AEROSPACE STANDARD

AS6060™ REV. A

Issued 2010-11
Reaffirmed 2015-04
Revised 2021-10

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.

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.

TO PLACE A DOCUMENT ORDER: **Tel: 877-606-7323 (inside USA and Canada)**
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
SAE WEB ADDRESS: **http://www.sae.org**

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

TABLE OF CONTENTS

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 examption 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

[nitf] 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

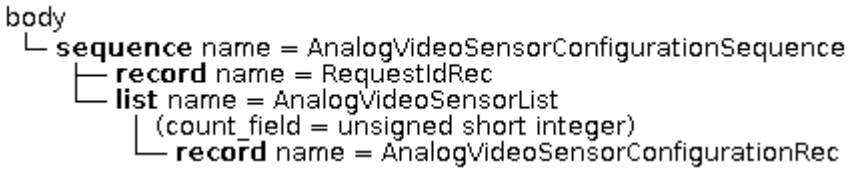
4.1.5 Encoding

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

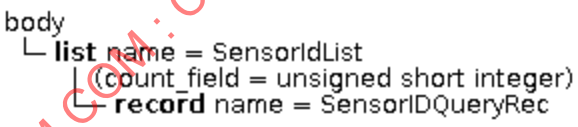


Record Name = RequestIdRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** RequestID	unsigned byte	one	false	
Record Name = AnalogVideoSensorConfigurationRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SensorID	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



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

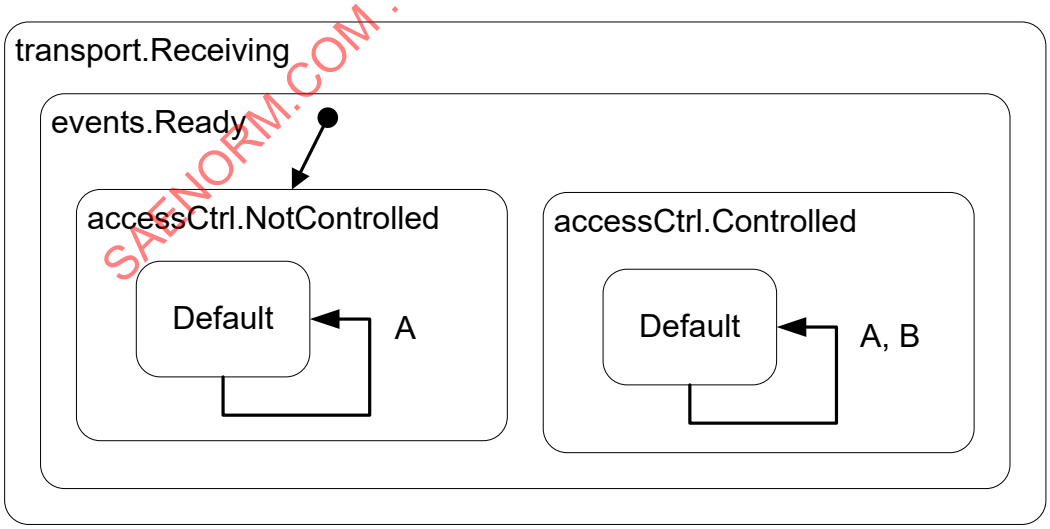


Figure 1 - AnalogVideo service protocol behavior

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

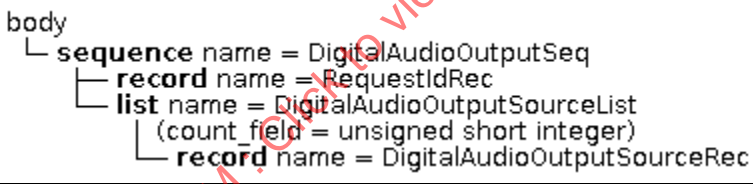
4.2.5 Encoding

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



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

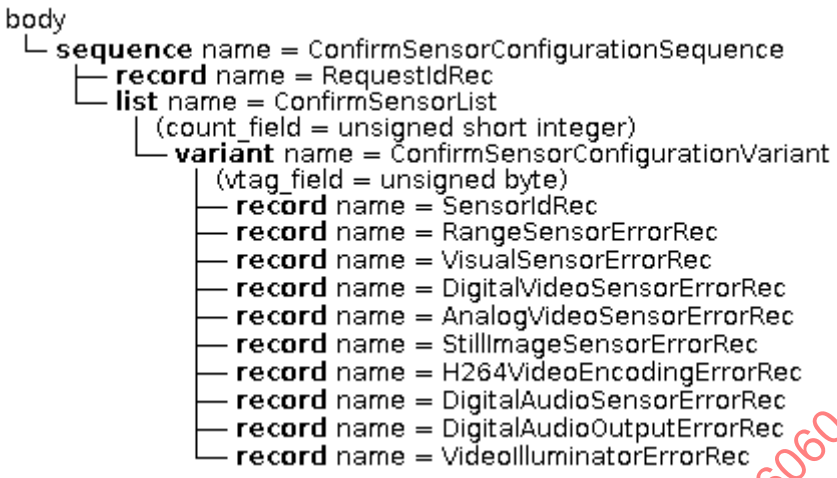
body └ **record** name = QueryDigitalAudioRec					
Record Name = QueryDigitalAudioRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SensorID	unsigned short integer	one	false	ID of the sensor being queried. Zero is used to query all sensors associated with 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



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

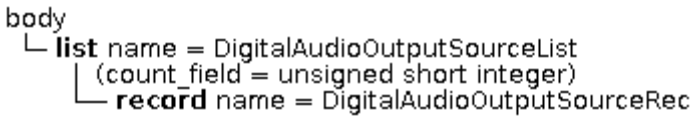
Field #	Name	Type	Units	Optional	Interpretation

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



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

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

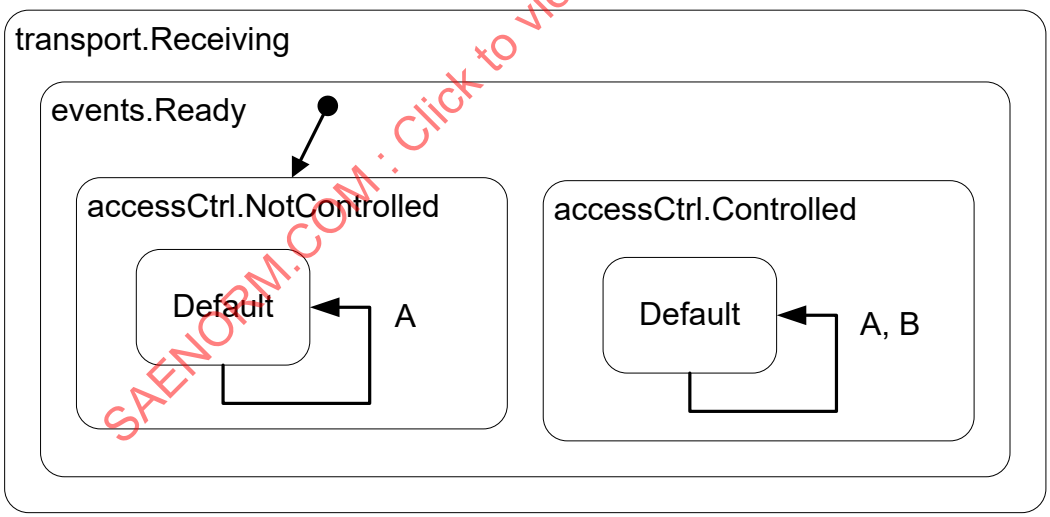


Figure 2 - DigitalAudioOutput service protocol behavior

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

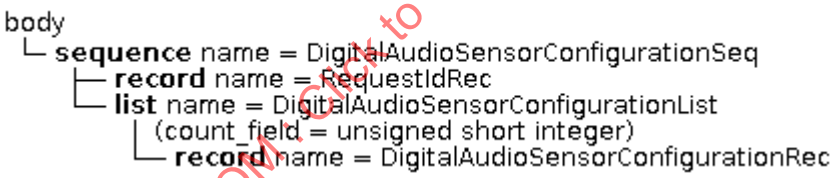
4.3.5 Encoding

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



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

body
└ **list** name = DigitalAudioSensorCapabilitiesList
  ├ (count_field = unsigned short integer)
  └ **record** name = DigitalAudioSensorCapabilitiesRec

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

body
└ **list** name = DigitalAudioSensorConfigurationList
  (count_field = unsigned short integer)
  └ **record** name = DigitalAudioSensorConfigurationRec

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

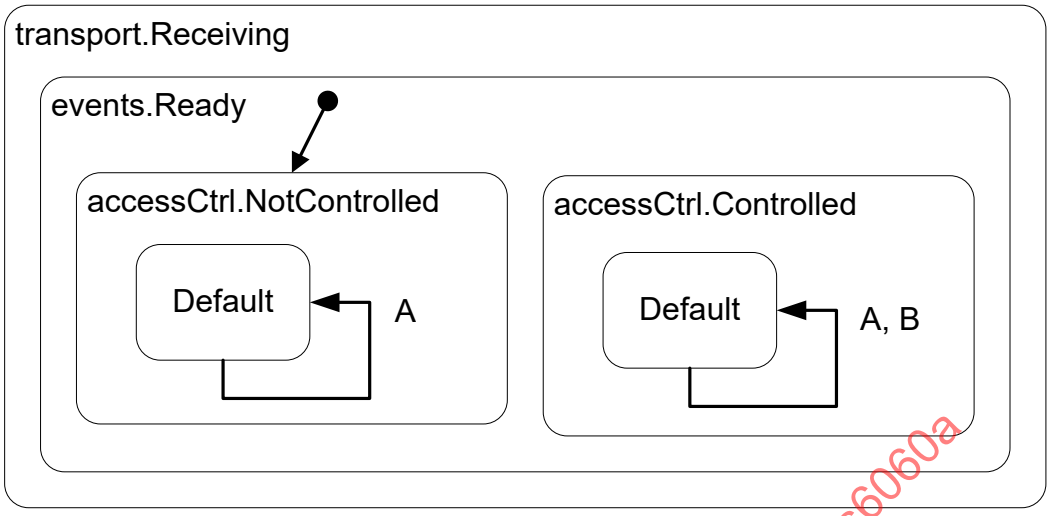


Figure 3 - DigitalAudioSensor service protocol behavior

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

body └ **sequence** name = RegisterDigitalResourceSeq ├ **record** name = RequestIDRec └ **record** name = DigitalResourceEndpointRec					
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

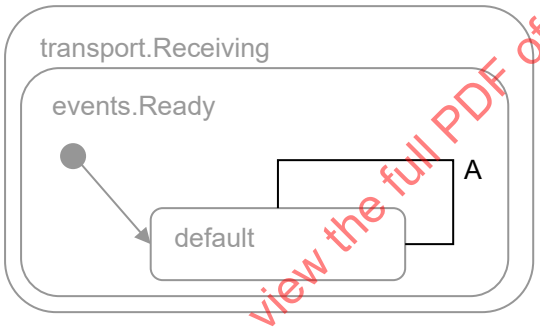


Figure 4 - DigitalResourceDiscovery service protocol behavior

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 redunant 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.

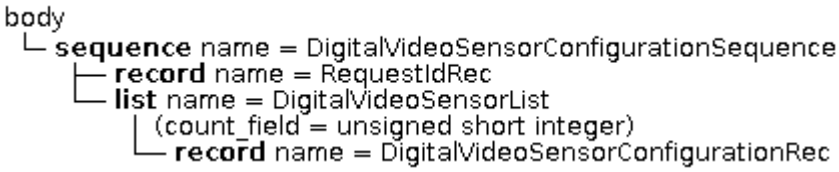
Table 40 - ControlDigitalVideoSensorRecording message encoding

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



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

body └ **list** name = QueryDigitalVideoConfigurationList │ (count_field = unsigned short integer) └ **record** name = QueryDigitalVideoConfigurationRec					
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

body └ **list** name = QueryDigitalVideoSensorCapabilitiesList │ (count_field = unsigned short integer) └ **record** name = QueryDigitalVideoSensorCapabilitiesRec					
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

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

4.5.5.2 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

body └ **list** name = DigitalVideoSensorList │ (count_field = unsigned short integer) └ **record** name = DigitalVideoSensorCapabilitiesRec					
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, wqsxga_3200x2048: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 21..21, wquxga_3840x2400: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 22..22, whsxga_6400x4096: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 23..23, whuxga_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

body
└ **list** name = DigitalVideoSensorConfigurationList
  (count_field = unsigned short integer)
  └ **record** name = DigitalVideoSensorConfigurationRec

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

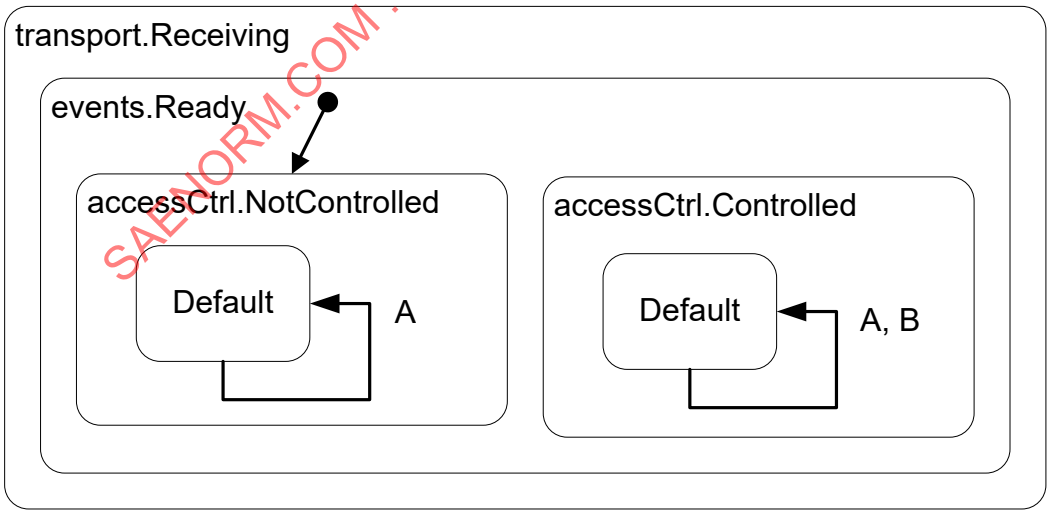


Figure 5 - DigitalVideo service protocol behavior

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

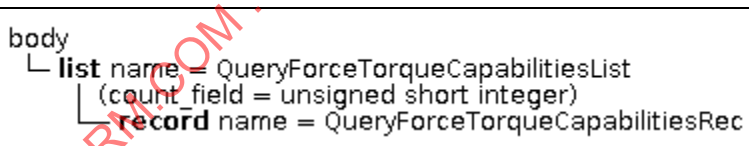
4.6.5 Encoding

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



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

body └ **list** name = QueryForceTorqueList (count_field = unsigned short integer) └ **record** name = QueryForceTorqueRec					
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

body └ **list** name = ReportForceTorqueCapabilitiesList (count_field = unsigned short integer) └ **record** name = ReportForceTorqueCapabilitiesRec					
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)

10	**<fixed_field>** MaxForceZ	unsigned short integer	newton	true	Most positive force measurable in the Z-direction. (scaled range = [-5000,5000], round)
11	**<fixed_field>** MinTorqueX	unsigned short integer	newton meter	true	Most negative torque measurable around the X-axis. (scaled range = [-5000,5000], round)
12	**<fixed_field>** MaxTorqueX	unsigned short integer	newton meter	true	Most positive torque measurable around the X-axis. (scaled range = [-5000,5000], round)
13	**<fixed_field>** MinTorqueY	unsigned short integer	newton meter	true	Most negative torque measurable around the Y-axis. (scaled range = [-5000,5000], round)
14	**<fixed_field>** MaxTorqueY	unsigned short integer	newton meter	true	Most positive torque measurable around the Y-axis. (scaled range = [-5000,5000], round)
15	**<fixed_field>** MinTorqueZ	unsigned short integer	newton meter	true	Most negative torque measurable around the Z-axis. (scaled range = [-5000,5000], round)
16	**<fixed_field>** MaxTorqueZ	unsigned short integer	newton meter	true	Most positive torque measurable around the Z-axis. (scaled range = [-5000,5000], round)

4.6.5.2.2 ID 480C: ReportForceTorque

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

Table 55 - ReportForceTorque message encoding

body
└ **list** name = ReportForceTorqueList
  (count_field = unsigned short integer)
  └ **record** name = ReportForceTorqueRec

Record Name = ReportForceTorqueRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned byte			
2	**<fixed_field>** SensorID	unsigned short integer	one	false	The ID of the sensor. A value of zero is not allowed
3	**<fixed_field>** CoordinateSystem	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= NativeCoordinateSystems 1= VehicleCoordinateSystems
4	**<fixed_field>** ForceX	unsigned short integer	newton	true	Current force measured in the X-direction (scaled range = [-5000,5000], round)
5	**<fixed_field>** ForceY	unsigned short integer	newton	true	Current force measured in the Y-direction (scaled range = [-5000,5000], round)
6	**<fixed_field>** ForceZ	unsigned short integer	newton	true	Current force measured in the Z-direction (scaled range = [-5000,5000], round)
7	**<fixed_field>** TorqueX	unsigned short integer	newton meter	true	Current torque measured around the X-axis (scaled range = [-5000,5000], round)
8	**<fixed_field>** TorqueY	unsigned short integer	newton meter	true	Current torque measured around the Y-axis (scaled range = [-5000,5000], round)
9	**<fixed_field>** TorqueZ	unsigned short integer	newton meter	true	Current torque measured around the Z-axis (scaled range = [-5000,5000], round)

4.6.6 Protocol Behavior

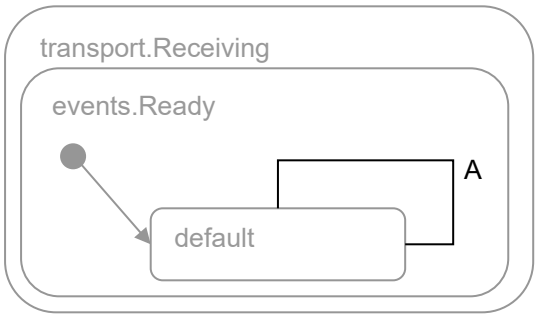


Figure 6 - ForceTorqueSensor service protocol behavior

Table 56 - ForceTorqueSensor service state transitions

Start State	Trigger	Conditions	Actions
A	QueryForceTorque		sendReportForceTorque (msg,transportData)
A	QueryForceTorqueCapabilities		sendReportForceTorqueCapabilities(msg, transportData)

Table 57 - ForceTorqueSensor service transition actions

Action	Interpretation
sendReportForceTorque	Send a ReportForceTorque message
sendReportForceTorqueCapabilities	Send a ReportForceTorqueCapabilities message

4.7 H264VideoEncoding

name=H264VideoEncoding
version=2.0
id=urn:jaus:jss:environmentSensing:H264VideoEncoding

4.7.1 Description

The H264 Video Encoding Service provides a mechanism for querying and configuring the H264 encoding of one or more video sensors.

