SampleRate 4= Invalid BitDepth 5= Invalid EncodingQuality 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
Record Name = DigitalAudioOutputErrorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> DigitalAudioOutputErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Stream Not Found 2= Stream Not Supported 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
Record Name = VideoIlluminatorErrorRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> IlluminatorID	unsigned short integer	one	false	

2	<fixed_field> VideoIlluminatorErrorRecCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Unknown Sensor ID 1= Unsupported Mode 2= Unsupported Beam Width 3= Unsupported Beam Height 255= Unknown Error / Fault
3	<variable_length_string> ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

4.3.5.2.2 ID 4817: ReportDigitalAudioSensorCapabilities

This message is used to report full set of capabilities (supported performance levels) for one or more sensors.

Table 25 - ReportDigitalAudioSensorCapabilities message encoding

<pre> body └─ list name = DigitalAudioSensorCapabilitiesList (count_field = unsigned short integer) └─ record name = DigitalAudioSensorCapabilitiesRec </pre>					
Record Name = DigitalAudioSensorCapabilitiesRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
3	<fixed_field> MinimumAllowedBitRate	unsigned short integer	one	true	kilobits per second
4	<fixed_field> MaximumAllowedBitRate	unsigned short integer	one	true	kilobits per second

5	<bit_field> SupportedDigitalFormats	unsigned integer	bit_field	true	Bits 0..0, LPCM_PCM: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 1..1, AIFF: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 2..2, WAV: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 3..3, ALAC: Value set, offset=false, ranges/enums: 1= SUPPORTED 0= NOT_SUPPORTED Bits 4..4, FLAC: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 5..5, RealAudio: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 6..6, WMA9_Lossless: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 7..7, TrueAudio: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 8..8, DolbyDigital: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 9..9, DTS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 10..10, MP2: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 11..11, MP3: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 12..12, AAC_MPEG2: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 13..13, AAC_MPEG4: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 14..14, VORBIS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 15..15, WMA: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 16..16, Speex: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED
6	<fixed_field> MinSampleRate	unsigned integer	one	true	Samples per second
7	<fixed_field> MaxSampleRate	unsigned integer	one	true	Samples per second



8	<bit_field> SupportedBitDepths	unsigned short integer	bit_field	true	Bits 0..0, EIGHT_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 1..1, TEN_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 2..2, TWELVE_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 3..3, FOURTEEN_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 4..4, SIXTEEN_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 5..5, TWENTY_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 6..6, TWENTY_FOUR_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 7..7, THIRTY_TWO_BITS: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED
9	<bit_field> SupportedQualityLevels	unsigned byte	bit_field	true	Bits 0..0, Best: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 1..1, Better: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 2..2, Good: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 3..3, Average: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 4..4, Adequate: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 5..5, Poor: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 6..6, Worst: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED

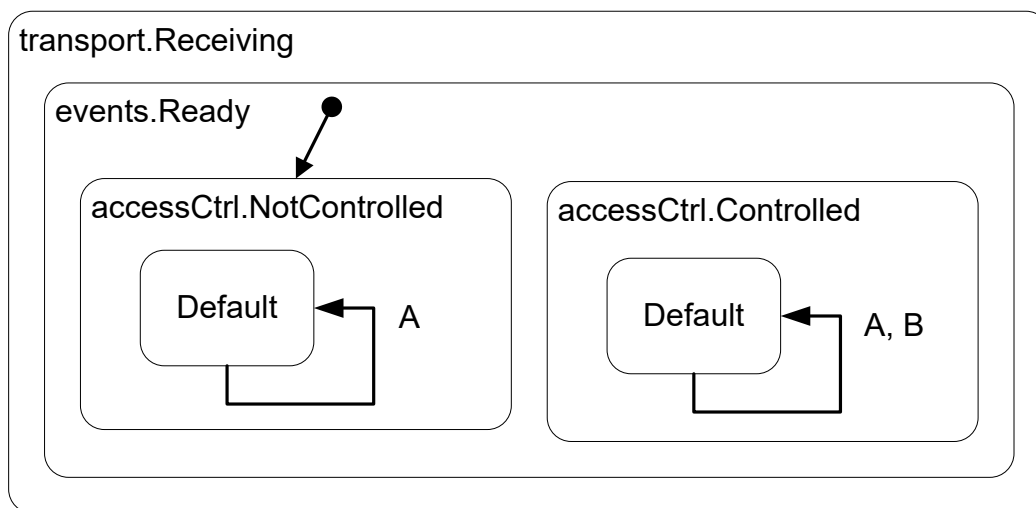
4.3.5.2.3 ID 480F: ReportDigitalAudioSensorConfiguration

This message is used to report the current configuration for one or more audio sensors.