4.7.2 Assumptions

Messages may be delayed, lost, or reordered.

4.7.3 References

Inherits-from=digitalVideo
version=2.0
id=urn:jaus:jss:environmentSensing:DigitalVideo

4.7.4 Vocabulary

Table 58 - H264VideoEncoding service message vocabulary

Message ID (hex)	Name	Command
Input Set		
080D	SetH264VideoEncodingConfiguration	true
280E	QueryH264VideoEncodingCapabilities	false
280D	QueryH264VideoEncodingConfiguration	false
Output Set		
0801	ConfirmSensorConfiguration	false
480E	ReportH264VideoEncodingCapabilities	false
480D	ReportH264VideoEncodingConfiguration	false

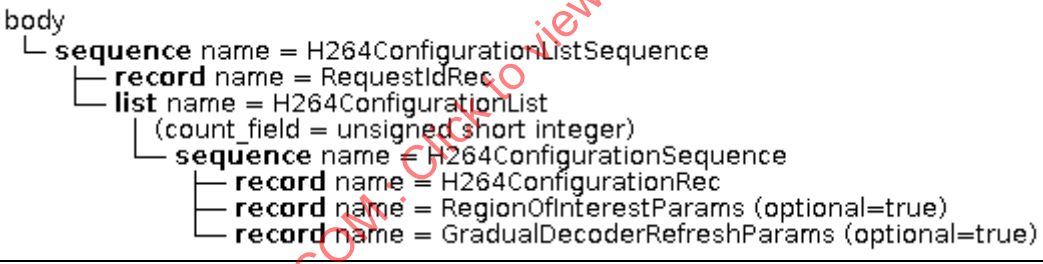
4.7.5 Encoding

4.7.5.1 Input Set

4.7.5.1.1 ID 080D: SetH264VideoEncodingConfiguration

This message is used to set the active configuration for one or more H264 encoded video streams.

Table 59 - SetH264VideoEncodingConfiguration message encoding



Record Name = RequestIdRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** RequestID	unsigned byte	one	false	

Record Name = H264ConfigurationRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned byte			
2	**<fixed_field>** SensorID	unsigned short integer	one	false	Note: Zero is not a valid VisualSensorID value

3	**<fixed_field>** Profile	unsigned byte	one	true	Enumeration for h264 profile setting. Value set, offset=false, ranges/enums: 0= Baseline 1= ConstrainedBaseline 2= Main 3= Extended 4= High 5= ProgressiveHigh 6= ConstrainedHigh 7= High10 8= High422 9= High444Predictive 10= High10Intra 11= High422Intra 12= High444Intra 13= CAVLC444 14= ScalableBaseline 15= ScalableConstrainedBaseline 16= ScalableHigh 17= ScalableConstrainedHigh 18= ScalableHighIntra 19= StereoHigh 20= MultiviewHigh
4	**<fixed_field>** Level	unsigned integer	one	true	Enumeration for target h264 level. Value set, offset=false, ranges/enums: 0= Level_1 1= Level_1B 2= Level_11 3= Level_12 4= Level_13 5= Level_2 6= Level_21 7= Level_22 8= Level_3 9= Level_31 10= Level_32 11= Level_4 12= Level_41 13= Level_42 14= Level_5 15= Level_51 16= Level_52 17= Level_6 18= Level_61 19= Level_62
5	**<fixed_field>** Preset	unsigned byte	one	true	Preconfigured settings for hardware specific H264 settings. The actual values for each preset are not specified by the standard and are up to the implementation. Value set, offset=false, ranges/enums: 0= LowLatency 1= SlowComms 2= PersistentStare 3= DriveVision 4= Manipulation 5= BestQuality 12= ProgramSpecific_1 13= ProgramSpecific_2 14= ProgramSpecific_3 15= ProgramSpecific_4
6	**<fixed_field>** GroupOfPictures	unsigned byte	one	true	Number of interim frames before sending a key frame
7	**<fixed_field>** GradualDecoderRefreshOnOff	unsigned byte	one	true	Turn on/off Gradual Decoder Refresh, also called Periodic Intra Refresh. Value set, offset=false, ranges/enums: 0= OFF 1= ON

8	**<fixed_field>** RegionOfInterestOnOff	unsigned byte	one	true	Turn on/off region of interest (foveation) encoding, if supported. The region of interest is specified by the RegionOfInterestParams array. Value set, offset=false, ranges/enums: 0= OFF 1= ON
9	**<fixed_field>** IntraMacroRefresh	unsigned short integer	one	true	This parameter is the random intra macroblock refresh; it provides the refresh rate in Macroblocks per picture. (scaled range = [0,100], round)

Record Name = RegionOfInterestParams

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** X_Start	unsigned short integer	percent	false	Horizontal start location of high bitrate window, measured as percent of total image width from the left (scaled range = [0,100], round)
2	**<fixed_field>** Y_Start	unsigned short integer	percent	false	Vertical start location of high bitrate window, measured as percent of total image height from the top (scaled range = [0,100], round)
3	**<fixed_field>** Width	unsigned short integer	percent	false	Width of high bitrate window, measured as percent of total image width (scaled range = [0,100], round)
4	**<fixed_field>** Height	unsigned short integer	percent	false	Height of high bitrate window , measured as percent of total image height (scaled range = [0,100], round)
5	**<fixed_field>** HighBitRate	unsigned short integer	one	false	Bitrate of ROI window, measured in kilobits per second
6	**<fixed_field>** LowBitRate	unsigned short integer	one	false	Bitrate of ROI window, measured in kilobits per second

Record Name = GradualDecoderRefreshParams

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** GDR_Duration	unsigned short integer	one	false	The number of frames needed to completely refresh the image
2	**<fixed_field>** GDR_Interval	unsigned short integer	one	false	The number of frames that need to be transmitted before starting a new GDR cycle. Note that the interval must be greater than or equal to the duration

4.7.5.1.2 ID 280E: QueryH264VideoEncodingCapabilities

This message must be used to query for the H264 encoding capabilities of one or more visual sensors on the receiving service.

Table 60 - QueryH264VideoEncodingCapabilities message encoding

body
└ **list** name = QueryH264VideoEncodingCapabilitiesList
  ├ (count_field = unsigned short integer)
  └ **record** name = QueryH264VideoEncodingRec

Record Name = QueryH264VideoEncodingRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** SensorID	unsigned short integer	one	false	The ID of the sensor for which the capabilities are being queried.Note: A value of 0 shall be interpreted as all associated sensors.

4.7.5.1.3 ID 280D: QueryH264VideoEncodingConfiguration

This message must be used to query for the currently configured H264 encoding parameters.

Table 61 - QueryH264VideoEncodingConfiguration message encoding

body
└ **list** name = QueryH264VideoEncodingConfigurationList
  │ (count_field = unsigned short integer)
  └ **record** name = QueryH264VideoEncodingRec					

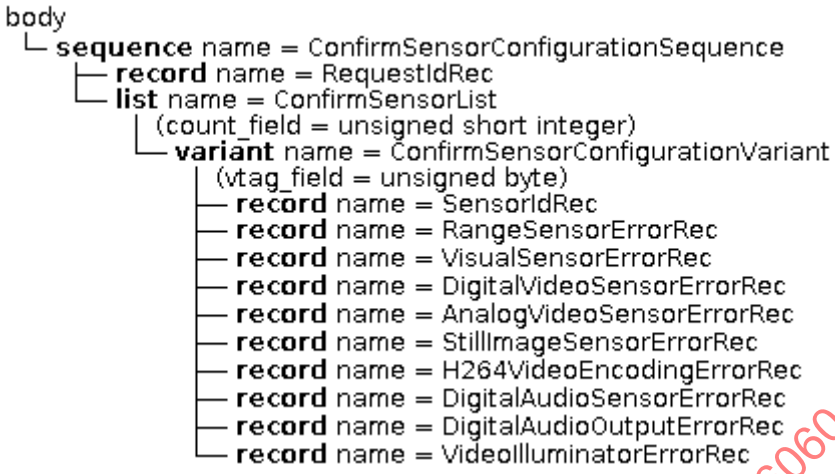
Record Name = QueryH264VideoEncodingRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SensorID	unsigned short integer	one	false	The ID of the sensor for which the capabilities are being queried.Note: A value of 0 shall be interpreted as all associated sensors.

4.7.5.2 Output Set

4.7.5.2.1 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 62 - ConfirmSensorConfiguration message encoding



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)

4.7.5.2.2 ID 480E: ReportH264VideoEncodingCapabilities

This message is used to report which configuration options are supported on one or more H264 encoded video streams.

Table 63 - ReportH264VideoEncodingCapabilities message encoding

body
└ **list** name = H264CapabilitiesList
  (count_field = unsigned short integer)
  └ **record** name = H264CapabilitiesRec

Record Name = H264CapabilitiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned byte			
2	**<fixed_field>** SensorID	unsigned short integer	one	false	Note: Zero is not a valid VisualSensorID value
3	**<bit_field>** SupportedProfiles	unsigned integer	bit_field	true	A high value (1) for a bit means the profile is supported. Bits 0..0, Baseline: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 1..1, ConstrainedBaseline: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 2..2, Main: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 3..3, Extended: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 4..4, High: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 5..5, ProgressiveHigh: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 6..6, ConstrainedHigh: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 7..7, High10: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 8..8, High422: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 9..9, High444Predictive: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 10..10, High10Intra: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 11..11, High422Intra: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 12..12, High444Intra: Value set, offset=false, ranges/enums: 0= NotSupported

					1= Supported Bits 13..13, CAVLC444: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 14..14, ScalableBaseline: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 15..15, ScalableConstrainedBaseline: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 16..16, ScalableHigh: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 17..17, ScalableConstrainedHigh: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 18..18, ScalableHighIntra: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 19..19, StereoHigh: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 20..20, MultiviewHigh: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported
4	**<bit_field>** SupportedLevels	unsigned integer	bit_field	true	A high value (1) for a bit means the level is supported. Bits 0..0, Level_1: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 1..1, Level_1B: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 2..2, Level_11: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 3..3, Level_12: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 4..4, Level_13: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 5..5, Level_2: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 6..6, Level_21: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 7..7, Level_22: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 8..8, Level_3: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 9..9, Level_31: Value set, offset=false, ranges/enums: 0= NotSupported

					1= Supported Bits 10..10, Level_32: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 11..11, Level_4: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 12..12, Level_41: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 13..13, Level_42: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 14..14, Level_5: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 15..15, Level_51: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 16..16, Level_52: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 17..17, Level_6: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 18..18, Level_61: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 19..19, Level_62: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported
5	**<bit_field>** SupportedPresets	unsigned short integer	bit_field	true	A high value (1) for a bit means the preset is supported. Bits 0..0, LowLatency: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 1..1, SlowComms: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 2..2, PersistentStare: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 3..3, DriveVision: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 4..4, Manipulation: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 5..5, BestQuality: Value set, offset=false, ranges/enums: 0= NotSupported 1= Supported Bits 6..11, Reserved: Value set, offset=false, ranges/enums: [0,63] Bits 12..15, ProgramSpecific: Value set, offset=false, ranges/enums: [0,15]

6	**<fixed_field>** GroupOfPicturesSupported	unsigned byte	one	true	Boolean for Group of Pictures support Value set, offset=false, ranges/enums: 0= FALSE 1= TRUE
7	**<fixed_field>** GradualDecoderRefreshSupported	unsigned byte	one	true	Boolean for Gradual Decoder Refresh (Periodic Intra Refresh) support Value set, offset=false, ranges/enums: 0= FALSE 1= TRUE
8	**<fixed_field>** RegionOfInterestSupported	unsigned byte	one	true	Boolean for Region of Interest (foveation) support Value set, offset=false, ranges/enums: 0= FALSE 1= TRUE

4.7.5.2.3 ID 480D: ReportH264VideoEncodingConfiguration

This message is used to report the active configuration for one or more H264 encoded video streams.

Table 64 - ReportH264VideoEncodingConfiguration message encoding

body
└ **list** name = H264ConfigurationList
  │ (count_field = unsigned short integer)
  └ **sequence** name = H264ConfigurationSequence
    ├ **record** name = H264ConfigurationRec
    ├ **record** name = RegionOfInterestParams (optional=true)
    └ **record** name = GradualDecoderRefreshParams (optional=true)

Record Name = H264ConfigurationRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned byte			
2	**<fixed_field>** SensorID	unsigned short integer	one	false	Note: Zero is not a valid VisualSensorID value
3	**<fixed_field>** Profile	unsigned byte	one	true	Enumeration for h264 profile setting. Value set, offset=false, ranges/enums: 0= Baseline 1= ConstrainedBaseline 2= Main 3= Extended 4= High 5= ProgressiveHigh 6= ConstrainedHigh 7= High10 8= High422 9= High444Predictive 10= High10Intra 11= High422Intra 12= High444Intra 13= CAVLC444 14= ScalableBaseline 15= ScalableConstrainedBaseline 16= ScalableHigh 17= ScalableConstrainedHigh 18= ScalableHighIntra 19= StereoHigh 20= MultiviewHigh

4	**<fixed_field>** Level	unsigned integer	one	true	Enumeration for target h264 level. Value set, offset=false, ranges/enums: 0= Level_1 1= Level_1B 2= Level_11 3= Level_12 4= Level_13 5= Level_2 6= Level_21 7= Level_22 8= Level_3 9= Level_31 10= Level_32 11= Level_4 12= Level_41 13= Level_42 14= Level_5 15= Level_51 16= Level_52 17= Level_6 18= Level_61 19= Level_62
5	**<fixed_field>** Preset	unsigned byte	one	true	Preconfigured settings for hardware specific H264 settings. The actual values for each preset are not specified by the standard and are up to the implementation. Value set, offset=false, ranges/enums: 0= LowLatency 1= SlowComms 2= PersistentStare 3= DriveVision 4= Manipulation 5= BestQuality 12= ProgramSpecific_1 13= ProgramSpecific_2 14= ProgramSpecific_3 15= ProgramSpecific_4
6	**<fixed_field>** GroupOfPictures	unsigned byte	one	true	Number of interim frames before sending a key frame
7	**<fixed_field>** GradualDecoderRefreshOnOff	unsigned byte	one	true	Turn on/off Gradual Decoder Refresh, also called Periodic Intra Refresh. Value set, offset=false, ranges/enums: 0= OFF 1= ON
8	**<fixed_field>** RegionOfInterestOnOff	unsigned byte	one	true	Turn on/off region of interest (foveation) encoding, if supported. The region of interest is specified by the RegionOfInterestParams array. Value set, offset=false, ranges/enums: 0= OFF 1= ON
9	**<fixed_field>** IntraMacroRefresh	unsigned short integer	one	true	This parameter is the random intra macroblock refresh; it provides the refresh rate in Macroblocks per picture. (scaled range = [0,100], round)

Record Name = RegionOfInterestParams

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** X_Start	unsigned short integer	percent	false	Horizontal start location of high bitrate window, measured as percent of total image width from the left (scaled range = [0,100], round)
2	**<fixed_field>** Y_Start	unsigned short integer	percent	false	Vertical start location of high bitrate window, measured as percent of total image height from the top (scaled range = [0,100], round)
3	**<fixed_field>** Width	unsigned short integer	percent	false	Width of high bitrate window, measured as percent of total image width (scaled range = [0,100], round)

4	**<fixed_field>** Height	unsigned short integer	percent	false	Height of high bitrate window , measured as percent of total image height (scaled range = [0,100], round)
5	**<fixed_field>** HighBitRate	unsigned short integer	one	false	Bitrate of ROI window, measured in kilobits per second
6	**<fixed_field>** LowBitRate	unsigned short integer	one	false	Bitrate of ROI window, measured in kilobits per second
Record Name = GradualDecoderRefreshParams					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** GDR_Duration	unsigned short integer	one	false	The number of frames needed to completely refresh the image
2	**<fixed_field>** GDR_Interval	unsigned short integer	one	false	The number of frames that need to be transmitted before starting a new GDR cycle. Note that the interval must be greater than or equal to the duration

4.7.6 Protocol Behavior

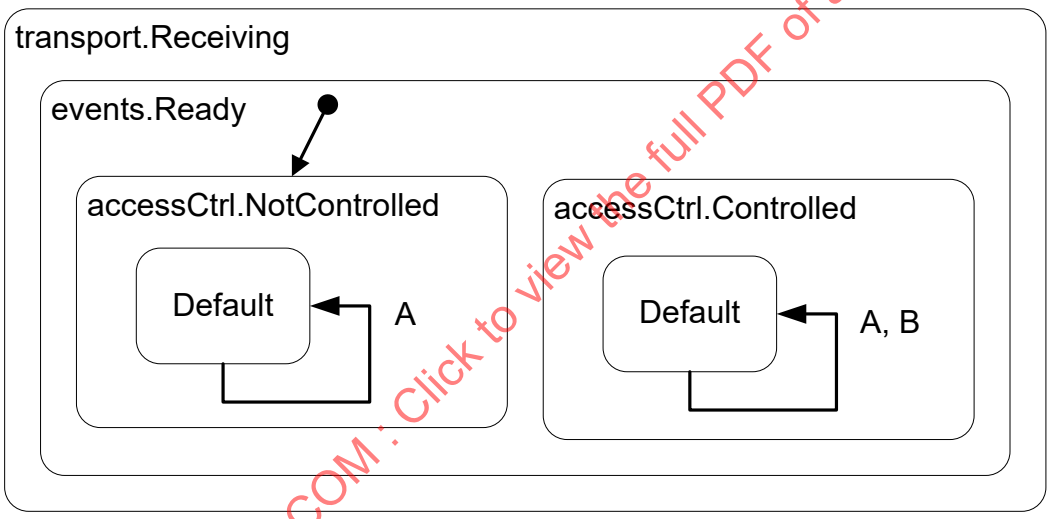


Figure 7 - H264VideoEncoding service protocol behavior

Table 65 - H264VideoEncoding service state transitions

Start State	Trigger	Conditions	Actions
A	QueryH264VideoEncodingCapabilities		sendReportH264VideoEncodingCapabilities (msg,transportData)
A	QueryH264VideoEncodingConfiguration		sendReportH264VideoEncodingConfiguration (msg,transportData)
B	SetH264VideoEncodingConfiguration	isControllingClient(transportData)	setH264VideoEncodingConfig (msg) , sendConfirmSensorConfiguration (msg, transportData)

Table 66 - H264VideoEncoding service conditions

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

Table 67 - H264VideoEncoding service transition actions

Action	Interpretation
sendReportH264VideoEncodingCapabilities	Send a ReportH264VideoEncodingCapabilities message to the original requestor
sendReportH264VideoEncodingConfiguration	Send a ReportH264VideoEncodingConfiguration message to the original requestor
setH264VideoEncodingConfig	Update the H264 encoding configuration if valid for the specified sensor ID
sendConfirmSensorConfiguration	Send sendConfirmSensorConfiguration message with confirmation or error code for each specified sensor ID

4.8 PortMapper

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

4.8.1 Description

A port mapping service is required to support streaming of data from a device residing on a different subnet. For example, a digital resource such as a video or audio stream described by an RTSP endpoint may be on a network internal to the host platform. However, a controller or other client may reside on a different external network. The Port Mapper Service allows for the bridging of these two networks, such that the stream is accessible on the external facing network. Effectively, the stream source uses to the Port Mapper to request port forwarding, such that any traffic received on the external facing IP address and port is redirected by the service implementation to the stream host on the internal IP address and port. The stream source can then use the external IP and port in its discovery registration process. Note that the implementation of the Port Mapper must have access to both networks.