Table 26 - ReportDigitalAudioSensorConfiguration message encoding

<pre> body └ list name = DigitalAudioSensorConfigurationList (count_field = unsigned short integer) └ record name = DigitalAudioSensorConfigurationRec </pre>					
Record Name = DigitalAudioSensorConfigurationRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
3	<fixed_field> Sensitivity	unsigned short integer	one	true	The gain or volume applied to the input as a percent. A value of 0% indicates that no output signal is generated, while a value of 100% indicates that maximum amplification is applied to the input. (scaled range = [0,100], round)
4	<fixed_field> MinimumBitRate	unsigned short integer	one	true	kilobits per second. Different min and max bitrates may result in a Variable Bit Rate (VBR) stream, if supported.
5	<fixed_field> MaximumBitRate	unsigned short integer	one	true	kilobits per second. Different min and max bitrates may result in a Variable Bit Rate (VBR) stream, if supported.
6	<fixed_field> DigitalFormat	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= LPCM_PCM 1= AIFF 2= WAV 3= ALAC 4= FLAC 5= RealAudio 6= WMA9_LOSSLESS 7= TrueAudio 8= DolbyDigital 9= DTS 10= MP2 11= MP3 12= AAC_MPEG2 13= AAC_MPEG4 14= VORBIS 15= WMA 16= Speex
7	<fixed_field> SampleRate	unsigned integer	one	true	Samples per second
8	<fixed_field> BitDepth	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= EIGHT_BITS 1= TEN_BITS 2= TWELVE_BITS 3= FOURTEEN_BITS 4= SIXTEEN_BITS 5= TWENTY_BITS 6= TWENTY_FOUR_BITS 7= THIRTY_TWO_BITS
9	<fixed_field> EncodingQuality	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Best 1= Better 2= Good 3= Average 4= Adequate 5= Poor 6= Worst

4.3.6 Protocol Behavior

**Figure 3 - DigitalAudioSensor service protocol behavior****Table 27 - DigitalAudioSensor service state transitions**

Start State	Trigger	Conditions	Actions
A	QueryDigitalAudioSensorCapabilities		sendReportDigitalAudioSensorCapabilities (msg, transportData)
A	QueryDigitalAudioSensorConfiguration		sendReportDigitalAudioSensorConfiguration (msg, transportData)
B	SetDigitalAudioSensorConfiguration	isControllingClient(transportData)	setConfiguration (msg) , sendConfirmSensorConfiguration (msg, transportData)

Table 28 - DigitalAudioSensor service conditions

Condition	Interpretation
isControllingClient(transportData)	True if the command message was received from the client currently controlling this component.

Table 29 - DigitalAudioSensor service transition actions

Action	Interpretation
sendReportDigitalAudioSensorCapabilities	Send a ReportDigitalAudioSensorCapabilities message to querying client
sendReportDigitalAudioSensorConfiguration	Send a ReportDigitalAudioSensorConfiguration message to querying client
setConfiguration	Update the settings for any sensor IDs with valid and supported configurations specified in the message
sendConfirmSensorConfiguration	Send sendConfirmSensorConfiguration message with confirmation or error code for each specified sensor ID

4.4 DigitalResourceDiscovery

name=DigitalResourceDiscovery
version=2.0
id=urn:jaus:jss:environmentSensing:DigitalResourceDiscovery

4.4.1 Description

The Digital Resource Discovery service provides a mechanism for SAE JAUS-based components to discover network entities that transmit digital data streams (usually video and/or audio) and files in a standards-compliant format. Because of the wide-spread support for numerous file transfer and streaming standards, this service does not propose a JAUS-specific format for data; it only provides a discovery mechanism based on a Uniform Resource Locator (URL).

4.4.2 Assumptions

Messages may be delayed, lost, or reordered.

4.4.3 References

Inherits-from=Events
version=1.1
id=urn:jaus:jss:core:Events

4.4.4 Vocabulary

Table 30 - DigitalResourceDiscovery service message vocabulary

Message ID (hex)	Name	Command
Input Set		
2816	QueryDigitalResourceEndpoint	false
0808	RegisterDigitalResourceEndpoint	true
0809	RemoveDigitalResourceEndpoint	true
Output Set		
4816	ReportDigitalResourceEndpoint	false
080A	ConfirmDigitalResourceEndpoint	false

4.4.5 Encoding

4.4.5.1 Input Set

4.4.5.1.1 ID 2816: QueryDigitalResourceEndpoint

Queries for a list of known digital resource endpoints.

Table 31 - QueryDigitalResourceEndpoint message encoding

body (empty)
Empty message body

4.4.5.1.2 ID 0808: RegisterDigitalResourceEndpoint

Registers a digital resource server with the service. Each endpoint is represented by a URL; however, the URL shall not require a Domain Name Service (DNS) to resolve. In addition, each stream may also specify a JAUS ID that hosts additional SAE JAUS services for the configuration and control of the digital resource, as well as a ResourceID that identifies the stream source.

Table 32 - RegisterDigitalResourceEndpoint message encoding

<pre> body └─ sequence name = RegisterDigitalResourceSeq └─ record name = RequestIDRec └─ record name = DigitalResourceEndpointRec </pre>					
Record Name = RequestIDRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestID	unsigned byte	one	false	Client provided ID to link the response to the request
Record Name = DigitalResourceEndpointRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> ServerType	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= RTSP 1= MPEG2_TS 2= FTP 3= SFTP 4= FTP_over_SSH 5= HTTP 6= HTTPS 7= SCP 8= CCSI (where URL is of the form 'ip address:port number' (no quotes)) [9,200] Reserved for future use [201,255] Reserved for program or implementation specific use
3	<variable_length_string> ServerURL	variable length string (byte[])	N/A	false	URL (or URL-like descriptor in the case of CCSI) of the digital resource server. This should not require a DNS to resolve; hence, an IP address should be substituted for a host name. (Length min..max = 0..255)
4	<bit_field> JAUS_ID	unsigned integer	bit_field	true	Bits 0..7, ComponentID: Value set, offset=false, ranges/enums: [0,255] Bits 8..15, NodeID: Value set, offset=false, ranges/enums: [0,255] Bits 16..31, SubsystemID: Value set, offset=false, ranges/enums: [0,65535]
5	<fixed_field> ResourceID	unsigned short integer	one	true	The ID used by the configuration and control service to identify this source. This is the SensorID for visual sensors.

4.4.5.1.3 ID 0809: RemoveDigitalResourceEndpoint

Remove a previously registered digital resource transfer server from the service.

Table 33 - RemoveDigitalResourceEndpoint message encoding

body └ record name = RemoveDigitalResourceEndpointRec					
Record Name = RemoveDigitalResourceEndpointRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> ID	unsigned byte	one	false	Unique ID that was returned in the confirm message
2	<fixed_field> RequestID	unsigned byte	one	false	Client provided ID to link the response to the request

4.4.5.2 Output Set

4.4.5.2.1 ID 4816: ReportDigitalResourceEndpoint

Reports a list of known digital resource servers. Each endpoint is represented by a URL; however, the URL shall not require a Domain Name Service (DNS) to resolve.

Table 34 - ReportDigitalResourceEndpoint message encoding

body └ list name = DigitalResourceEndpointList (count_field = unsigned byte) └ record name = DigitalResourceEndpointRec					
Record Name = DigitalResourceEndpointRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> ServerType	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= RTSP 1= MPEG2_TS 2= FTP 3= SFTP 4= FTP_over_SSH 5= HTTP 6= HTTPS 7= SCP 8= CCSI (where URL is of the form 'ip address:port number' (no quotes)) [9,200] Reserved for future use [201,255] Reserved for program or implementation specific use
3	<variable_length_string> ServerURL	variable length string (byte[])	N/A	false	URL (or URL-like descriptor in the case of CCSI) of the digital resource server. This should not require a DNS to resolve; hence, an IP address should be substituted for a host name. (Length min..max = 0..255)
4	<bit_field> JAUS_ID	unsigned integer	bit_field	true	Bits 0..7, ComponentID: Value set, offset=false, ranges/enums: [0,255] Bits 8..15, NodeID: Value set, offset=false, ranges/enums: [0,255] Bits 16..31, SubsystemID: Value set, offset=false, ranges/enums: [0,65535]
5	<fixed_field> ResourceID	unsigned short integer	one	true	The ID used by the configuration and control service to identify this source. This is the SensorID for visual sensors.

4.4.5.2.2 ID 080A: ConfirmDigitalResourceEndpoint

Confirm a digital resource endpoint registration or removal. Provides a unique ID for referencing the server in the future.