4.8.2 Assumptions

Messages may be delayed, lost, or reordered.

4.8.3 References

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

4.8.4 Vocabulary

Table 68 - PortMapper service message vocabulary

Message ID (hex)	Name	Command
Input Set		
080B	CancelPortMapping	false
080C	RequestPortMapping	false
2815	QueryPortMappings	false
Output Set		
080E	GrantPortMapping	false
4815	ReportPortMappings	false

4.8.5 Encoding

4.8.5.1 Input Set

4.8.5.1.1 ID 080B: CancelPortMapping

This message is sent to cancel any mapping to the endpoint specified in the message.

Table 69 - CancelPortMapping message encoding

body └ **record** name = RequestPortMappingRec					
Record Name = RequestPortMappingRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SocketType	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= UDP 1= TCP
2	**<fixed_field>** IPAddr	unsigned integer	one	false	IPv4 address of endpoint port to which PortMapper will map, in network byte order
3	**<fixed_field>** Port	unsigned short integer	one	false	

4.8.5.1.2 ID 080C: RequestPortMapping

This message is sent to request a port mapping at the network gateway (node hosting the PortMapper) such that network packets arriving on the gateway's mapped port are forwarded to the endpoint specified in this request.

Table 70 - RequestPortMapping message encoding

body └ **record** name = RequestPortMappingRec					
Record Name = RequestPortMappingRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SocketType	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= UDP 1= TCP
2	**<fixed_field>** IPAddr	unsigned integer	one	false	IPv4 address of endpoint port to which PortMapper will map, in network byte order
3	**<fixed_field>** Port	unsigned short integer	one	false	

4.8.5.1.3 ID 2815: QueryPortMappings

This message is sent to request a list of all current mappings

Table 71 - QueryPortMappings message encoding

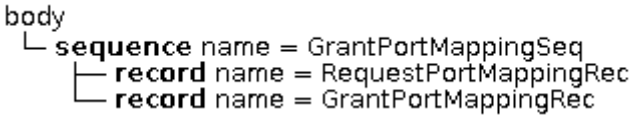
body └ (empty)
Empty message body

4.8.5.2 Output Set

4.8.5.2.1 ID 080E: GrantPortMapping

The GrantPortMapping message is sent in response to the RequestPortMapping message. GrantPortMapping contains both the requested endpoint that was specified in the request, and the mapped endpoint on the external-facing network.

Table 72 - GrantPortMapping message encoding

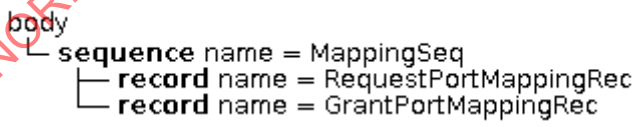


Record Name = RequestPortMappingRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SocketType	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= UDP 1= TCP
2	**<fixed_field>** IPAddr	unsigned integer	one	false	IPv4 address of endpoint port to which PortMapper will map, in network byte order
3	**<fixed_field>** Port	unsigned short integer	one	false	
Record Name = GrantPortMappingRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** IPAddr	unsigned integer	one	false	IPv4 address of client side mapping of endpoint, in network byte order
2	**<fixed_field>** Port	unsigned short integer	one	false	

4.8.5.2.2 ID 4815: ReportPortMappings

This message is includes a list of all current mappings.

Table 73 - ReportPortMappings message encoding



Record Name = RequestPortMappingRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SocketType	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= UDP 1= TCP
2	**<fixed_field>** IPAddr	unsigned integer	one	false	IPv4 address of endpoint port to which PortMapper will map, in network byte order
3	**<fixed_field>** Port	unsigned short integer	one	false	
Record Name = GrantPortMappingRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** IPAddr	unsigned integer	one	false	IPv4 address of client side mapping of endpoint, in network byte order
2	**<fixed_field>** Port	unsigned short integer	one	false	

4.8.6 Protocol Behavior

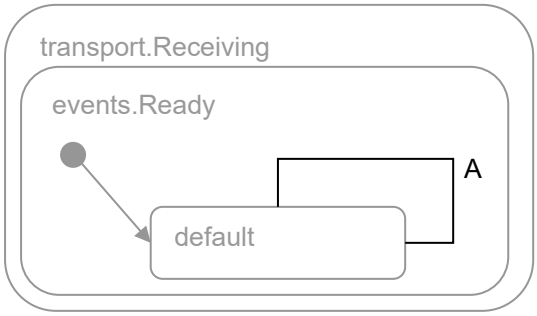


Figure 8 - PortMapper service protocol behavior

Table 74 - PortMapper service state transitions

Start State	Trigger	Conditions	Actions
A	RequestPortMapping		createMappedPort (msg,transportData) , sendGrantPortMapping (msg,transportData)
A	CancelPortMapping		cancelMappedPort (msg)
A	QueryPortMappings		sendReportPortMappings (msg,transportData)

Table 75 - PortMapper service transition actions

Action	Interpretation
createMappedPort	Assign an external address and port to the endpoint specified in the message and begin forwarding traffic received on that port
sendGrantPortMapping	Send a Grant Port Mapping message to the requesting client
cancelMappedPort	Remove the specified mapping from the list of active maps
sendReportPortMappings	Send a Report Port Mappings message to the requesting client

4.9 RangeSensor

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

4.9.1 Description

The function of the Range Sensor Service is to provide information from proximity sensors. This service will output the location of various Data Points with a certain measure of accuracy. A given Range Sensor service may be comprised of one to many actual physical sensors or technologies. Each sub-sensor can be assigned (by the developer) a unique Sensor ID. When appropriate, the reserved Sensor ID of zero may be used to refer to all sensors attached to a given Range Sensor Service.

The Data Points are measured in the sensor's native coordinate system and are expressed in terms of range, bearing and inclination. Range is the distance, in meters, along the line from the origin of the sensor's native coordinate system (sensor's origin) to the specified point. Bearing is the angle, in radians, that the line from the sensor's origin to the specified point makes about the sensor's z-axis in the right handed sense (Figure 9). Inclination is the the angle, in radians, that the line from the sensor origin to the specified point makes about the sensor's y-axis in the right handed sense (Figure 9).

Each data point has an optional ID parameter. This parameter is provided for those sensor technologies which may assign and/or track entities based on unique ID values; however, such tracking capabilities are not required for a compliant Range Sensor Service. The behavior of the data point ID is not specified, i.e., IDs may repeat in a given report and IDs may persist from one report to another. No semantic value should be placed on the ID values in a generalized way. Data Point ID behavior should be derived from the underlying sensor or algorithm technology and is mearly provided to be used in those situations where mutiple parties can agree upon the behavior and semantics of the ID values.

Data from the range sensor can be reported in both a compressed and uncompressed format, different query and report messages are provided for each exchange and the kind of data compression supported by the service is reported in the Report Range Sensor Capabilities message. Requests for unsupported data compression algorithms will result in the generation of a Report Sensor Error message indicating an unsupported compression request.

The range sensor can express the bearing, inclination and range terms with respect to either its native coordinate system or the vehicle coordinate system if coordinate transforms are supported. The Query Sensor Geometric Properties message can be used to determine the geometric relationship between the sensor and the vehicle coordinate system. Three possible coordinate responses are possible: (a) the service does not know the sensor's position, (b) the sensor coordinate system is fixed with respect to the vehicle, and (c) the sensor is attached to some manipulator. These cases are supported by the Report Sensor Geometric Properties message and are described therein.

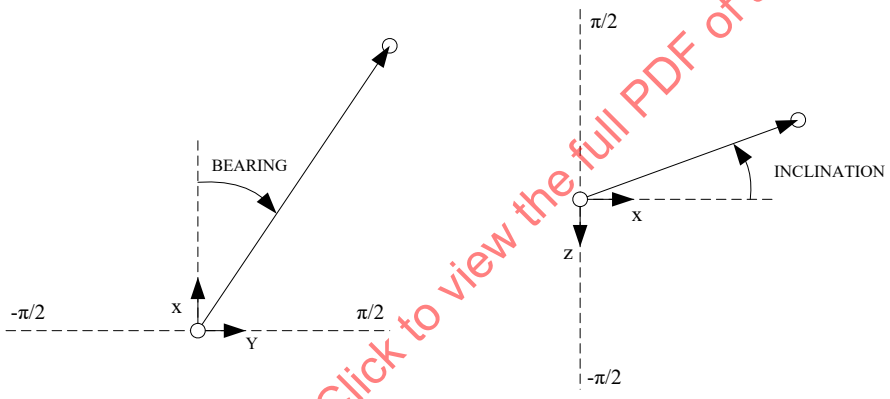


Figure 9 - Bearing and inclination definition

4.9.2 Assumptions

Messages may be delayed, lost, or reordered.

4.9.3 References

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

4.9.4 Vocabulary

Table 76 - RangeSensor service message vocabulary

Message ID (hex)	Name	Command
Input Set		
0802	SetRangeSensorConfiguration	true
2805	QuerySensorGeometricProperties	false
2804	QueryRangeSensorCompressedData	false
2803	QueryRangeSensorData	false
2801	QueryRangeSensorCapabilities	false
2802	QueryRangeSensorConfiguration	false
Output Set		
0801	ConfirmSensorConfiguration	false
4801	ReportRangeSensorCapabilities	false
4802	ReportRangeSensorConfiguration	false
4803	ReportRangeSensorData	false
4804	ReportRangeSensorCompressedData	false
4805	ReportSensorGeometricProperties	false

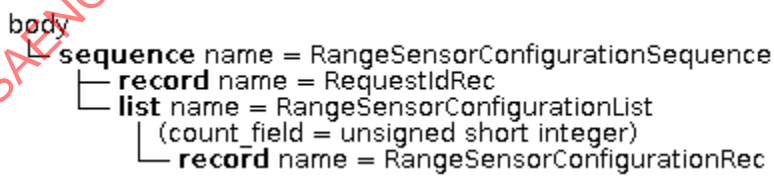
4.9.5 Encoding

4.9.5.1 Input Set

4.9.5.1.1 ID 0802: SetRangeSensorConfiguration

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

Table 77 - SetRangeSensorConfiguration message encoding



Record Name = RequestIdRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** RequestID	unsigned byte	one	false	

Record Name = RangeSensorConfigurationRec

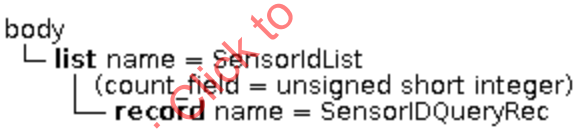
Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned short integer			
2	**<fixed_field>** SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]

3	**<fixed_field>** HorizontalFieldOfViewStartAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
4	**<fixed_field>** HorizontalFieldOfViewStopAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
5	**<fixed_field>** VerticalFieldOfViewStartAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
6	**<fixed_field>** VerticalFieldOfViewStopAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
7	**<fixed_field>** UpdateRate	unsigned short integer	hertz	true	(scaled range = [0,1000], round)
8	**<fixed_field>** MinimumRange	unsigned integer	meter	true	(scaled range = [0,1000000.0], round)
9	**<fixed_field>** MaximumRange	unsigned integer	meter	true	(scaled range = [0,1000000.0], round)
10	**<fixed_field>** SensorState	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Active 1= Standby 2= Off

4.9.5.1.2 ID 2805: QuerySensorGeometricProperties

This message shall cause the receiving component to reply to the requestor with a Report Sensor Geometric Properties message.

Table 78 - QuerySensorGeometricProperties message encoding



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

4.9.5.1.3 ID 2804: QueryRangeSensorCompressedData

This message shall cause the receiving service to reply to the requestor with a Report Range Sensor Compressed Data 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. The third field specifies which coordinate system the data should be reported in, either the sensor's native coordinate system or, if supported, a coordinate system specified by a Set Specified Sensor Coordinate System message. The fourth field specifies the data compression algorithm. Three compression algorithms are supported: DEFLATE, bzip2, and LZMA.

Table 79 - QueryRangeSensorCompressedData message encoding

body └ **list** name = QueryRangeSensorCompressedDataList (count_field = unsigned short integer) └ **record** name = QueryRangeSensorCompressedDataRec					
Record Name = QueryRangeSensorCompressedDataRec					
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>** ReportCoordinateSystem	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Native Coordinate System 1= Vehicle Coordinate System
3	**<fixed_field>** DataCompression	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= None 1= DEFLATE 2= bzip2 3= LZMA
4	**<fixed_field>** QueryPresenceVector	unsigned short integer	one	false	

4.9.5.1.4 ID 2803: QueryRangeSensorData

This message shall cause the receiving service to reply to the requestor with a Report Range Sensor Data 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. The second field specifies which coordinate system the data should be reported in, either the sensor's native coordinate system or, if supported, a coordinate system specified by a Set Specified Sensor Coordinate System message.

Table 80 - QueryRangeSensorData message encoding

body └ **list** name = QueryRangeSensorDataList (count_field = unsigned short integer) └ **record** name = QueryRangeSensorDataRec					
Record Name = QueryRangeSensorDataRec					
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>** ReportCoordinateSystem	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Native Coordinate System 1= Vehicle Coordinate System
3	**<fixed_field>** QueryPresenceVector	unsigned short integer	one	false	

4.9.5.1.5 ID 2801: QueryRangeSensorCapabilities

This message shall cause the receiving service to reply to the requestor with a Report Range Sensor Capabilities 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 81 - QueryRangeSensorCapabilities message encoding

body └ **list** name = RangeSensorCapabilitiesList (count_field = unsigned short integer) └ **record** name = QueryRangeSensorCapabilitiesRec					
Record Name = QueryRangeSensorCapabilitiesRec					
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 short integer	one	false	

4.9.5.1.6 ID 2802: QueryRangeSensorConfiguration

This message shall cause the receiving service to reply to the requestor with a Report Range 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 82 - QueryRangeSensorConfiguration message encoding

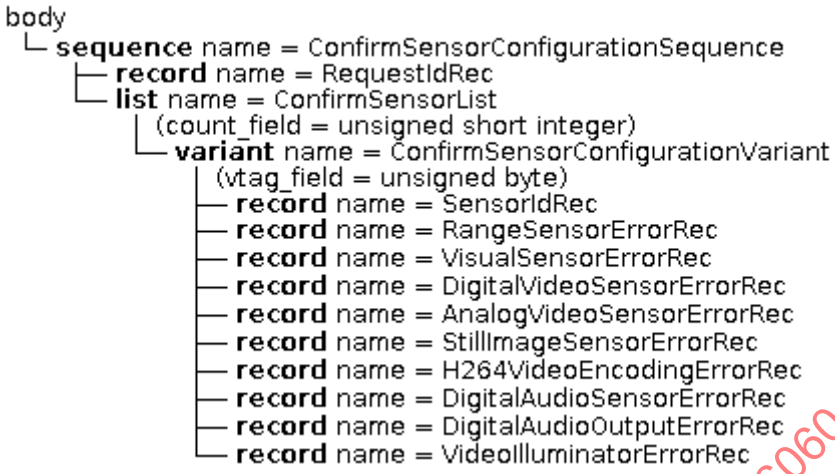
body └ **list** name = RangeSensorConfigurationList (count_field = unsigned short integer) └ **record** name = QueryRangeSensorConfigurationRec					
Record Name = QueryRangeSensorConfigurationRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SensorID	unsigned 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 short integer	one	false	

4.9.5.2 Output Set

4.9.5.2.1 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 83 - ConfirmSensorConfiguration message encoding



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)

4.9.5.2.2 ID 4801: ReportRangeSensorCapabilities

This message is used to report the range sensors' capabilities upon receipt of a Query Range Sensor Capabilities message. Capabilities include both static sensor properties and valid values and ranges for properties which can be modified by the Set Range Sensor Capabilities Message.