Table 35 - ConfirmDigitalResourceEndpoint message encoding

body └ record name = ConfirmDigitalResourceEndpointRec					
Record Name = ConfirmDigitalResourceEndpointRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> ID	unsigned byte	one	false	Unique ID identifying this resource endpoint
2	<fixed_field> RequestID	unsigned byte	one	false	Client provided ID to link the response to the request

4.4.6 Protocol Behavior

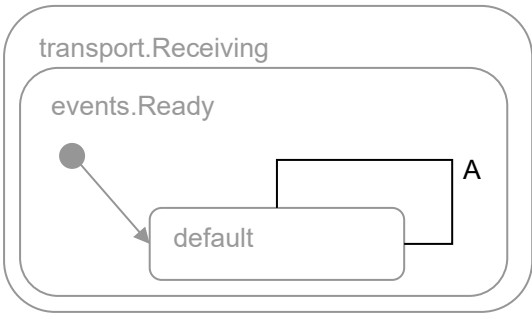


Figure 4 - DigitalResourceDiscovery service protocol behavior

Table 36 - DigitalResourceDiscovery service state transitions

Start State	Trigger	Conditions	Actions
A	QueryDigitalResourceEndpoint		sendReportDigitalResourceEndpoint (msg, transportData)
A	RegisterDigitalResourceEndpoint		AddDigitalResourceEndpoint (msg,transportData) , sendConfirmDigitalResourceEndpoint (msg, transportData)
A	RemoveDigitalResourceEndpoint		RemoveDigitalResourceEndpoint(msg, transportData), sendConfirmDigitalResourceEndpoint (msg, transportData)

Table 37 - DigitalResourceDiscovery service transition actions

Action	Interpretation
sendReportDigitalResourceEndpoint	Send a ReportDigitalResourceEndpoint message to querying client
AddDigitalResourceEndpoint	Adds the specified endpoint to the list of known endpoints
sendConfirmDigitalResourceEndpoint	Send a ConfirmDigitalResourceEndpoint message to querying client
RemoveDigitalResourceEndpoint	Removes the specified endpoint from the list of known endpoints
sendConfirmDigitalResourceEndpoint	Send a ConfirmDigitalResourceEndpoint message to querying client

4.5 DigitalVideo

name=DigitalVideo

version=2.0

id=urn:jaus:jss:environmentSensing:DigitalVideo

4.5.1 Description

This service provides access to the capabilities and configuration of the digital visual sensor, allowing the controlling component to set the visual sensor to a particular operational profile. The actual transmission of the video stream is outside the scope of this service. The ability to start, stop and pause the video stream is provided in the message protocol. There may also be mechanisms in the chosen video transmission protocol to control the video stream. In such situations, the messages defined herein are redundant and either mechanism may be used by sensor's client.

4.5.2 Assumptions

Messages may be delayed, lost, or reordered.

4.5.3 References

Inherits-from=visualSensor

version=2.0

id=urn:jaus:jss:environmentSensing:VisualSensor

4.5.4 Vocabulary

Table 38 - DigitalVideo service message vocabulary

Message ID (hex)	Name	Command
Input Set		
0805	ControlDigitalVideoSensorStream	true
0810	ControlDigitalVideoSensorRecording	true
0804	SetDigitalVideoSensorConfiguration	true
2809	QueryDigitalVideoSensorConfiguration	false
281D	QueryDigitalVideoSensorCapabilitiesExt	false
281E	QueryDigitalVideoSensorStatus	false
Output Set		
481D	ReportDigitalVideoSensorCapabilitiesExt	false
4809	ReportDigitalVideoSensorConfiguration	false
481E	ReportDigitalVideoSensorStatus	false

4.5.5 Encoding

4.5.5.1 Input Set

4.5.5.1.1 ID 0805: ControlDigitalVideoSensorStream

This message is used to control the playback state of the video stream from a digital video service. The actual stream protocol for this is outside the scope of the protocol. The streaming mechanism selected may support other methods to control the stream within its own protocol. In such a case, this message shall be a redundant mechanism and a service client may choose to use either the native protocol or this message for stream control.

Table 39 - ControlDigitalVideoSensorStream message encoding

body └ record name = ControlDigitalVideoSensorStreamRec					
Record Name = ControlDigitalVideoSensorStreamRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> StreamState	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Play 1= Pause 2= Stop

4.5.5.1.2 ID 0810: ControlDigitalVideoSensorRecording

This message is used to control the recording state of the video from a digital video service. The details of the recording (filename, location, maximum file size, etc.) are left to the implementation.