Table 84 - ReportRangeSensorCapabilities message encoding

body
└ **list** name = RangeSensorCapabilitiesList
  (count_field = unsigned short integer)
  └ **record** name = RangeSensorCapabilitiesRec

Record Name = RangeSensorCapabilitiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned short integer			
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	**<variable_length_string>** SensorName	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
4	**<bit_field>** SupportedStates	unsigned byte	bit_field	true	Bits 0..0, Active: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Standby: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Off: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
5	**<fixed_field>** MinimumHorizontalFieldOfViewStartAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
6	**<fixed_field>** MaximumHorizontalFieldOfViewStopAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
7	**<fixed_field>** MinimumVerticalFieldOfViewStartAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
8	**<fixed_field>** MaximumVerticalFieldOfViewStopAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
9	**<fixed_field>** MiniumumUpdateRate	unsigned short integer	hertz	true	(scaled range = [0,1000], round)
10	**<fixed_field>** MaximumUpdateRate	unsigned short integer	hertz	true	(scaled range = [0,1000], round)
11	**<fixed_field>** MinimumRange	unsigned integer	meter	true	(scaled range = [0,1000000.0], round)
12	**<fixed_field>** MaximumRange	unsigned integer	meter	true	(scaled range = [0,1000000.0], round)

13	**<bit_field>** SupportedCompression	unsigned byte	bit_field	true	Bits 0..0, NoCompression: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, DEFLATE: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, bzip2: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, LZMA: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
14	**<fixed_field>** CoordinateTransformationSupported	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= False 1= True

4.9.5.2.3 ID 4802: ReportRangeSensorConfiguration

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

Table 85 - ReportRangeSensorConfiguration message encoding

body
└ **list** name = RangeSensorConfigurationList
  │ (count_field = unsigned short integer)
  └ **record** name = RangeSensorConfigurationRec

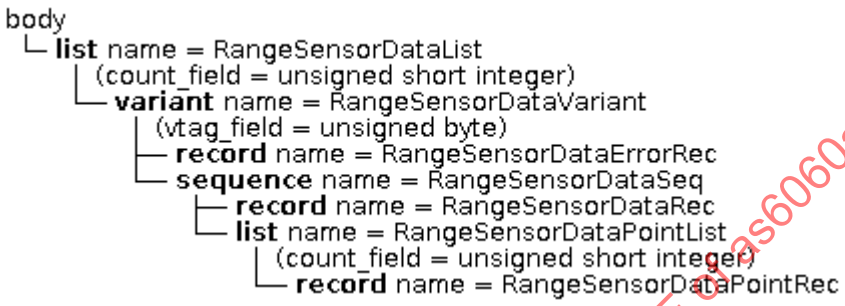
Record Name = RangeSensorConfigurationRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned short integer			
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>** HorizontalFieldOfViewStartAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
4	**<fixed_field>** HorizontalFieldOfViewStopAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
5	**<fixed_field>** VerticalFieldOfViewStartAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
6	**<fixed_field>** VerticalFieldOfViewStopAngle	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
7	**<fixed_field>** UpdateRate	unsigned short integer	hertz	true	(scaled range = [0,1000], round)
8	**<fixed_field>** MinimumRange	unsigned integer	meter	true	(scaled range = [0,1000000.0], round)
9	**<fixed_field>** MaximumRange	unsigned integer	meter	true	(scaled range = [0,1000000.0], round)
10	**<fixed_field>** SensorState	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Active 1= Standby 2= Off

4.9.5.2.4 ID 4803: ReportRangeSensorData

This message is sent by a receiving component upon receipt of a Query Range Sensor Data message. This message reports a list of detected data points for a given time. The RangeSensorDataSeq is used to report the data from a given sensor along with meta data such as sensor ID and timestamp. The timestamp defines the time at which the collected data was valid. Data points are defined by range, bearing and inclination with respect to either the native coordinate system or the vehicle coordinate system. Data is only reported for sensors which are in the active state. If data is queried for a sensor which is not active, the RangeSensorDataErrorRec is returned for that sensor ID report.

Table 86 - ReportRangeSensorData message encoding



Record Name = RangeSensorDataErrorRec

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>** DataErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Sensor is not Active 1= Invalid compression format 255= Unknown Error / Failure
3	**<variable_length_string>** ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = RangeSensorDataRec

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>** ReportCoordinateSystem	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Native Coordinate System 1= Vehicle Coordinate System
3	**<bit_field>** TimeStamp	unsigned integer	bit_field	false	Bits 0..9, Milliseconds: Value set, offset=false, ranges/enums: [0,999] Bits 10..15, Seconds: Value set, offset=false, ranges/enums: [0,59] Bits 16..21, Minutes: Value set, offset=false, ranges/enums: [0,59] Bits 22..26, Hour: Value set, offset=false, ranges/enums: [0,23] Bits 27..31, Day: Value set, offset=false, ranges/enums: [1,31]

Record Name = RangeSensorDataPointRec

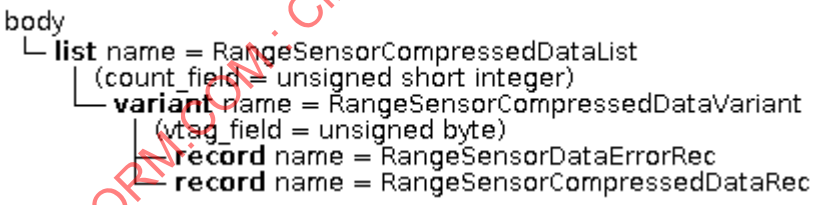
Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned short integer			
2	**<fixed_field>** PointID	unsigned integer	one	true	

3	**<fixed_field>** Range	unsigned integer	meter	false	(scaled range = [0,1000000.0], round)
4	**<fixed_field>** RangeValidity	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= False 1= True
5	**<fixed_field>** RangeErrorRMS	unsigned integer	meter	true	(scaled range = [0.0,100000.0], round)
6	**<fixed_field>** Bearing	unsigned integer	radian	false	(scaled range = [-3.141592653589793,3.141592653589793], round)
7	**<fixed_field>** BearingValidity	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= False 1= True
8	**<fixed_field>** BearingErrorRMS	unsigned integer	radian	true	(scaled range = [0,3.141592653589793], round)
9	**<fixed_field>** Inclination	unsigned integer	radian	false	(scaled range = [-3.141592653589793,3.141592653589793], round)
10	**<fixed_field>** InclinationValidity	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= False 1= True
11	**<fixed_field>** InclinationErrorRMS	unsigned integer	radian	true	(scaled range = [0.0,3.141592653589793], round)

4.9.5.2.5 ID 4804: ReportRangeSensorCompressedData

This message is used to report a list of detected data points for a given time. This message differs from the Report Range Sensor Data message in that the data contained herein has undergone compression by the specified algorithm. The timestamp defines the time at which the collected data was valid. Data points are defined by range, bearing and inclination with respect to either the sensor coordinate system or a specified coordinate system.

Table 87 - ReportRangeSensorCompressedData message encoding



Record Name = RangeSensorDataErrorRec

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>** DataErrorCode	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Sensor is not Active 1= Invalid compression format 255= Unknown Error / Failure
3	**<variable_length_string>** ErrorMessage	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

Record Name = RangeSensorCompressedDataRec

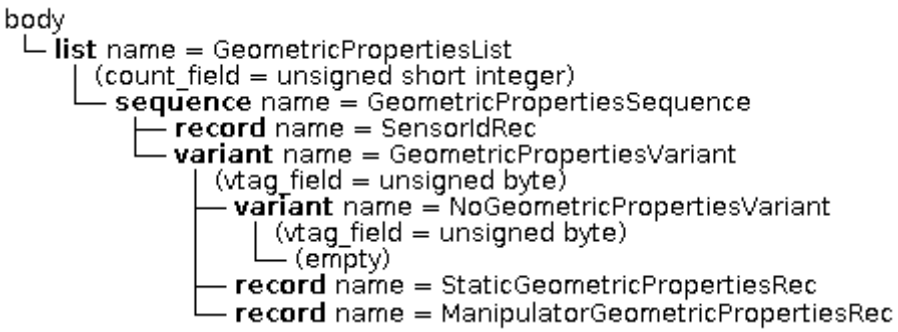
Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** SensorID	unsigned short integer	one	false	ID of the sensor. Zero is not a valid ID in this message. Value set, offset=false, ranges/enums:[1,65535]

2	**<fixed_field>** ReportCoordinateSystem	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Native Coordinate System 1= Vehicle Coordinate System
3	**<bit_field>** TimeStamp	unsigned integer	bit_field	false	Bits 0..9, Milliseconds: Value set, offset=false, ranges/enums: [0,999] Bits 10..15, Seconds: Value set, offset=false, ranges/enums: [0,59] Bits 16..21, Minutes: Value set, offset=false, ranges/enums: [0,59] Bits 22..26, Hour: Value set, offset=false, ranges/enums: [0,23] Bits 27..31, Day: Value set, offset=false, ranges/enums: [1,31]
4	**<fixed_field>** DataCompression	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= None 1= DEFLATE 2= bzip2 3= LZMA
5	**<variable_length_field>** CompressedData	RangeSensorDataPointList	N/A	false	Count field type = unsigned integer (min/max count = 0..2147483647)

4.9.5.2.6 ID 4805: ReportSensorGeometricProperties

This message is used to report information about the geometric properties of a sensor with respect to the vehicle. Three possible use cases are supported: (1) The sensor has no knowledge of its geometric properties. (2) The sensor is rigidly mounted to the vehicle and the position and orientation with respect to the vehicle coordinate system is known. In this case the position and orientation of the sensor is defined by a position vector (x, y, z) and unit quaternion (d; a, b, c) which specifies the axis and angle of rotation used to establish the orientation of the sensor coordinate system with respect to the vehicle coordinate system. (3) The sensor is mounted to a manipulator in which the JAUS ID, joint number and the sensor offset are described. In this case the position and orientation of the sensor is defined by a position vector (x, y, z) and unit quaternion (d; a, b, c) which specifies the axis and angle of rotation used to establish the orientation of the sensor coordinate system with respect to the link coordinate system.

Table 88 - ReportSensorGeometricProperties message encoding



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 = StaticGeometricPropertiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<array>** SensorPosition	Array of unsigned integer	N/A	false	dimension size = 3 <fixed_field>
2	**<array>** UnitQuaternion	Array of unsigned integer	N/A	false	dimension size = 4 <fixed_field>

Record Name = ManipulatorGeometricPropertiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** SubsystemID	unsigned short integer	one	false	
2	**<fixed_field>** NodeID	unsigned byte	one	false	
3	**<fixed_field>** ComponentID	unsigned byte	one	false	
4	**<fixed_field>** JointNumber	unsigned byte	one	false	
5	**<array>** SensorPosition	Array of unsigned integer	N/A	false	dimension size = 3 <fixed_field>
6	**<array>** UnitQuaternion	Array of unsigned integer	N/A	false	dimension size = 4 <fixed_field>

4.9.6 Protocol Behavior

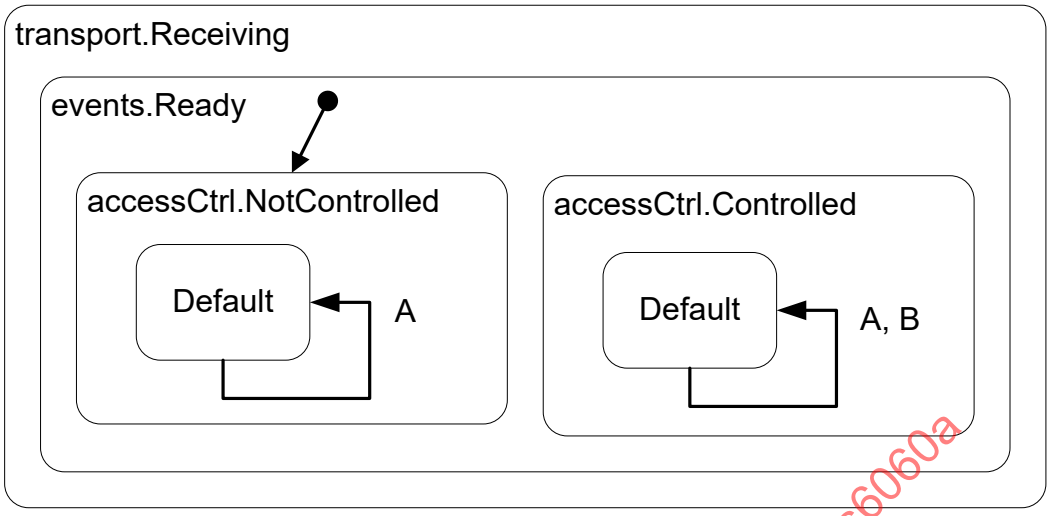


Figure 10 - RangeSensor service protocol behavior

Table 89 - RangeSensor service state transitions

Start State	Trigger	Conditions	Actions
A	QueryRangeSensorCapabilities		sendReportRangeSensorCapabilities (msg, transportData)
A	QueryRangeSensorConfiguration		sendReportRangeSensorConfiguration (msg,transportData)
A	QuerySensorGeometricProperties		sendReportSensorGeometricProperties (msg,transportData)
A	QueryRangeSensorData	isCoordinateTranformSupported (msg)	sendReportRangeSensorDataInRequestedCoordinateSystem (msg,transportData)
A	QueryRangeSensorData	! isCoordinateTranform Supported(msg)	sendReportRangeSensorDataInNativeCoordinateSystem (msg,transportData)
A	QueryRangeSensorCompressedData	isCoordinateTranform Supported(msg)	sendReportRangeSensorCompressedDataInRequestedCoordinateSystem (msg, transportData)
A	QueryRangeSensorCompressedData	! isCoordinateTranform Supported(msg)	sendReportRangeSensorCompressedDataInNativeCoordinateSystem (msg, transportData)
B	SetRangeSensorConfiguration	isControllingClient(transportData)	sendConfirmSensorConfiguration (msg, transportData) , updateRangeSensorConfiguration (msg)

Table 90 - RangeSensor service conditions

Condition	Interpretation
isCoordinateTranformSupported(msg)	True if the service supports coordinate system transformations for data reports.
! isCoordinateTranformSupported(msg)	True if the service does not support coordinate system transformations for data reports.
isControllingClient(transportData)	True if the message that triggered the transition is received from the client that is in control of this service.

Table 91 - RangeSensor service transition actions

Action	Interpretation
sendReportRangeSensorCapabilities	Send a Report Range Sensor Capabilities message
sendReportRangeSensorConfiguration	Send a ReportRangeSensorConfiguration message
sendReportSensorGeometricProperties	Send a ReportSensorGeometricProperties message
sendReportRangeSensorDataInRequestedCoordinateSystem	Send a ReportRangeSensorData message using the requested coordinate system
sendReportRangeSensorDataInNativeCoordinateSystem	Send a ReportRangeSensorData message in native coordinate system
sendReportRangeSensorCompressedDataInRequestedCoordinateSystem	Send a ReportRangeSensorCompressedData message using the requested coordinate system
sendReportRangeSensorCompressedDataInNativeCoordinateSystem	Send a ReportRangeSensorCompressedData message in native coordinate system
sendConfirmSensorConfiguration	Send ConfirmSensorConfiguration message
updateRangeSensorConfiguration	Update the sensor user controllable configuration parameters according to the ones specified.

4.10 StillImage

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

4.10.1 Description

This service provides access to the capabilities and configuration of a camera, allowing the controlling component to set the camera to a particular operational profile and to obtain images from the camera. While this service reports each image individually, the Events service can be used to automatically report images at a specified rate thereby simulating video (such as is typically done to create an MJPEG video stream).

4.10.2 Assumptions

Messages may be delayed, lost, or reordered.

4.10.3 References

Inherits-from=visualSensor
version=2.0
id=urn:jaus:jss:environmentSensing:VisualSensor

4.10.4 Vocabulary

Table 92 - StillImage service message vocabulary

Message ID (hex)	Name	Command
Input Set		
0807	SetStillImageSensorConfiguration	true
281C	QueryStillImageDataExt	false
2813	QueryStillImageSensorConfiguration	false
2812	QueryStillImageSensorCapabilities	false
Output Set		
481C	ReportStillImageDataExt	false
4813	ReportStillImageSensorConfiguration	false
4812	ReportStillImageSensorCapabilities	false

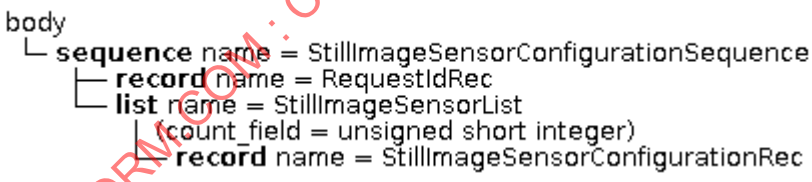
4.10.5 Encoding

4.10.5.1 Input Set

4.10.5.1.1 ID 0807: SetStillImageSensorConfiguration

This message is used to set the configuration of the still image sensor associated with the service. Configuration is based off of each sensor's capabilities as described in the Report Still Image 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 Still Image Error Record for the given Sensor ID however other, valid, configurations specified shall be set (if they exist).

Table 93 - SetStillImageSensorConfiguration message encoding



Record Name = RequestIdRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** RequestID	unsigned byte	one	false	

Record Name = StillImageSensorConfigurationRec

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>** 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)
4	**<fixed_field>** StillImageFormat	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= JPEG 1= GIF 2= PNG 3= BMP 4= TIFF 5= PPM 6= PGM 7= PNM 8= NEF 9= CR2 10= DNG

4.10.5.1.2 ID 281C: QueryStillImageDataExt

This message shall cause the receiving component to reply to the requestor with a Report Still Image Data Ext message. By setting the associated bits in the QueryPresenceVector, the client can request that the response contain the image data directly in the ImageFrame field, or the image data can be stored locally by the sensor component and a file name or other identifier for the data can be returned in the StillImageURI field, or both. 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 94 - QueryStillImageDataExt message encoding

body └ **list** name = QueryStillImageDataList │ (count_field = unsigned short integer) └ **record** name = QueryStillImageDataRec					
Record Name = QueryStillImageDataRec					
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>** ReportCoordinateSystem	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Native Coordinate System 1= Vehicle Coordinate System
3	**<fixed_field>** QueryPresenceVector	unsigned byte	one	false	

4.10.5.1.3 ID 2813: QueryStillImageSensorConfiguration

This message shall cause the receiving service to reply to the requestor with a Report Still Image 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 95 - QueryStillImageSensorConfiguration message encoding

body └ **list** name = QueryStillImageSensorConfigurationList │ (count_field = unsigned short integer) └ **record** name = QueryStillImageSensorConfigurationRec					
Record Name = QueryStillImageSensorConfigurationRec					
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	

4.10.5.1.4 ID 2812: QueryStillImageSensorCapabilities

This message shall cause the receiving service to reply to the requestor with a Report Still Image Sensor Capabilities 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 96 - QueryStillImageSensorCapabilities message encoding

body └ **list** name = QueryStillImageSensorCapabilitiesList │ (count_field = unsigned short integer) └ **record** name = QueryStillImageSensorCapabilitiesRec					
Record Name = QueryStillImageSensorCapabilitiesRec					
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	

4.10.5.2 Output Set

4.10.5.2.1 ID 481C: ReportStillImageDataExt

This message is used to report the most recent still image that was taken by the sensor. The timestamp defines the time at which the collected data was valid. The image data is defined with respect to either the sensor coordinate system or a specified coordinate system.

Table 97 - ReportStillImageDataExt message encoding

body └ **list** name = StillImageDataList │ (count_field = unsigned short integer) └ **record** name = StillImageDataRec					
Record Name = StillImageDataRec					
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>** ReportCoordinateSystem	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Native Coordinate System 1= Vehicle Coordinate System
4	**<bit_field>** TimeStamp	unsigned integer	bit_field	true	Bits 0..9, Milliseconds: Value set, offset=false, ranges/enums: [0,999] Bits 10..15, Seconds: Value set, offset=false, ranges/enums: [0,59] Bits 16..21, Minutes: Value set, offset=false, ranges/enums: [0,59] Bits 22..26, Hour: Value set, offset=false, ranges/enums: [0,23] Bits 27..31, Day: Value set, offset=false, ranges/enums: [1,31]

5	**<variable_format_field>** ImageFrame	count_field = unsigned integer	N/A	true	(min/max count = 0..2147483647) Format_enum: 0 = JPEG 1 = GIF 2 = PNG 3 = BMP 4 = TIFF 5 = PPM 6 = PGM 7 = PNM 8 = NEF 9 = CR2 10 = DNG
6	**<variable_length_string>** StillImageURI	variable length string (byte[])	N/A	true	Filename or other URI indicating the location of the still image data captured by the associated request. If the saving process fails for any reason (insufficient storage space, I/O error), the URI shall be an empty (zero length) string. (Length min..max = 0..255)

4.10.5.2.2 ID 4813: ReportStillImageSensorConfiguration

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

Table 98 - ReportStillImageSensorConfiguration message encoding

body
└ **list** name = StillImageSensorConfigurationList
(count_field = unsigned short integer)
└ **record** name = StillImageSensorConfigurationRec

Record Name = StillImageSensorConfigurationRec

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>** 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)

4	**<fixed_field>** StillImageFormat	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= JPEG 1= GIF 2= PNG 3= BMP 4= TIFF 5= PPM 6= PGM 7= PNM 8= NEF 9= CR2 10= DNG

4.10.5.2.3 ID 4812: ReportStillImageSensorCapabilities

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

Table 99 - ReportStillImageSensorCapabilities message encoding

body
└ **list** name = StillImageSensorList
  (count_field = unsigned short integer)
  └ **record** name = StillImageSensorCapabilitiesRec

Record Name = StillImageSensorCapabilitiesRec

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	**<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, wqsxga_3200x2048: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 21..21, wquxga_3840x2400: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 22..22, whsxga_6400x4096: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 23..23, whuxga_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
4	**<bit_field>** SupportedImageFormats	unsigned short integer	bit_field	true	Bits 0..0, JPEG: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, GIF: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, PNG: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, BMP: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, TIFF: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, PPM: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, PGM: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 7..7, PNM: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 8..8, NEF_Nikon_RAW: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 9..9, CR2_Canon_RAW: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 10..10, DNG_Adobe_RAW: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported

4.10.6 Protocol Behavior

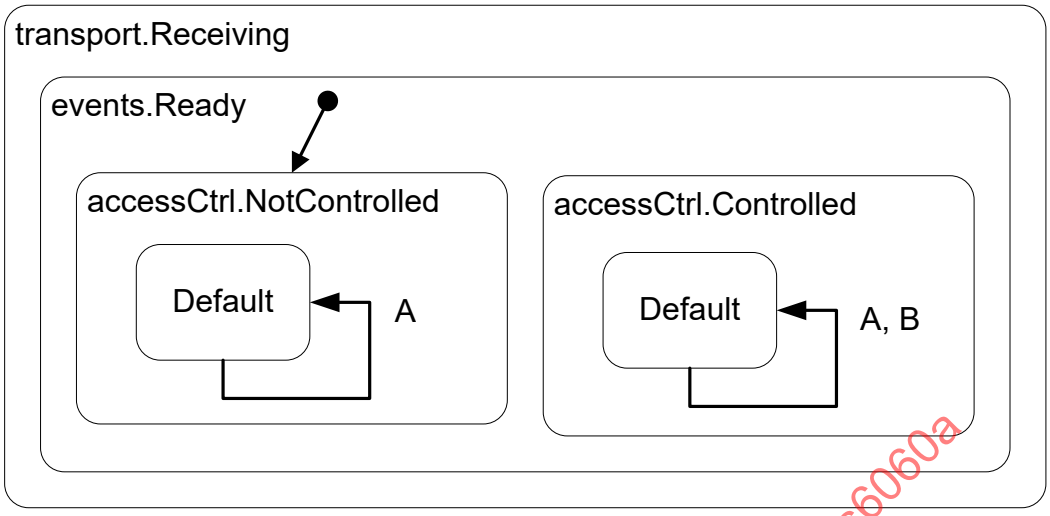


Figure 11 - StillImage service protocol behavior

Table 100 - StillImage service state transitions

Start State	Trigger	Conditions	Actions
A	QueryStillImageSensorCapabilities		sendReportStillImageSensorCapabilities (msg,transportData)
A	QueryStillImageSensorConfiguration		sendReportStillImageSensorConfiguration (msg,transportData)
A	QueryStillImageDataExt	isCoordinateTransform Supported(msg)	captureStillImage (msg) , sendReportStillImageDataExtInRequestedCoordinateSystem (msg,transportData)
A	QueryStillImageDataExt	! isCoordinateTransform Supported(msg)	captureStillImage (msg) , sendReportStillImageDataExtInNativeCoordinateSystem (msg,transportData)
B	SetStillImageSensorConfiguration	isControllingClient(transportData)	sendConfirmSensorConfiguration (msg, transportData) , updateStillImageSensorConfiguration (msg)

Table 101 - StillImage service conditions

Condition	Interpretation
isCoordinateTransformSupported(msg)	True if the service supports coordinate system transformations for data reports.
! isCoordinateTransformSupported(msg)	True if the service does not support coordinate system transformations for data reports.
isControllingClient(transportData)	True if the message that triggered the transition is received from the client that is in control of this service.

Table 102 - StillImage service transition actions

Action	Interpretation
sendReportStillImageSensorCapabilities	Send a ReportStillImageSensorCapabilities message
sendReportStillImageSensorConfiguration	Send a ReportStillImageSensorConfiguration message
captureStillImage	Trigger a still image capture from the specified sensor(s)
sendReportStillImageDataExtInRequestedCoordinateSystem	Send a ReportStillImageDataExt message using the requested coordinate system
sendReportStillImageDataExtInNative CoordinateSystem	Send a ReportStillImageDataExt message in native coordinate system
sendConfirmSensorConfiguration	Send sendConfirmSensorConfiguration message
updateStillImageSensorConfiguration	Update the sensor user controllable configuration parameters according to the ones specified.

4.11 VideoIlluminator

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

4.11.1 Description

The Video Illuminator Service provides support for and control of Illumination devices associated with video streaming. The Video Illuminator Service is associated with specific cameras by implementing both the video service and the illuminator service within a single enclosing component, and correlating the Illuminator ID with one or more Sensor IDs in the Capabilities message.

4.11.2 Assumptions

4.11.3 References

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

4.11.4 Vocabulary

Table 103 - VideoIlluminator service message vocabulary

Message ID (hex)	Name	Command
Input Set		
081A	SetVideoIlluminatorConfiguration	true
281A	QueryVideoIlluminatorConfiguration	false
281B	QueryVideoIlluminatorCapabilities	false
Output Set		
0801	ConfirmSensorConfiguration	false
481A	ReportVideoIlluminatorConfiguration	false
481B	ReportVideoIlluminatorCapabilities	false

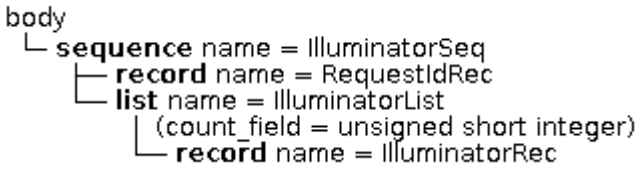
4.11.5 Encoding

4.11.5.1 Input Set

4.11.5.1.1 ID 081A: SetVideoIlluminatorConfiguration

This message commands a level of illumination output to the device associated with the specified illuminatorID.

Table 104 - SetVideoIlluminatorConfiguration message encoding



Record Name = RequestIdRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** RequestID	unsigned byte	one	false	
Record Name = IlluminatorRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<presence_vector>**	unsigned byte			
2	**<fixed_field>** IlluminatorID	unsigned short integer	one	false	Identifies an illumination device supported by the service. Zero is not a valid value.
3	**<fixed_field>** Level	unsigned short integer	percent	false	Desired intensity level, as a percent of capability. If the illuminator does not support variable levels of brightness, any non-zero value shall be interpreted as on, and a zero level value shall be interpreted as off. (scaled range = [0,100], round)
4	**<fixed_field>** Mode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= VISIBLE 1= INFRARED 2= GREYSCALE 3= LOWLIGHT [4,20] These values are reserved for future use [21,31] These values are reserved for program- or implementation-specific use
5	**<fixed_field>** BeamWidth	unsigned short integer	degree	true	(scaled range = [0,360], round)
6	**<fixed_field>** BeamHeight	unsigned short integer	degree	true	(scaled range = [0,360], round)

4.11.5.1.2 ID 281A: QueryVideoIlluminatorConfiguration

This message requests a report of the commanded illumination output level of the illuminator device associated with the specified illuminatorID(s).

Table 105 - QueryVideoIlluminatorConfiguration message encoding

body
└ **list** name = IlluminatorIDList
  │ (count_field = unsigned short integer)
  └ **record** name = IlluminatorIDRec					

Record Name = IlluminatorIDRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** IlluminatorID	unsigned short integer	one	false	Identifies an illumination device supported by the service. A value of zero shall be interpreted as query for all available illuminators.

4.11.5.1.3 ID 281B: QueryVideoIlluminatorCapabilities

This message requests a report of the capabilities of the illuminator device associated with the specified illuminatorID(s).

Table 106 - QueryVideoIlluminatorCapabilities message encoding

body
└ **list** name = IlluminatorIDList
  │ (count_field = unsigned short integer)
  └ **record** name = IlluminatorIDRec					

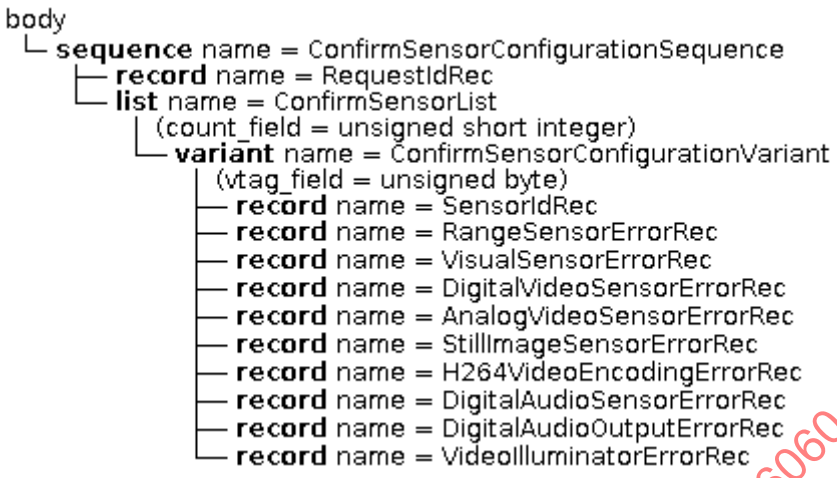
Record Name = IlluminatorIDRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** IlluminatorID	unsigned short integer	one	false	Identifies an illumination device supported by the service. A value of zero shall be interpreted as query for all available illuminators.

4.11.5.2 Output Set

4.11.5.2.1 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 107 - ConfirmSensorConfiguration message encoding



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

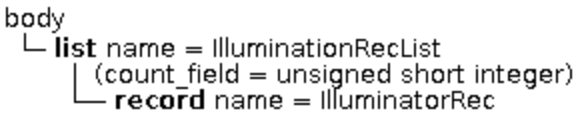
Field #	Name	Type	Units	Optional	Interpretation

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)

4.11.5.2.2 ID 481A: ReportVideoIlluminatorConfiguration

This message provides a report of the commanded illumination level of the illumination devices associated with the specified illuminatorID(s).

Table 108 - ReportVideoIlluminatorConfiguration message encoding



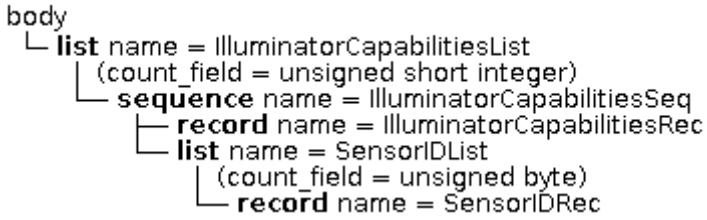
Record Name = IlluminatorRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned byte			
2	**<fixed_field>** IlluminatorID	unsigned short integer	one	false	Identifies an illumination device supported by the service. Zero is not a valid value.
3	**<fixed_field>** Level	unsigned short integer	percent	false	Requested intensity level, as a percent of capability. If the illuminator does not support variable levels of brightness, any non-zero value shall be interpreted as on, and a zero level value shall be interpreted as off. (scaled range = [0,100], round)
4	**<fixed_field>** Mode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= VISIBLE 1= INFRARED 2= GREYSCALE 3= LOWLIGHT [4,20] These values are reserved for future use [21,31] These values are reserved for program- or implementation-specific use
5	**<fixed_field>** BeamWidth	unsigned short integer	degree	true	(scaled range = [0,360], round)
6	**<fixed_field>** BeamHeight	unsigned short integer	degree	true	(scaled range = [0,360], round)

4.11.5.2.3 ID 481B: ReportVideoIlluminatorCapabilities

This message provides a report of the capabilities of the specified illuminator device.

Table 109 - ReportVideoIlluminatorCapabilities message encoding



Record Name = IlluminatorCapabilitiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned byte			
2	**<fixed_field>** IlluminatorID	unsigned short integer	one	false	Identifies an illumination device supported by the service. Zero is not a valid value.
3	**<variable_length_string>** IlluminatorName	variable length string (byte[])	N/A	false	(Length min..max = 0..255)

4	**<bit_field>** SupportedIlluminationModes	unsigned integer	bit_field	false	Bits 0..0, Visible: Traditional white light source, visible to the human eye Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 1..1, Infrared: Traditional IR light source, invisible to the human eye Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 2..2, GreyScale: Illumination type suitable for use in greyscale imaging. Specific interpretation of this mode is not defined by the Standard, and may vary between implementations. Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 3..3, LowLight: Illumination type suitable for use in lowlight imaging. Specific interpretation of this mode is not defined by the Standard, and may vary between implementations. Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 4..20, ReservedForFutureUse: Value set, offset=false, ranges/enums: [0,1] Bits 21..31, ReservedForProgramOrImplementationSpecificUse: Value set, offset=false, ranges/enums: [0,1]
5	**<fixed_field>** MinimumIntensity	unsigned short integer	lumen	true	Minimum achievable brightness level. If this value is the same as the maximum, the illuminator is assumed to not support variable levels, and acts in an on/off fashion only.
6	**<fixed_field>** MaximumIntensity	unsigned short integer	lumen	true	Maximum achievable brightness level. If this value is the same as the minimum, the illuminator is assumed to not support variable levels, and acts in an on/off fashion only.
7	**<fixed_field>** MinimumSupportedBeamWidth	unsigned short integer	degree	true	(scaled range = [0,360], round)
8	**<fixed_field>** MaximumSupportedBeamWidth	unsigned short integer	degree	true	(scaled range = [0,360], round)
9	**<fixed_field>** MinimumSupportedBeamHeight	unsigned short integer	degree	true	(scaled range = [0,360], round)
10	**<fixed_field>** MaximumSupportedBeamHeight	unsigned short integer	degree	true	(scaled range = [0,360], round)
Record Name = SensorIDRec					
Field #	**Name**	**Type**	**Units**	**Optional**	**Interpretation**
1	**<fixed_field>** SensorID	unsigned short integer	one	false	SensorID of a camera associated with this illuminator.

4.11.6 Protocol Behavior

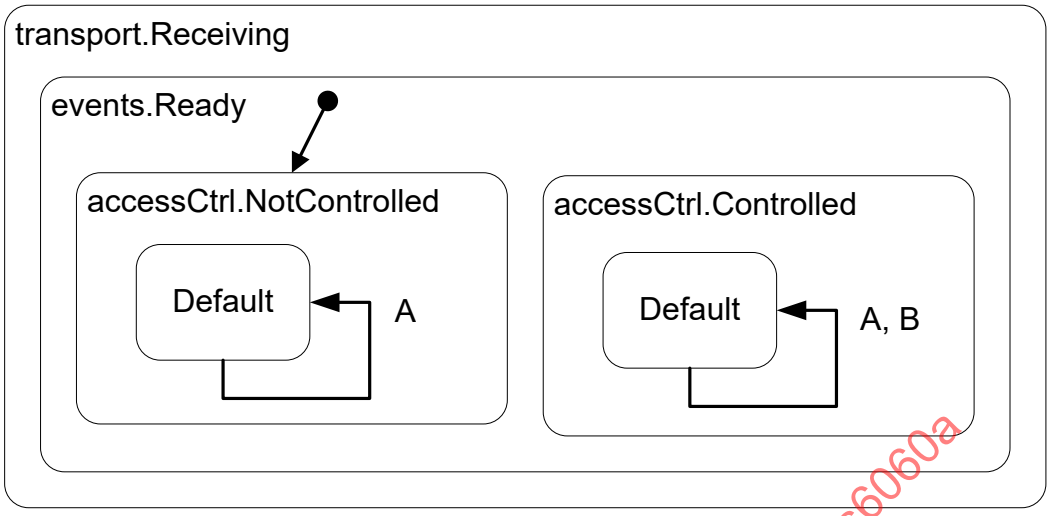


Figure 12 - VideoIlluminator service protocol behavior

Table 110 - VideoIlluminator service state transitions

Start State	Trigger	Conditions	Actions
B	SetVideoIlluminatorConfiguration	isControllingClient(transportData)	setValidVideoIlluminatorConfigurationAction (msg) , sendConfirmSensorConfiguration (msg, transportData)
A	QueryVideoIlluminatorCapabilities		sendReportVideoIlluminatorCapabilities (msg, transportData)
A	QueryVideoIlluminatorConfiguration		sendReportVideoIlluminatorConfiguration (msg, transportData)

Table 111 - VideoIlluminator 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 112 - VideoIlluminator service transition actions

Action	Interpretation
setValidVideoIlluminatorConfigurationAction	Set the illuminator configuration specified in the message if valid for the given ID
sendConfirmSensorConfiguration	Send sendConfirmSensorConfiguration message with confirmation or error code for each specified illuminator ID
sendReportVideoIlluminatorCapabilities	Construct and send ReportVideoIlluminatorCapabilities to querying client
sendReportVideoIlluminatorConfiguration	Construct and send ReportVideoIlluminatorConfiguration to querying client

4.12 VisualSensor

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

4.12.1 Description

This service provides access to the basic capabilities and configuration of a visual sensor, allowing the controlling component to set the visual sensor to a particular operational profile. The Query Sensor Geometric Properties message can be used to determine the geometric relationship between the sensor and the vehicle coordinate system. Three possible coordinate responses are possible; (a) the service does not know the sensor's position, (b) the sensor coordinate system is fixed with respect to the vehicle and (c) the sensor is attached to some manipulator. These cases are supported by the Report Sensor Geometric Properties message and are described therein.

4.12.2 Assumptions

Messages may be delayed, lost, or reordered.