Table 40 - ControlDigitalVideoSensorRecording message encoding

body └ record name = ControlDigitalVideoSensorRecordingRec					
Record Name = ControlDigitalVideoSensorRecordingRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> RequestedRecordingState	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= NotRecording 1= Recording

4.5.5.1.3 ID 0804: SetDigitalVideoSensorConfiguration

This message is used to set the configuration of the digital video sensor associated with the service. Configuration is based off of each sensor's capabilities as described in the Report Digital Video Sensor Capabilities message. This message shall cause the receiving service to reply to the sender with a Confirm Sensor Configuration message. If the configuration specified is invalid for a given sensor ID, the confirm message shall contain a Digital Video Error Record for the given Sensor ID however other, valid, configurations specified shall be set (if they exist).

Table 41 - SetDigitalVideoSensorConfiguration message encoding

<pre> body └─ sequence name = DigitalVideoSensorConfigurationSequence └─ record name = RequestIdRec └─ list name = DigitalVideoSensorList (count_field = unsigned short integer) └─ record name = DigitalVideoSensorConfigurationRec </pre>					
Record Name = RequestIdRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestId	unsigned byte	one	false	
Record Name = DigitalVideoSensorConfigurationRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
3	<fixed_field> MinimumBitRate	unsigned short integer	one	true	measured kilobits per second
4	<fixed_field> MaximumBitRate	unsigned short integer	one	true	measured kilobits per second
5	<fixed_field> FrameRate	unsigned byte	one	true	
6	<fixed_field> FrameSize	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= 128x96 (sqcif) 1= 176x144 (qcif) 2= 352x288 (cif) 3= 704x576 (4cif) 4= 1408x1152 (16cif) 5= 160x120 (qqvga) 6= 320x240 (qvga) 7= 640x480 (vga) 8= 800x600 (svga) 9= 1024x768 (xga) 10= 1600x1200 (uxga) 11= 2048x1536 (qxga) 12= 1280x1024 (sxga) 13= 2560x2048 (qsxga) 14= 5120x4096 (hsxga) 15= 852x480 (wvga) 16= 1366x768 (wxga) 17= 1600x1024 (wsxga) 18= 1920x1200 (wuxga) 19= 2560x1600 (woxga) 20= 3200x2048 (wqsxga) 21= 3840x2400 (wquxga) 22= 6400x4096 (whsxga) 23= 7680x4800 (whuxga) 24= 320x200 (cga) 25= 640x350 (ega) 26= 852x480 (hd480) 27= 1280x720 (hd720) 28= 1920x1080 (hd1080)
7	<fixed_field> DigitalFormat	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= AVI 1= MJPEG 2= MPEG-2 3= H.263 4= H.263+ 5= MPEG-4 Visual (MPEG-4 Part 2) 6= MPEG-4 AVC

4.5.5.1.4 ID 2809: QueryDigitalVideoSensorConfiguration

This message shall cause the receiving service to reply to the requestor with a Report Digital Video Sensor Configuration message. A logical AND shall be performed on the requested presence vector and that representing the available fields from the responder. The resulting message shall contain the fields indicated by the result of this logical AND operation.

Table 42 - QueryDigitalVideoSensorConfiguration message encoding

<pre> body └─ list name = QueryDigitalVideoConfigurationList └─ (count_field = unsigned short integer) └─ record name = QueryDigitalVideoConfigurationRec </pre>					
Record Name = QueryDigitalVideoConfigurationRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried
2	<fixed_field> QueryPresenceVector	unsigned byte	one	false	

4.5.5.1.5 ID 281D: QueryDigitalVideoSensorCapabilitiesExt

This message shall cause the receiving service to reply to the requestor with a Report Digital Video Sensor Capabilities Ext message. A logical AND shall be performed on the requested presence vector and that representing the available fields from the responder. The resulting message shall contain the fields indicated by the result of this logical AND operation.

Table 43 - QueryDigitalVideoSensorCapabilitiesExt message encoding

<pre> body └─ list name = QueryDigitalVideoSensorCapabilitiesList └─ (count_field = unsigned short integer) └─ record name = QueryDigitalVideoSensorCapabilitiesRec </pre>					
Record Name = QueryDigitalVideoSensorCapabilitiesRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried
2	<fixed_field> QueryPresenceVector	unsigned byte	one	false	

4.5.5.1.6 ID 281E: QueryDigitalVideoSensorStatus

This message shall cause the receiving service to reply to the requestor with a Report Digital Video Sensor Status message.