4.12.3 References

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

4.12.4 Vocabulary

Table 113 - VisualSensor service message vocabulary

Message ID (hex)	Name	Command
Input Set		
2805	QuerySensorGeometricProperties	false
2807	QueryVisualSensorConfiguration	false
2806	QueryVisualSensorCapabilities (Deprecated)	false
280A	QueryVisualSensorCapabilitiesExt	false
0803	SetVisualSensorConfiguration	true
Output Set		
0801	ConfirmSensorConfiguration	false
4805	ReportSensorGeometricProperties	false
4807	ReportVisualSensorConfiguration	false
4806	ReportVisualSensorCapabilities (Deprecated)	false
480A	ReportVisualSensorCapabilitiesExt	false

4.12.5 Encoding

4.12.5.1 Input Set

4.12.5.1.1 ID 2805: QuerySensorGeometricProperties

This message shall cause the receiving component to reply to the requestor with a Report Sensor Geometric Properties message.

Table 114 - QuerySensorGeometricProperties 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

4.12.5.1.2 ID 2807: QueryVisualSensorConfiguration

This message must be used to query for the current configuration of the given visual sensor(s). It shall cause the receiving service to reply to the requestor with a Report Visual 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 115 - QueryVisualSensorConfiguration message encoding

body └ **list** name = QueryVisualSensorConfigurationList │ (count_field = unsigned short integer) └ **record** name = QueryVisualSensorConfigurationRec					
Record Name = QueryVisualSensorConfigurationRec					
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 short integer	one	false	

4.12.5.1.3 ID 2806: QueryVisualSensorCapabilities (Deprecated)

QueryVisualSensorCapabilities is deprecated and is scheduled to be removed during a future revision.

This message must be used to query for the capabilities of all available visual sensors on the receiving service. It shall cause the receiving service to reply to the requestor with a Report Visual Sensor Capabilities 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 116 - QueryVisualSensorCapabilities message encoding

body └ **list** name = QueryVisualSensorCapabilitiesList (count_field = unsigned short integer) └ **record** name = QueryVisualSensorCapabilitiesRec					
Record Name = QueryVisualSensorCapabilitiesRec					
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 short integer	one	false	

4.12.5.1.4 ID 280A: QueryVisualSensorCapabilitiesExt

This message must be used to query for the extended capabilities of all available visual sensors on the receiving service. It shall cause the receiving service to reply to the requestor with a Report Visual 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.

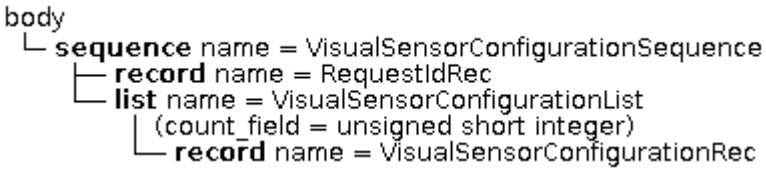
Table 117 - QueryVisualSensorCapabilitiesExt message encoding

body └ **list** name = QueryVisualSensorCapabilitiesList (count_field = unsigned short integer) └ **record** name = QueryVisualSensorCapabilitiesRec					
Record Name = QueryVisualSensorCapabilitiesRec					
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 integer	one	false	

4.12.5.1.5 ID 0803: SetVisualSensorConfiguration

This message is used to set the configuration of the visual sensors associated with the service. Configuration is based off of each sensor's capabilities as described in the Report Visual 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 Visual Sensor Error Record for the given Sensor ID however other, valid, configurations specified shall be set (if they exist).

Table 118 - SetVisualSensorConfiguration message encoding



Record Name = RequestIdRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** RequestID	unsigned byte	one	false	

Record Name = VisualSensorConfigurationRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned short integer			
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>** SensorState	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Active 1= Standby 2= Off
4	**<fixed_field>** ZoomMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Mixed (When active, this mode allows both analog (optical) and digital zoom.) 1= Analog Only (When active, this mode restricts zoom to analog (optical) capabilities only.) 2= Digital Only (When active, this mode restricts zoom to digital capabilities only. This may result in a loss of resolution.) 3= Off (When active, this mode prohibits any zoom.)
5	**<fixed_field>** ZoomLevel	unsigned short integer	percent	true	(scaled range = [0.0,100.0], round)
6	**<fixed_field>** FocalLength	unsigned integer	meter	true	(scaled range = [0.0,2.0], round)
7	**<fixed_field>** HorizontalFieldOfView	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
8	**<fixed_field>** VerticalFieldOfView	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
9	**<fixed_field>** FocusMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto Focus 1= Manual Focus
10	**<fixed_field>** FocusValue	unsigned short integer	percent	true	(scaled range = [0.0,100.0], round)
11	**<fixed_field>** WhiteBalance	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto 1= Daylight 2= Cloudy 3= Shade 4= Tungsten 5= Flurescent 6= Flash

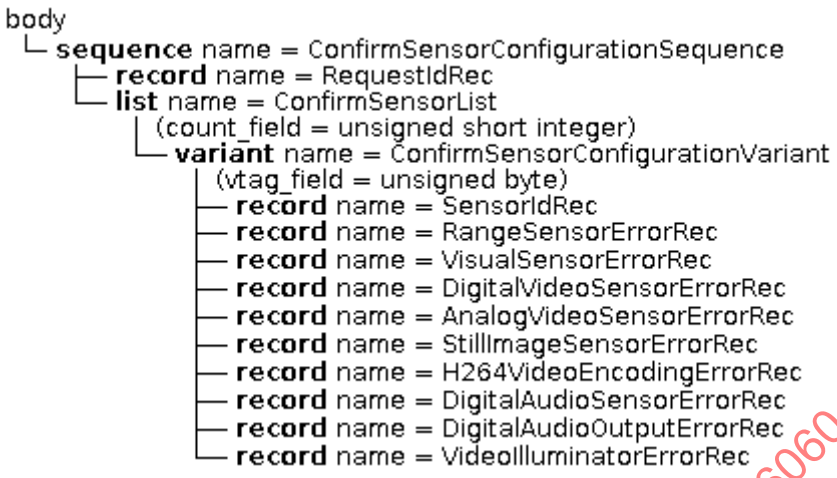
12	**<fixed_field>** ImagingMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Color 1= Greyscale 2= Infrared 3= Low Light 4= Thermal_WhiteHot 5= Thermal_BlackHot 6= Thermal_Rainbow [7,20] Reserved for future use [21,31] Reserved for program, implementation, or sensor specific modes
13	**<fixed_field>** ExposureMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto 1= Manual 2= Shutter Priority 3= Aperture Priority
14	**<fixed_field>** MeteringMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto 1= CenterWeighted 2= Spot
15	**<fixed_field>** ShutterSpeed	unsigned short integer	second	true	(scaled range = [0.0,60.0], round)
16	**<fixed_field>** Aperture	unsigned short integer	one	true	(scaled range = [0.1,128.0], round)
17	**<fixed_field>** LightSensitivity	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto 1= ISO 100 2= ISO 200 3= ISO 400 4= ISO 800 5= ISO 1600 6= ISO 3200
18	**<fixed_field>** ImageStabilization	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Off 1= On

4.12.5.2 Output Set

4.12.5.2.1 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 119 - ConfirmSensorConfiguration message encoding



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

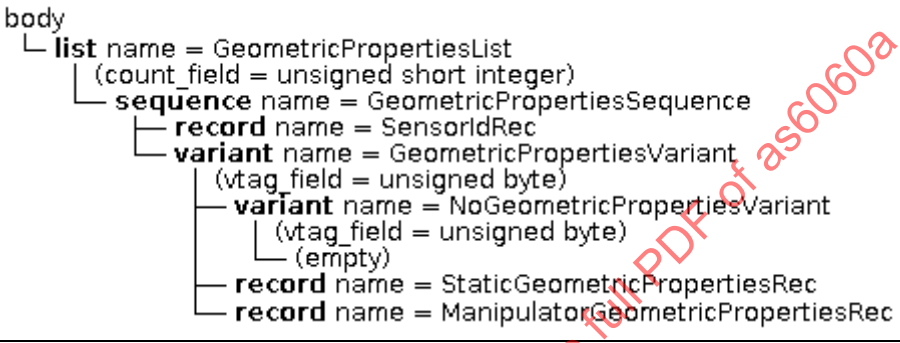
Field #	Name	Type	Units	Optional	Interpretation

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)

4.12.5.2.2 ID 4805: ReportSensorGeometricProperties

This message is used to report information about the geometric properties of a sensor with respect to the vehicle. Three possible use cases are supported: 1.The sensor has no knowledge of its geometric properties. 2.The sensor is rigidly mounted to the vehicle and the position and orientation with respect to the vehicle coordinate system is known. In this case the position and orientation of the sensor is defined by a position vector (x, y, z) and unit quaternion (d; a, b, c) which specifies the axis and angle of rotation used to establish the orientation of the sensor coordinate system with respect to the vehicle coordinate system. 3.The sensor is mounted to a manipulator in which the JAUS ID, joint number and the sensor offset are described. In this case the position and orientation of the sensor is defined by a position vector (x, y, z) and unit quaternion (d; a, b, c) which specifies the axis and angle of rotation used to establish the orientation of the sensor coordinate system with respect to the link coordinate system.

Table 120 - ReportSensorGeometricProperties message encoding



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 = StaticGeometricPropertiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<array>** SensorPosition	Array of unsigned integer	N/A	false	dimension size = 3 <fixed_field>
2	**<array>** UnitQuaternion	Array of unsigned integer	N/A	false	dimension size = 4 <fixed_field>

Record Name = ManipulatorGeometricPropertiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<fixed_field>** SubsystemID	unsigned short integer	one	false	
2	**<fixed_field>** NodeID	unsigned byte	one	false	
3	**<fixed_field>** ComponentID	unsigned byte	one	false	
4	**<fixed_field>** JointNumber	unsigned byte	one	false	
5	**<array>** SensorPosition	Array of unsigned integer	N/A	false	dimension size = 3 <fixed_field>
6	**<array>** UnitQuaternion	Array of unsigned integer	N/A	false	dimension size = 4 <fixed_field>

4.12.5.2.3 ID 4807: ReportVisualSensorConfiguration

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

Table 121 - ReportVisualSensorConfiguration message encoding

body
└ **list** name = VisualSensorConfigurationList
(count_field = unsigned short integer)
└ **record** name = VisualSensorConfigurationRec

Record Name = VisualSensorConfigurationRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned short integer			
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>** SensorState	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Active 1= Standby 2= Off
4	**<fixed_field>** ZoomMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Mixed (When active, this mode allows both analog (optical) and digital zoom.) 1= Analog Only (When active, this mode restricts zoom to analog (optical) capabilities only.) 2= Digital Only (When active, this mode restricts zoom to digital capabilities only. This may result in a loss of resolution.) 3= Off (When active, this mode prohibits any zoom.)
5	**<fixed_field>** ZoomLevel	unsigned short integer	percent	true	(scaled range = [0.0,100.0], round)
6	**<fixed_field>** FocalLength	unsigned integer	meter	true	(scaled range = [0.0,2.0], round)
7	**<fixed_field>** HorizontalFieldOfView	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
8	**<fixed_field>** VerticalFieldOfView	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
9	**<fixed_field>** FocusMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto Focus 1= Manual Focus
10	**<fixed_field>** FocusValue	unsigned short integer	percent	true	(scaled range = [0.0,100.0], round)
11	**<fixed_field>** WhiteBalance	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto 1= Daylight 2= Cloudy 3= Shade 4= Tungsten 5= Flurescent 6= Flash

12	**<fixed_field>** ImagingMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Color 1= Greyscale 2= Infrared 3= Low Light 4= Thermal_WhiteHot 5= Thermal_BlackHot 6= Thermal_Rainbow [7,20] Reserved for future use [21,31] Reserved for program, implementation, or sensor specific modes
13	**<fixed_field>** ExposureMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto 1= Manual 2= Shutter Priority 3= Aperture Priority
14	**<fixed_field>** MeteringMode	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto 1= CenterWeighted 2= Spot
15	**<fixed_field>** ShutterSpeed	unsigned short integer	second	true	(scaled range = [0.0,60.0], round)
16	**<fixed_field>** Aperture	unsigned short integer	one	true	(scaled range = [0.1,128.0], round)
17	**<fixed_field>** LightSensitivity	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Auto 1= ISO 100 2= ISO 200 3= ISO 400 4= ISO 800 5= ISO 1600 6= ISO 3200
18	**<fixed_field>** ImageStabilization	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= Off 1= On

4.12.5.2.4 ID 4806: ReportVisualSensorCapabilities (Deprecated)

ReportVisualSensorCapabilities is deprecated and is scheduled to be removed during a future revision.

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

Table 122 - ReportVisualSensorCapabilities message encoding

body
└ **list** name = VisualSensorCapabilitiesList
(count_field = unsigned short integer)
└ **record** name = VisualSensorCapabilitiesRec

Record Name = VisualSensorCapabilitiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned short integer			
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	**<variable_length_string>** SensorName	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
4	**<bit_field>** SupportedStates	unsigned byte	bit_field	true	Bits 0..0, Active: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Standby: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Off: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
5	**<bit_field>** ZoomModes	unsigned byte	bit_field	true	While many cameras offer zoom using purely analog (or optical) or digital techniques, some use a 'mixed' mode where both are used in combination. This is often of the form of analog (optical) zoom to its physical maximum, e.g., 10x, and digital beyond that to achieve higher magnifications, e.g., 150x. Those cameras that offer mixed mode can usually be configured to perform in a purely analog or digital function, effectively shutting off the other zoom behavior. This bitfield specifies which zoom techniques are available in the interface. Bits 0..0, Mixed: When active, this mode allows both analog (optical) and digital zoom. Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, AnalogOnly: When active, this mode restricts zoom to analog (optical) capabilities only. Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, DigitalOnly: When active, this mode restricts zoom to digital capabilities only. This may result in a loss of resolution. Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, None: When active, this mode prohibits any zoom. Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported

6	**<bit_field>** FocusModes	unsigned byte	bit_field	true	Bits 0..0, AutoFocus: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, ManualFocus: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
7	**<bit_field>** WhiteBalance	unsigned byte	bit_field	true	Bits 0..0, Auto: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Daylight: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Cloudy: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, Shade: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, Tungsten: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, Flurescent: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, Flash: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
8	**<bit_field>** ImagingModes	unsigned byte	bit_field	true	Bits 0..0, Color: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Greyscale: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Infrared: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, Lowlight: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, ThermalWhiteHot: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, ThermalBlackHot: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, ThermalRainbow: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 7..7, ReservedForFutureUse: Value set, offset=false, ranges/enums: 0= Reserved

9	**<bit_field>** ExposureModes	unsigned byte	bit_field	true	Bits 0..0, Auto: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Manual: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, ShutterPriority: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, AperturePriority: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
10	**<bit_field>** MeteringModes	unsigned byte	bit_field	true	Bits 0..0, MatrixOrAuto: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, CenterWeighted: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Spot: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
11	**<fixed_field>** MinimumShutterSpeed	unsigned short integer	second	true	(scaled range = [0.0,60.0], round)
12	**<fixed_field>** MaximumShutterSpeed	unsigned short integer	second	true	(scaled range = [0.0,60.0], round)
13	**<fixed_field>** MinimumAperture	unsigned short integer	one	true	(scaled range = [0.1,128.0], round)
14	**<fixed_field>** MaximumAperture	unsigned short integer	one	true	(scaled range = [0.1,128.0], round)
15	**<fixed_field>** MinimumFocalLength	unsigned integer	meter	true	(scaled range = [0.0,2.0], round)
16	**<fixed_field>** MaximumFocalLength	unsigned integer	meter	true	(scaled range = [0.0,2.0], round)
17	**<bit_field>** LightSensitivityLevels	unsigned byte	bit_field	true	Bits 0..0, Auto: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, ISO100: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, ISO200: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, ISO400: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, ISO800: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, ISO1600: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, ISO3200: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported

18	**<fixed_field>** ImageStabilization	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= False 1= True

4.12.5.2.5 ID 480A: ReportVisualSensorCapabilitiesExt

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

Table 123 - ReportVisualSensorCapabilitiesExt message encoding

body
└ **list** name = VisualSensorCapabilitiesList
  (count_field = unsigned short integer)
  └ **record** name = VisualSensorCapabilitiesRec

Record Name = VisualSensorCapabilitiesRec

Field #	Name	Type	Units	Optional	Interpretation
1	**<presence_vector>**	unsigned integer			
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	**<variable_length_string>** SensorName	variable length string (byte[])	N/A	false	(Length min..max = 0..255)
4	**<bit_field>** SupportedStates	unsigned byte	bit_field	true	Bits 0..0, Active: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Standby: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Off: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
5	**<bit_field>** ZoomModes	unsigned byte	bit_field	true	While many cameras offer zoom using purely analog (or optical) or digital techniques, some use a 'mixed' mode where both are used in combination. This is often of the form of analog (optical) zoom to its physical maximum, e.g., 10x, and digital beyond that to achieve higher magnifications, e.g., 150x. Those cameras that offer mixed mode can usually be configured to perform in a purely analog or digital function, effectively shutting off the other zoom behavior. This bitfield specifies which zoom techniques are available in the interface. Bits 0..0, Mixed: When active, this mode allows both analog (optical) and digital zoom. Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, AnalogOnly: When active, this mode restricts zoom to analog (optical) capabilities only. Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, DigitalOnly: When active, this mode restricts zoom to digital capabilities only. This may result in a loss of resolution.

					Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, None: When active, this mode prohibits any zoom. Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
6	**<bit_field>** FocusModes	unsigned byte	bit_field	true	Bits 0..0, AutoFocus: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, ManualFocus: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
7	**<bit_field>** WhiteBalance	unsigned byte	bit_field	true	Bits 0..0, Auto: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Daylight: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Cloudy: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, Shade: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, Tungsten: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, Flurescent: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, Flash: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
8	**<bit_field>** ImagingModes	unsigned integer	bit_field	true	Bits 0..0, Color: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Greyscale: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Infrared: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, Lowlight: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, ThermalWhiteHot: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, ThermalBlackHot: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, ThermalRainbow: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 7..20, ReservedForFutureUse:

					Value set, offset=false, ranges/enums: 0= Reserved Bits 21..31, ReservedForProgramSpecificUse: Value set, offset=false, ranges/enums: 0= Reserved
9	**<bit_field>** ExposureModes	unsigned byte	bit_field	true	Bits 0..0, Auto: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, Manual: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, ShutterPriority: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, AperturePriority: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
10	**<bit_field>** MeteringModes	unsigned byte	bit_field	true	Bits 0..0, MatrixOrAuto: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, CenterWeighted: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, Spot: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
11	**<fixed_field>** MinimumShutterSpeed	unsigned short integer	second	true	(scaled range = [0.0,60.0], round)
12	**<fixed_field>** MaximumShutterSpeed	unsigned short integer	second	true	(scaled range = [0.0,60.0], round)
13	**<fixed_field>** MinimumAperture	unsigned short integer	one	true	(scaled range = [0.1,128.0], round)
14	**<fixed_field>** MaximumAperture	unsigned short integer	one	true	(scaled range = [0.1,128.0], round)
15	**<fixed_field>** MinimumFocalLength	unsigned integer	meter	true	(scaled range = [0.0,2.0], round)
16	**<fixed_field>** MaximumFocalLength	unsigned integer	meter	true	(scaled range = [0.0,2.0], round)
17	**<bit_field>** LightSensitivityLevels	unsigned byte	bit_field	true	Bits 0..0, Auto: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 1..1, ISO100: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 2..2, ISO200: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 3..3, ISO400: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 4..4, ISO800: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 5..5, ISO1600:

					Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported Bits 6..6, ISO3200: Value set, offset=false, ranges/enums: 0= Unsupported 1= Supported
18	**<fixed_field>** ImageStabilization	unsigned byte	one	true	Value set, offset=false, ranges/enums: 0= False 1= True
19	**<fixed_field>** MinimumHorizontalFieldOfView	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
20	**<fixed_field>** MaximumHorizontalFieldOfView	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
21	**<fixed_field>** MinimumVerticalFieldOfView	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)
22	**<fixed_field>** MaximumVerticalFieldOfView	unsigned integer	radian	true	(scaled range = [-3.141592653589793,3.141592653589793], round)

4.12.6 Protocol Behavior

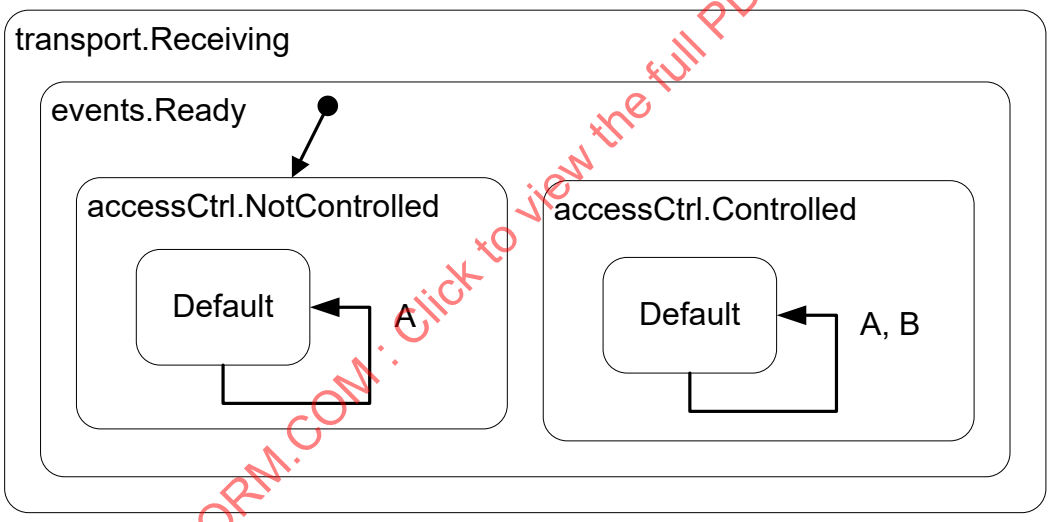


Figure 13 - VisualSensor service protocol behavior

Table 124 - VisualSensor service state transitions

Start State	Trigger	Conditions	Actions
A	QueryVisualSensorCapabilities		sendReportVisualSensorCapabilities (msg, transportData)
A	QueryVisualSensorCapabilitiesExt		sendReportVisualSensorCapabilitiesExt (msg,transportData)
A	QueryVisualSensorConfiguration		sendReportVisualSensorConfiguration (msg,transportData)
A	QuerySensorGeometricProperties		sendReportSensorGeometricProperties (msg,transportData)
B	SetVisualSensorConfiguration	isControllingClient(transportData)	sendConfirmSensorConfiguration (msg, transportData) , updateVisualSensorConfiguration (msg)

Table 125 - VisualSensor 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 126 - VisualSensor service transition actions

Action	Interpretation
sendReportVisualSensorCapabilities	Send a ReportVisualSensorCapabilities message
sendReportVisualSensorCapabilitiesExt	Send a ReportVisualSensorCapabilitiesExt message
sendReportVisualSensorConfiguration	Send a ReportVisualSensorConfiguration message
sendReportSensorGeometricProperties	Send a ReportSensorGeometricProperties message
sendConfirmSensorConfiguration	Send sendConfirmSensorConfiguration message
updateVisualSensorConfiguration	Update the sensor user controllable configuration parameters according to the ones specified.

5. NOTES

5.1 Revision Indicator

A change bar (I) located in the left margin is for the convenience of the user in locating areas where technical revisions, not editorial changes, have been made to the previous issue of this document. An (R) symbol to the left of the document title indicates a complete revision of the document, including technical revisions. Change bars and (R) are not used in original publications, nor in documents that contain editorial changes only.

PREPARED BY THE AS-4JAUS JOINT ARCHITECTURE FOR UNMANNED SYSTEMS COMMITTEE

APPENDIX A - XML FOR SERVICE DEFINITIONS

This appendix contains JSIDL XML text of the documented services.

A.1 ANALOGVIDEO

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="AnalogVideo" id="urn:jaus:jss:environmentSensing:AnalogVideo" version="2.0"
xmlns="urn:jaus:jsidl:1.1" xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">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.</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost or
  reordered.</assumptions>
    <references>
        <inherits_from name="visualSensor" id="urn:jaus:jss:environmentSensing:VisualSensor" version="2.0"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="SetAnalogVideoSensorConfiguration" message_id="0806" is_command="true">
                <description xml:space="preserve">This message is used to set the configuration of the analog video
sensor associated with the service. Configuration is based off of 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).</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="AnalogVideoSensorConfigurationSequence" optional="false">
                        <record name="RequestIdRec" optional="false">
                            <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                        <list name="AnalogVideoSensorList" optional="false">
                            <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                            <record name="AnalogVideoSensorConfigurationRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is
not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="AnalogFormat" optional="false" field_type="unsigned byte"
field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NTSC-M"/>
                                        <value_enum enum_index="1" enum_const="NTSC-J"/>
                                        <value_enum enum_index="2" enum_const="PAL-N"/>
                                        <value_enum enum_index="3" enum_const="PAL-M"/>
                                        <value_enum enum_index="4" enum_const="SECAM-L"/>
                                        <value_enum enum_index="5" enum_const="SECAM-B/G"/>
                                    </value_set>
                                </fixed_field>
                            </record>
                        </list>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryAnalogVideoSensorConfiguration" message_id="2811" is_command="false">
                <description xml:space="preserve">This message shall cause the
   receiving service to reply to the requestor with a Report
   Analog Video Sensor Configuration message.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="SensorIdList" optional="false">
```

```
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="SensorIDQueryRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryAnalogVideoSensorCapabilities" message_id="2810" is_command="false">
                <description xml:space="preserve">This message shall cause the
   receiving service to reply to the requestor with a Report
   Analog Video Sensor Capabilities message.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="SensorIdList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="SensorIDQueryRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
        </input_set>
        <output_set>
            <message_def name="ReportAnalogVideoSensorConfiguration" message_id="4811" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="AnalogVideoSensorConfigurationList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="AnalogVideoSensorConfigurationRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is not
a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="AnalogFormat" optional="false" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NTSC-M"/>
                                    <value_enum enum_index="1" enum_const="NTSC-J"/>
                                    <value_enum enum_index="2" enum_const="PAL-N"/>
                                    <value_enum enum_index="3" enum_const="PAL-M"/>
                                    <value_enum enum_index="4" enum_const="SECAM-L"/>
                                    <value_enum enum_index="5" enum_const="SECAM-B/G"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </body>
```

```
            <footer name="Footer"/>
        </message_def>
        <message_def name="ReportAnalogVideoSensorCapabilities" message_id="4810" is_command="false">
            <description xml:space="preserve">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.</description>
            <header name="AppHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="AnalogVideoSensorCapabilitiesList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <record name="AnalogVideoSensorCapabilitiesRec" optional="false">
                        <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is not
a valid ID in this message." field_type="unsigned short integer" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                            </value_set>
                        </fixed_field>
                        <bit_field name="SupportedAnalogFormats" optional="false" field_type_unsigned="unsigned
byte">
                            <sub_field name="NTSCM">
                                <bit_range from_index="0" to_index="0"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="NTSCJ">
                                <bit_range from_index="1" to_index="1"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="PALN">
                                <bit_range from_index="2" to_index="2"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="PALM">
                                <bit_range from_index="3" to_index="3"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="SECAML">
                                <bit_range from_index="4" to_index="4"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="SECAMBG">
                                <bit_range from_index="5" to_index="5"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                        </bit_field>
                    </record>
                </list>
            </body>
            <footer name="Footer"/>
        </message_def>
    </output_set>
</message_set>
<internal_events_set/>
<protocol_behavior is_stateless="false">
    <start state_machine_name="visualSensor.accessControl.events.transport.ReceiveFSM"
state_name="Receiving.Ready.NotControlled"/>
    <start state_machine_name="visualSensor.accessControl.events.transport.SendFSM" state_name="Sending"/>
    <state_machine name="visualSensor.accessControl.events.transport.ReceiveFSM">
```

```
            <state name="Receiving" initial_state="Ready">
                <state name="Ready" initial_state="NotControlled">
                    <state name="NotControlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryAnalogVideoSensorCapabilities" value="msg"
interpretation="enveloped QueryAnalogVideoSensorCapabilities message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportAnalogVideoSensorCapabilities" interpretation="Send a
ReportAnalogVideoSensorCapabilities message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryAnalogVideoSensorConfiguration" value="msg"
interpretation="enveloped QueryAnalogVideoSensorConfiguration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportAnalogVideoSensorConfiguration" interpretation="Send a
ReportAnalogVideoSensorConfiguration message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                        </state>
                    <state name="Controlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="SetAnalogVideoSensorConfiguration" value="msg"
interpretation="enveloped SetAnalogVideoSensorConfiguration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <guard condition="accessControl.isControllingClient( transportData )"
interpretation="True if the message that triggered          the transition is received from the client that is in
control of this service."/>
                                <internal/>
                                <action name="sendConfirmSensorConfiguration" interpretation="Send
sendConfirmSensorConfiguration message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                                <action name="updateAnalogVideoSensorConfiguration" interpretation="Update the sensor
user controllable           configuration parameters according to the ones specified.">
                                    <argument value="msg"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryAnalogVideoSensorCapabilities" value="msg"
interpretation="enveloped QueryAnalogVideoSensorCapabilities message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportAnalogVideoSensorCapabilities" interpretation="Send a
ReportAnalogVideoSensorCapabilities message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryAnalogVideoSensorConfiguration" value="msg"
interpretation="enveloped QueryAnalogVideoSensorConfiguration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportAnalogVideoSensorConfiguration" interpretation="Send a
ReportAnalogVideoSensorConfiguration message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
```

```
                            </action>
                        </transition>
                    </default_state>
                                        </state>
                <default_state>
                                        </default_state>
                            </state>
                        </state>
                            </state_machine>
        <state_machine name="visualSensor.accessControl.events.transport.SendFSM">
            <state name="Sending">
                        </state>
                            </state_machine>
    </protocol_behavior>
</service_def>
```

A.2 DIGITALAUDIOOUTPUT

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="DigitalAudioOutput" id="urn:jaus:jss:environmentSensing:DigitalAudioOutput" version="2.0"
xmlns="urn:jaus:jsidl:1.1" xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">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.</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost, or reordered.</assumptions>
    <references>
        <inherits_from name="accessControl" id="urn:jaus:jss:core:AccessControl" version="1.1"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="SetDigitalAudioOutputSource" message_id="0818" is_command="true">
                <description xml:space="preserve">This message is used to specify the stream source to play.  Note that
the specified stream must not require a DNS lookup to resolve.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="DigitalAudioOutputSeq" optional="false">
                        <record name="RequestIdRec" optional="false">
                            <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                        <list name="DigitalAudioOutputSourceList" optional="false">
                            <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                            <record name="DigitalAudioOutputSourceRec" optional="false">
                                <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned
short integer" interpretation="ID of the sensor. Zero is not a valid ID in this message.">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit_type="inclusive" lower_limit="1"
                                            upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="StreamURL" optional="false" interpretation="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.">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned byte"/>
                                </variable_length_string>
                                <fixed_field name="Repeat" optional="false" interpretation="Specifies behavior for RTSP
streams that have a finite play time." field_type="unsigned byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="PlayOnce"/>
                                        <value_enum enum_index="1" enum_const="RepeatUntilNewStreamSpecified"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="Volume" optional="false" interpretation="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." field_type="unsigned short integer" field_units="percent">
                                    <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                                </fixed_field>
                            </record>
                        </list>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
```

```
            <message_def name="QueryDigitalAudioOutputSource" message_id="2818" is_command="false">
                <description xml:space="preserve">This message is used to query the stream source currently being
played for each sensor.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="QueryDigitalAudioRec" optional="false">
                        <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned short
integer" interpretation="ID of the sensor being queried. Zero is used to query all sensors associated with this
service.">
                          <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="AllSensors"/>
                            <value_range lower_limit_type="inclusive" lower_limit="1"
                                upper_limit="65535" upper_limit_type="inclusive"
                                interpretation="Specific ID to be queried"/>
                          </value_set>
                        </fixed_field>
                    </record>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryDigitalAudioOutputCapabilities" message_id="2819" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="QueryDigitalAudioRec" optional="false">
                        <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned short
integer" interpretation="ID of the sensor being queried. Zero is used to query all sensors associated with this
service.">
                          <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="AllSensors"/>
                            <value_range lower_limit_type="inclusive" lower_limit="1"
                                upper_limit="65535" upper_limit_type="inclusive"
                                interpretation="Specific ID to be queried"/>
                          </value_set>
                        </fixed_field>
                    </record>
                </body>
                <footer name="Footer"/>
            </message_def>
        </input_set>
        <output_set>
            <message_def name="ConfirmSensorConfiguration" message_id="0801" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="ConfirmSensorConfigurationSequence" optional="false">
                        <record name="RequestIdRec" optional="false">
                            <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                        <list name="ConfirmSensorList" optional="false">
                            <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                            <variant name="ConfirmSensorConfigurationVariant" optional="false">
                                <vtag_field min_count="0" max_count="9" field_type_unsigned="unsigned byte"/>
                                <record name="SensorIdRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
```

```
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                </record>
                                <record name="RangeSensorErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="RangeSensorErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid Horizontal Field of View"/>
                                            <value_enum enum_index="2" enum_const="Invalid Vertical Field of View"/>
                                            <value_enum enum_index="3" enum_const="Invalid Update Rate"/>
                                            <value_enum enum_index="4" enum_const="Invalid Sensor Range"/>
                                            <value_enum enum_index="5" enum_const="Invalid Sensor State"/>
                                            <value_enum enum_index="6" enum_const="Multiple Invalid Parameters"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="VisualSensorErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="VisualSensorErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid Sensor State"/>
                                            <value_enum enum_index="2" enum_const="Invalid Zoom Mode"/>
                                            <value_enum enum_index="3" enum_const="Invalid Zoom Value"/>
                                            <value_enum enum_index="4" enum_const="Invalid Focus Mode"/>
                                            <value_enum enum_index="5" enum_const="Invalid Focus Value"/>
                                            <value_enum enum_index="6" enum_const="Invalid White Balance"/>
                                            <value_enum enum_index="7" enum_const="Invalid Imaging Mode"/>
                                            <value_enum enum_index="8" enum_const="Invalid Exposure Mode"/>
                                            <value_enum enum_index="9" enum_const="Invalid Metering Mode"/>
                                            <value_enum enum_index="10" enum_const="Invalid Shutter Speed"/>
                                            <value_enum enum_index="11" enum_const="Invalid Aperture Value"/>
                                            <value_enum enum_index="12" enum_const="Invalid Light Sensitivity"/>
                                            <value_enum enum_index="13" enum_const="Invalid Image Stabilization"/>
                                            <value_enum enum_index="14" enum_const="Invalid Horizontal FOV"/>
                                            <value_enum enum_index="15" enum_const="Invalid Vertical FOV"/>
                                            <value_enum enum_index="16" enum_const="Multiple Invalid Parameters"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="DigitalVideoSensorErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="DigitalVideoErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
```

```
                                        <value_enum enum_index="1" enum_const="Invalid Minimum Bit Rate"/>
                                        <value_enum enum_index="2" enum_const="Invalid Maximum Bit Rate"/>
                                        <value_enum enum_index="3" enum_const="Requested Frame Rate Too Low"/>
                                        <value_enum enum_index="4" enum_const="Requested Frame Rate Too High"/>
                                        <value_enum enum_index="5" enum_const="Invalid Frame Size"/>
                                        <value_enum enum_index="6" enum_const="Invalid Format"/>
                                        <value_enum enum_index="7" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="AnalogVideoSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="AnalogVideoErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Format"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="StillImageSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="StillImageErrorCode" optional="false" field_type="unsigned byte"
field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Frame Size"/>
                                        <value_enum enum_index="2" enum_const="Invalid Format"/>
                                        <value_enum enum_index="3" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="H264VideoEncodingErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="H264VideoEncodingErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Profile"/>
                                        <value_enum enum_index="2" enum_const="Invalid Preset"/>
                                        <value_enum enum_index="3" enum_const="Invalid GroupOfPictures"/>
                                        <value_enum enum_index="4" enum_const="Invalid GDR"/>
                                        <value_enum enum_index="5" enum_const="Invalid RegionOfInterest"/>
                                        <value_enum enum_index="6" enum_const="Invalid IntraMacroRefresh"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
```

```
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="DigitalAudioSensorErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="DigitalAudioSensorErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid BitRate"/>
                                            <value_enum enum_index="2" enum_const="Invalid Format"/>
                                            <value_enum enum_index="3" enum_const="Invalid SampleRate"/>
                                            <value_enum enum_index="4" enum_const="Invalid BitDepth"/>
                                            <value_enum enum_index="5" enum_const="Invalid EncodingQuality"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="DigitalAudioOutputErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="DigitalAudioOutputErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Stream Not Found"/>
                                            <value_enum enum_index="2" enum_const="Stream Not Supported"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="VideoIlluminatorErrorRec" optional="false">
                                    <fixed_field name="IlluminatorID" optional="false" field_type="unsigned short
integer" field_units="one"/>
                                    <fixed_field name="VideoIlluminatorErrorRecCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Unsupported Mode"/>
                                            <value_enum enum_index="2" enum_const="Unsupported Beam Width"/>
                                            <value_enum enum_index="3" enum_const="Unsupported Beam Height"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                            </variant>
                        </list>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="ReportDigitalAudioOutputSource" message_id="4818" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
```

```
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="DigitalAudioOutputSourceList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="DigitalAudioOutputSourceRec" optional="false">
                            <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned short
integer" interpretation="ID of the sensor. Zero is not a valid ID in this message.">
                                <value_set offset_to_lower_limit="false">
                                  <value_range lower_limit_type="inclusive" lower_limit="1"
                                     upper_limit="65535" upper_limit_type="inclusive"/>
                                </value_set>
                            </fixed_field>
                            <variable_length_string name="StreamURL" optional="false" interpretation="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.">
                                <count_field min_count="0" max_count="255" field_type_unsigned="unsigned byte"/>
                            </variable_length_string>
                            <fixed_field name="Repeat" optional="false" interpretation="Specifies behavior for RTSP
streams that have a finite play time." field_type="unsigned byte" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="PlayOnce"/>
                                    <value_enum enum_index="1" enum_const="RepeatUntilNewStreamSpecified"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="Volume" optional="false" interpretation="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." field_type="unsigned short integer" field_units="percent">
                                <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                            </fixed_field>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="ReportDigitalAudioOutputCapabilities" message_id="4819" is_command="false">
                <description xml:space="preserve">This message is used to report full set of capabilities (supported
performance levels) for one or more sensors.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="DigitalAudioOutputCapabilitiesList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="DigitalAudioOutputCapabilitiesRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned byte"/>
                            <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned short
integer" interpretation="ID of the sensor. Zero is not a valid ID in this message.">
                                <value_set offset_to_lower_limit="false">
                                  <value_range lower_limit_type="inclusive" lower_limit="1"
                                     upper_limit="65535" upper_limit_type="inclusive"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="MinimumAllowedBitRate" optional="true" interpretation="kilobits per
second" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="MaximumAllowedBitRate" optional="true" interpretation="kilobits per
second" field_type="unsigned short integer" field_units="one"/>
                            <bit_field name="SupportedDigitalFormats" optional="true" field_type_unsigned="unsigned
integer">
                                <sub_field name="LPCM_PCM">
                                    <bit_range from_index="0" to_index="0" interpretation="LPCM_PCM"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                        <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="AIFF">
                                    <bit_range from_index="1" to_index="1" interpretation="AIFF"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                        <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="WAV">
                                    <bit_range from_index="2" to_index="2" interpretation="WAV"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
```

```
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="ALAC">
        <bit_range from_index="3" to_index="3" interpretation="ALAC"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="FLAC">
        <bit_range from_index="4" to_index="4" interpretation="FLAC"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="RealAudio">
        <bit_range from_index="5" to_index="5" interpretation="RealAudio"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="WMA9_Lossless">
        <bit_range from_index="6" to_index="6" interpretation="WMA9_Lossless"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="TrueAudio">
        <bit_range from_index="7" to_index="7" interpretation="TrueAudio"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="DolbyDigital">
        <bit_range from_index="8" to_index="8" interpretation="DolbyDigital"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="DTS">
        <bit_range from_index="9" to_index="9" interpretation="DTS"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="MP2">
        <bit_range from_index="10" to_index="10" interpretation="MP2"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="MP3">
        <bit_range from_index="11" to_index="11" interpretation="MP3"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="AAC_MPEG2">
        <bit_range from_index="12" to_index="12" interpretation="AAC_MPEG2"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="AAC_MPEG4">
        <bit_range from_index="13" to_index="13" interpretation="AAC_MPEG4"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
            <value_enum enum_index="1" enum_const="SUPPORTED"/>
        </value_set>
    </sub_field>
    <sub_field name="VORBIS">
```

```
                                <bit_range from_index="14" to_index="14" interpretation="VORBIS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="WMA">
                                <bit_range from_index="15" to_index="15" interpretation="WMA"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Speex">
                                <bit_range from_index="16" to_index="16" interpretation="Speex"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                        </bit_field>
                        <fixed_field name="MinSampleRate" optional="true" interpretation="Samples per second"
field_type="unsigned integer" field_units="one"/>
                        <fixed_field name="MaxSampleRate" optional="true" interpretation="Samples per second"
field_type="unsigned integer" field_units="one"/>
                        <bit_field name="SupportedBitDepths" optional="true" field_type_unsigned="unsigned short
integer">
                            <sub_field name="EIGHT_BITS">
                                <bit_range from_index="0" to_index="0" interpretation="EIGHT_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="TEN_BITS">
                                <bit_range from_index="1" to_index="1" interpretation="TEN_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="TWELVE_BITS">
                                <bit_range from_index="2" to_index="2" interpretation="TWELVE_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="FOURTEEN_BITS">
                                <bit_range from_index="3" to_index="3" interpretation="FOURTEEN_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="SIXTEEN_BITS">
                                <bit_range from_index="4" to_index="4" interpretation="SIXTEEN_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="TWENTY_BITS">
                                <bit_range from_index="5" to_index="5" interpretation="TWENTY_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="TWENTY_FOUR_BITS">
                                <bit_range from_index="6" to_index="6" interpretation="TWENTY_FOUR_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="THIRTY_TWO_BITS">
                                <bit_range from_index="7" to_index="7" interpretation="THIRTY_TWO_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
```

```
                                    </value_set>
                                </sub_field>
                            </bit_field>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
        </output_set>
    </message_set>
    <internal_events_set/>
    <protocol_behavior is_stateless="false">
        <start state_machine_name="accessControl.events.transport.ReceiveFSM"
state_name="Receiving.Ready.NotControlled"/>
        <start state_machine_name="accessControl.events.transport.SendFSM" state_name="Sending"/>
        <state_machine name="accessControl.events.transport.ReceiveFSM">
            <state name="Receiving" initial_state="Ready">
                <state name="Ready" initial_state="NotControlled">
                    <state name="NotControlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalAudioOutputSource" value="msg" interpretation="enveloped
Query Digital Audio OutputSource message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <internal/>
                                <action name="sendReportDigitalAudioOutputSource" interpretation="Send a
ReportDigitalAudioOutputSource message to querying client">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalAudioOutputCapabilities" value="msg"
interpretation="enveloped Query Digital Audio Output Capabilities message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <internal/>
                                <action name="sendReportDigitalAudioOutputCapabilities" interpretation="Send a
ReportDigitalAudioOutputCapabilities message to querying client">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                                </state>
                    <state name="Controlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalAudioOutputSource" value="msg" interpretation="enveloped
Query Digital Audio OutputSource message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <internal/>
                                <action name="sendReportDigitalAudioOutputSource" interpretation="Send a
ReportDigitalAudioOutputSource message to querying client">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalAudioOutputCapabilities" value="msg"
interpretation="enveloped Query Digital Audio Output Capabilities message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <internal/>
                                <action name="sendReportDigitalAudioOutputCapabilities" interpretation="Send a
ReportDigitalAudioOutputCapabilities message to querying client">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
```

```
                                <parameter type="SetDigitalAudioOutputSource" value="msg" interpretation="enveloped Set
Digital Audio Output Source message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <guard condition="isControllingClient( transportData )" interpretation="True if the
command message was received from the client currently controlling this component."/>
                                <internal/>
                                <action name="playStream" interpretation="Begin playback for any sensor IDs with valid
and supported streams specified in the message with the given repeat behavior and volume.">
                                    <argument value="msg"/>
                                </action>
                                <action name="sendConfirmSensorConfiguration" interpretation="Send
sendConfirmSensorConfiguration message with confirmation or error code for each specified sensor ID">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                            </state>
                    <default_state>
                                            </default_state>
                                    </state>
                            </state>
                                </state_machine>
        <state_machine name="accessControl.events.transport.SendFSM">
            <state name="Sending">
                            </state>
                                </state_machine>
    </protocol_behavior>
</service_def>
```