Table 44 - QueryDigitalVideoSensorStatus message encoding

<pre> body └─ list name = SensorIDList (count_field = unsigned short integer) └─ record name = SensorIDQueryRec </pre>					
Record Name = SensorIDQueryRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with this service. Value set, offset=false, ranges/enums: 0= AllSensors [1,65535] Specific ID to be queried

4.5.5.2 Output Set

4.5.5.2.1 ID 481D: ReportDigitalVideoSensorCapabilitiesExt

This message is used to report the sensors' capabilities upon receipt of a Query Digital Video Sensor Capabilities Ext message. Capabilities include sensor properties, values and ranges which can be modified by the Set Digital Video Sensor Configuration message.

Table 45 - ReportDigitalVideoSensorCapabilitiesExt message encoding

<pre> body └─ list name = DigitalVideoSensorList (count_field = unsigned short integer) └─ record name = DigitalVideoSensorCapabilitiesRec </pre>					
Record Name = DigitalVideoSensorCapabilitiesRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
3	<fixed_field> MinimumBitRate	unsigned short integer	one	true	measured kilobits per second
4	<fixed_field> MaximumBitRate	unsigned short integer	one	true	measured kilobits per second
5	<fixed_field> MinimumFrameRate	unsigned byte	one	true	
6	<fixed_field> MaximumFrameRate	unsigned byte	one	true	

7	<bit_field> SupportedFrameSizes	unsigned integer	bit_field	true	Bits 0..0, sqcif_128x96: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, qcif_176x144: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, cif_352x288: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, cif4_704x576: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, cif16_1408x1152: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, qqvga_160x120: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, qvga_320x240: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 7..7, vga_640x480: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 8..8, svga_800x600: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 9..9, xga_1024x768: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 10..10, uxga_1600x1200: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 11..11, qxga_2048x1536: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 12..12, sxga_1280x1024: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 13..13, qsxga_2560x2048: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 14..14, hsxga_5120x4096: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 15..15, wvga_852x480: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 16..16, wxga_1366x768: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 17..17, wsxga_1600x1024: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 18..18, wuxga_1920x1200: Value set, offset=false, ranges/enums:
---	------------------------------------	------------------	-----------	------	---

					0= Unsupported 1= Supported Bits 19..19, woxga_2560x1600: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 20..20, wqsga_3200x2048: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 21..21, wqsga_3840x2400: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 22..22, whsga_6400x4096: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 23..23, whsga_7680x4800: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 24..24, cga_320x200: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 25..25, ega_640x350: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 26..26, hd480_852x480: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 27..27, hd720_1280x720: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 28..28, hd1080_1920x1080: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
8	<bit_field> SupportedDigitalFormats	unsigned byte	bit_field	true	Bits 0..0, AVI: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, MJPEG: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, MPEG2: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, h263: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, h263plus: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, MPEG4_Visual: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, MPEG4_AVC_h264: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported

9	<fixed_field> RecordingSupported	unsigned byte	one	true	Indicates if the implementation supports local recording of video for the sensor Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
---	--	---------------	-----	------	--

4.5.5.2.2 ID 4809: ReportDigitalVideoSensorConfiguration

This message is sent in response to a Query Digital Video Sensor Configuration message. It is populated with the current sensor configuration (per sensor ID) as defined in the table below.

Table 46 - ReportDigitalVideoSensorConfiguration message encoding

<pre> body └─ list name = DigitalVideoSensorConfigurationList (count_field = unsigned short integer) └─ record name = DigitalVideoSensorConfigurationRec </pre>					
Record Name = DigitalVideoSensorConfigurationRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
3	<fixed_field> MinimumBitRate	unsigned short integer	one	true	measured kilobits per second
4	<fixed_field> MaximumBitRate	unsigned short integer	one	true	measured kilobits per second
5	<fixed_field> FrameRate	unsigned byte	one	true	
6	<fixed_field> FrameSize	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= 128x96 (sqcif) 1= 176x144 (qcif) 2= 352x288 (cif) 3= 704x576 (4cif) 4= 1408x1152 (16cif) 5= 160x120 (qqvga) 6= 320x240 (qvga) 7= 640x480 (vga) 8= 800x600 (svga) 9= 1024x768 (xga) 10= 1600x1200 (uxga) 11= 2048x1536 (qxga) 12= 1280x1024 (sxga) 13= 2560x2048 (qsxga) 14= 5120x4096 (hsxga) 15= 852x480 (wvga) 16= 1366x768 (wxga) 17= 1600x1024 (wsxga) 18= 1920x1200 (wuxga) 19= 2560x1600 (woxga) 20= 3200x2048 (wqsxga) 21= 3840x2400 (wquxga) 22= 6400x4096 (whsxga) 23= 7680x4800 (whuxga) 24= 320x200 (cga) 25= 640x350 (ega) 26= 852x480 (hd480) 27= 1280x720 (hd720) 28= 1920x1080 (hd1080)

7	<fixed_field> DigitalFormat	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= AVI 1= MJPEG 2= MPEG-2 3= H.263 4= H.263+ 5= MPEG-4 Visual (MPEG-4 Part 2) 6= MPEG-4 AVC
---	--------------------------------	---------------	-----	------	---

4.5.5.2.3 ID 481E: ReportDigitalVideoSensorStatus

This message is sent in response to a Query Digital Video Sensor Status message. It is populated with the current sensor status (per sensor ID) as defined in the table below.

Table 47 - ReportDigitalVideoSensorStatus message encoding

body └─ list name = DigitalVideoSensorStatusList └─ (count_field = unsigned short integer) └─ record name = DigitalVideoSensorStatusRec					
Record Name = DigitalVideoSensorStatusRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]
2	<fixed_field> StreamingState	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Play 1= Pause 2= Stop
3	<fixed_field> RecordingState	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= NotRecording 1= Recording

4.5.6 Protocol Behavior

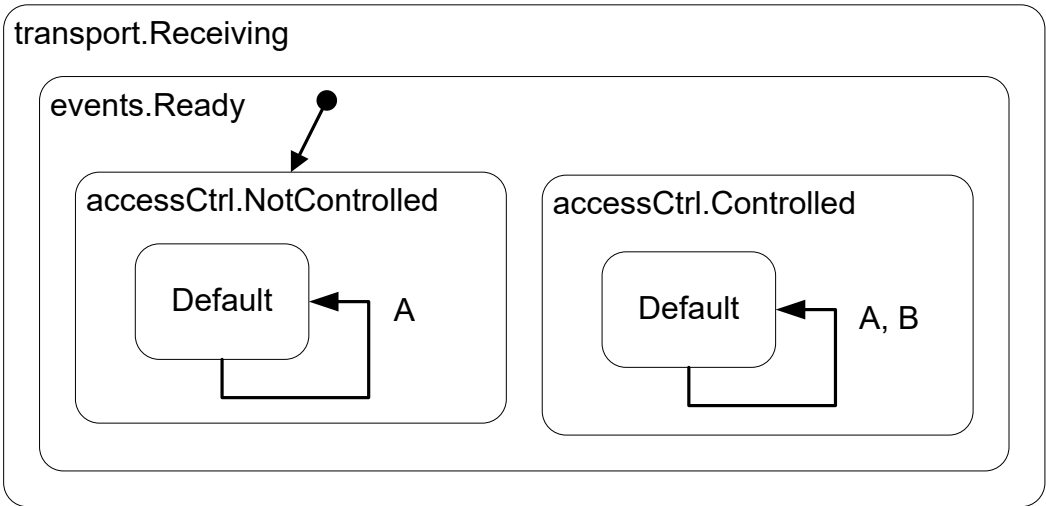


Figure 5 - DigitalVideo service protocol behavior

Table 48 - DigitalVideo service state transitions

Start State	Trigger	Conditions	Actions
A	QueryDigitalVideoSensorCapabilitiesExt		sendReportDigitalVideoSensorCapabilitiesExt (msg,transportData)
A	QueryDigitalVideoSensorConfiguration		sendReportDigitalVideoSensorConfiguration (msg,transportData)
A	QueryDigitalVideoSensorStatus		sendReportDigitalVideoSensorStatus (msg, transportData)
B	SetDigitalVideoSensorConfiguration	isControllingClient(transportData)	sendConfirmSensorConfiguration (msg, transportData) , updateDigitalVideoSensorConfiguration (msg)
B	ControlDigitalVideoSensorStream	isControllingClient(transportData)	modifyDigitalVideoSensorStream (msg)
B	ControlDigitalVideoSensorRecording	isControllingClient(transportData) && isSupported(msg)	updateDigitalVideoSensorRecordingState (msg)

Table 49 - DigitalVideo service conditions

Condition	Interpretation
isControllingClient(transportData)	True if the message that triggered the transition is received from the client that is in control of this service.
isControllingClient(transportData) && isSupported(msg)	True if the message that triggered the transition is received from the client that is in control of this service AND the requested recording state is supported by the implementation.

Table 50 - DigitalVideo service transition actions

Action	Interpretation
sendReportDigitalVideoSensorCapabilitiesExt	Send a ReportDigitalVideoSensorCapabilitiesExt message
sendReportDigitalVideoSensorConfiguration	Send a ReportDigitalVideoSensorConfiguration message
sendReportDigitalVideoSensorStatus	Send a ReportDigitalVideoSensorStatus message
sendConfirmSensorConfiguration	Send sendConfirmSensorConfiguration message
updateDigitalVideoSensorConfiguration	Update the sensor user controllable configuration parameters according to the ones specified.
modifyDigitalVideoSensorStream	Modify the video stream according to the specified message.
updateDigitalVideoSensorRecordingState	Update the video recording state according to the specified message.

4.6 ForceTorqueSensor

name=ForceTorqueSensor

version=2.0

id=urn:jaus:jss:environmentSensing:ForceTorqueSensor

4.6.1 Description

The Force Torque Sensor Service provides a means to get force or torque information from one or more devices. While the nominal use case is a bump sensor or collision detector, the service is suitable for other applications. The data can be reported in either the sensor coordinate system or the vehicle coordinate system (if supported).

4.6.2 Assumptions

Messages may be delayed, lost, or reordered.

4.6.3 References

Inherits-from=events

version=1.1

id=urn:jaus:jss:core:Events

4.6.4 Vocabulary

Table 51 - ForceTorqueSensor service message vocabulary

Message ID (hex)	Name	Command
Input Set		
280B	QueryForceTorqueCapabilities	false
280C	QueryForceTorque	false
Output Set		
480B	ReportForceTorqueCapabilities	false
480C	ReportForceTorque	false

4.6.5 Encoding

4.6.5.1 Input Set

4.6.5.1.1 ID 280B: QueryForceTorqueCapabilities

This message is used to query the capabilities for one or more force/torque devices.

Table 52 - QueryForceTorqueCapabilities message encoding

<pre> body └─ list name = QueryForceTorqueCapabilitiesList (count_field = unsigned short integer) └─ record name = QueryForceTorqueCapabilitiesRec </pre>					
Record Name = QueryForceTorqueCapabilitiesRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	The ID of the sensor. A value of zero means all sensors
2	<fixed_field> QueryPresenceVector	unsigned short integer	one	false	See ReportForceTorqueCapabilites

4.6.5.1.2 ID 280C: QueryForceTorque

This message is used to query the force/torque for one or more devices. Note that the requested coordinate system may not be respected if the underlying service does not support transformations, as given in the ReportForceTorqueCapabilities message.

Table 53 - QueryForceTorque message encoding

<pre> body └─ list name = QueryForceTorqueList └─ (count_field = unsigned short integer) └─ record name = QueryForceTorqueRec </pre>					
Record Name = QueryForceTorqueRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> SensorID	unsigned short integer	one	false	The ID of the sensor. A value of zero means all sensors
2	<fixed_field> CoordinateSystem	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= NativeCoordinateSystems 1= VehicleCoordinateSystems
3	<fixed_field> QueryPresenceVector	unsigned byte	one	false	See ReportForceTorqueRec

4.6.5.2 Output Set

4.6.5.2.1 ID 480B: ReportForceTorqueCapabilities

This message is used to report the capabilities of one or more force/torque sensing devices.

Table 54 - ReportForceTorqueCapabilities message encoding

<pre> body └─ list name = ReportForceTorqueCapabilitiesList └─ (count_field = unsigned short integer) └─ record name = ReportForceTorqueCapabilitiesRec </pre>					
Record Name = ReportForceTorqueCapabilitiesRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned short integer			
2	<fixed_field> SensorID	unsigned short integer	one	false	The ID of the sensor. A value of zero means all sensors
3	<variable_length_string> SensorName	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
4	<fixed_field> CoordinateTransformSupported	unsigned byte	one	true	True or false depending on if the service supports transforming data into the vehicle coordinate system. Value set, offset=false, ranges/enums: 0= False 1= True
5	<fixed_field> MinForceX	unsigned short integer	newton	true	Most negative force measurable in the X-direction. (scaled range = [-5000,5000], round)
6	<fixed_field> MaxForceX	unsigned short integer	newton	true	Most positive force measurable in the X-direction. (scaled range = [-5000,5000], round)
7	<fixed_field> MinForceY	unsigned short integer	newton	true	Most negative force measurable in the Y-direction. (scaled range = [-5000,5000], round)
8	<fixed_field> MaxForceY	unsigned short integer	newton	true	Most positive force measurable in the Y-direction. (scaled range = [-5000,5000], round)
9	<fixed_field> MinForceZ	unsigned short integer	newton	true	Most negative force measurable in the Z-direction. (scaled range = [-5000,5000], round)