A.3 DIGITALAUDIOSENSOR

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="DigitalAudioSensor" id="urn:jaus:jss:environmentSensing:DigitalAudioSensor" version="2.0"
xmlns="urn:jaus:jsidl:1.1" xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">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.</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost, or reordered.</assumptions>
    <references>
        <inherits_from name="AccessControl" id="urn:jaus:jss:core:AccessControl" version="1.1"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="SetDigitalAudioSensorConfiguration" message_id="080F" is_command="true">
                <description xml:space="preserve">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.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="DigitalAudioSensorConfigurationSeq" optional="false">
                        <record name="RequestIdRec" optional="false">
                            <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                        <list name="DigitalAudioSensorConfigurationList" optional="false">
                            <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                            <record name="DigitalAudioSensorConfigurationRec" optional="false">
                                <presence_vector field_type_unsigned="unsigned byte"/>
                                <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned
short integer" interpretation="ID of the sensor. Zero is not a valid ID in this message.">
                                    <value_set offset_to_lower_limit="false">
                                       <value_range lower_limit_type="inclusive" lower_limit="1"
                                        upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="Sensitivity" optional="true" interpretation="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." field_type="unsigned short integer" field_units="one">
                                    <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                                </fixed_field>
```

```
                                <fixed_field name="MinimumBitRate" optional="true" interpretation="kilobits per second.
Different min and max bitrates may result in a Variable Bit Rate (VBR) stream, if supported." field_type="unsigned
short integer" field_units="one"/>
                                <fixed_field name="MaximumBitRate" optional="true" interpretation="kilobits per second.
Different min and max bitrates may result in a Variable Bit Rate (VBR) stream, if supported." field_type="unsigned
short integer" field_units="one"/>
                                <fixed_field name="DigitalFormat" optional="true" field_type="unsigned byte"
field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="LPCM_PCM"/>
                                        <value_enum enum_index="1" enum_const="AIFF"/>
                                        <value_enum enum_index="2" enum_const="WAV"/>
                                        <value_enum enum_index="3" enum_const="ALAC"/>
                                        <value_enum enum_index="4" enum_const="FLAC"/>
                                        <value_enum enum_index="5" enum_const="RealAudio"/>
                                        <value_enum enum_index="6" enum_const="WMA9_LOSSLESS"/>
                                        <value_enum enum_index="7" enum_const="TrueAudio"/>
                                        <value_enum enum_index="8" enum_const="DolbyDigital"/>
                                        <value_enum enum_index="9" enum_const="DTS"/>
                                        <value_enum enum_index="10" enum_const="MP2"/>
                                        <value_enum enum_index="11" enum_const="MP3"/>
                                        <value_enum enum_index="12" enum_const="AAC_MPEG2"/>
                                        <value_enum enum_index="13" enum_const="AAC_MPEG4"/>
                                        <value_enum enum_index="14" enum_const="VORBIS"/>
                                        <value_enum enum_index="15" enum_const="WMA"/>
                                        <value_enum enum_index="16" enum_const="Speex"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="SampleRate" optional="true" interpretation="Samples per second"
field_type="unsigned integer" field_units="one"/>
                                <fixed_field name="BitDepth" optional="true" field_type="unsigned byte"
field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="EIGHT_BITS"/>
                                        <value_enum enum_index="1" enum_const="TEN_BITS"/>
                                        <value_enum enum_index="2" enum_const="TWELVE_BITS"/>
                                        <value_enum enum_index="3" enum_const="FOURTEEN_BITS"/>
                                        <value_enum enum_index="4" enum_const="SIXTEEN_BITS"/>
                                        <value_enum enum_index="5" enum_const="TWENTY_BITS"/>
                                        <value_enum enum_index="6" enum_const="TWENTY_FOUR_BITS"/>
                                        <value_enum enum_index="7" enum_const="THIRTY_TWO_BITS"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="EncodingQuality" optional="true" field_type="unsigned byte"
field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Best"/>
                                        <value_enum enum_index="1" enum_const="Better"/>
                                        <value_enum enum_index="2" enum_const="Good"/>
                                        <value_enum enum_index="3" enum_const="Average"/>
                                        <value_enum enum_index="4" enum_const="Adequate"/>
                                        <value_enum enum_index="5" enum_const="Poor"/>
                                        <value_enum enum_index="6" enum_const="Worst"/>
                                    </value_set>
                                </fixed_field>
                            </record>
                        </list>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryDigitalAudioSensorCapabilities" message_id="2817" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="QueryDigitalAudioSensorRec" optional="false">
                        <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned short
integer" interpretation="ID of the sensor being queried. Zero is used to query all sensors associated with this
service.">
                          <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="AllSensors"/>
                            <value_range lower_limit_type="inclusive" lower_limit="1"
                                upper_limit="65535" upper_limit_type="inclusive"
                                interpretation="Specific ID to be queried"/>
```

```
                        </value_set>
                    </fixed_field>
                </record>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="QueryDigitalAudioSensorConfiguration" message_id="280F" is_command="false">
            <description xml:space="preserve">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.</description>
            <header name="JAUSApplicationLayerHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <record name="QueryDigitalAudioSensorRec" optional="false">
                    <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned short
integer" interpretation="ID of the sensor being queried. Zero is used to query all sensors associated with this
service.">
                        <value_set offset_to_lower_limit="false">
                          <value_enum enum_index="0" enum_const="AllSensors"/>
                          <value_range lower_limit_type="inclusive" lower_limit="1"
                              upper_limit="65535" upper_limit_type="inclusive"
                              interpretation="Specific ID to be queried"/>
                        </value_set>
                    </fixed_field>
                </record>
            </body>
            <footer name="Footer"/>
        </message_def>
    </input_set>
    <output_set>
        <message_def name="ConfirmSensorConfiguration" message_id="0801" is_command="false">
            <description xml:space="preserve">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.</description>
            <header name="AppHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <sequence name="ConfirmSensorConfigurationSequence" optional="false">
                    <record name="RequestIdRec" optional="false">
                        <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                    </record>
                    <list name="ConfirmSensorList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <variant name="ConfirmSensorConfigurationVariant" optional="false">
                            <vtag_field min_count="0" max_count="9" field_type_unsigned="unsigned byte"/>
                            <record name="SensorIdRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                            </record>
                            <record name="RangeSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="RangeSensorErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
```

```
                                        <value_enum enum_index="1" enum_const="Invalid Horizontal Field of View"/>
                                        <value_enum enum_index="2" enum_const="Invalid Vertical Field of View"/>
                                        <value_enum enum_index="3" enum_const="Invalid Update Rate"/>
                                        <value_enum enum_index="4" enum_const="Invalid Sensor Range"/>
                                        <value_enum enum_index="5" enum_const="Invalid Sensor State"/>
                                        <value_enum enum_index="6" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="VisualSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="VisualSensorErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Sensor State"/>
                                        <value_enum enum_index="2" enum_const="Invalid Zoom Mode"/>
                                        <value_enum enum_index="3" enum_const="Invalid Zoom Value"/>
                                        <value_enum enum_index="4" enum_const="Invalid Focus Mode"/>
                                        <value_enum enum_index="5" enum_const="Invalid Focus Value"/>
                                        <value_enum enum_index="6" enum_const="Invalid White Balance"/>
                                        <value_enum enum_index="7" enum_const="Invalid Imaging Mode"/>
                                        <value_enum enum_index="8" enum_const="Invalid Exposure Mode"/>
                                        <value_enum enum_index="9" enum_const="Invalid Metering Mode"/>
                                        <value_enum enum_index="10" enum_const="Invalid Shutter Speed"/>
                                        <value_enum enum_index="11" enum_const="Invalid Aperture Value"/>
                                        <value_enum enum_index="12" enum_const="Invalid Light Sensitivity"/>
                                        <value_enum enum_index="13" enum_const="Invalid Image Stabilization"/>
                                        <value_enum enum_index="14" enum_const="Invalid Horizontal FOV"/>
                                        <value_enum enum_index="15" enum_const="Invalid Vertical FOV"/>
                                        <value_enum enum_index="16" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="DigitalVideoSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="DigitalVideoErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Minimum Bit Rate"/>
                                        <value_enum enum_index="2" enum_const="Invalid Maximum Bit Rate"/>
                                        <value_enum enum_index="3" enum_const="Requested Frame Rate Too Low"/>
                                        <value_enum enum_index="4" enum_const="Requested Frame Rate Too High"/>
                                        <value_enum enum_index="5" enum_const="Invalid Frame Size"/>
                                        <value_enum enum_index="6" enum_const="Invalid Format"/>
                                        <value_enum enum_index="7" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="AnalogVideoSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
```

```
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="AnalogVideoErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Format"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="StillImageSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="StillImageErrorCode" optional="false" field_type="unsigned byte"
field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Frame Size"/>
                                        <value_enum enum_index="2" enum_const="Invalid Format"/>
                                        <value_enum enum_index="3" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="H264VideoEncodingErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="H264VideoEncodingErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Profile"/>
                                        <value_enum enum_index="2" enum_const="Invalid Preset"/>
                                        <value_enum enum_index="3" enum_const="Invalid GroupOfPictures"/>
                                        <value_enum enum_index="4" enum_const="Invalid GDR"/>
                                        <value_enum enum_index="5" enum_const="Invalid RegionOfInterest"/>
                                        <value_enum enum_index="6" enum_const="Invalid IntraMacroRefresh"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="DigitalAudioSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="DigitalAudioSensorErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid BitRate"/>
                                        <value_enum enum_index="2" enum_const="Invalid Format"/>
                                        <value_enum enum_index="3" enum_const="Invalid SampleRate"/>
```

```
                                        <value_enum enum_index="4" enum_const="Invalid BitDepth"/>
                                        <value_enum enum_index="5" enum_const="Invalid EncodingQuality"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="DigitalAudioOutputErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="DigitalAudioOutputErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Stream Not Found"/>
                                        <value_enum enum_index="2" enum_const="Stream Not Supported"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="VideoIlluminatorErrorRec" optional="false">
                                <fixed_field name="IlluminatorID" optional="false" field_type="unsigned short
integer" field_units="one"/>
                                <fixed_field name="VideoIlluminatorErrorRecCode" optional="false"
field_type="unsigned byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Unsupported Mode"/>
                                        <value_enum enum_index="2" enum_const="Unsupported Beam Width"/>
                                        <value_enum enum_index="3" enum_const="Unsupported Beam Height"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                        </variant>
                    </list>
                </sequence>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="ReportDigitalAudioSensorCapabilities" message_id="4817" is_command="false">
            <description xml:space="preserve">This message is used to report full set of capabilities (supported
performance levels) for one or more sensors.</description>
            <header name="JAUSApplicationLayerHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="DigitalAudioSensorCapabilitiesList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <record name="DigitalAudioSensorCapabilitiesRec" optional="false">
                        <presence_vector field_type_unsigned="unsigned byte"/>
                        <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned short
integer" interpretation="ID of the sensor. Zero is not a valid ID in this message.">
                            <value_set offset_to_lower_limit="false">
                              <value_range lower_limit_type="inclusive" lower_limit="1"
                                 upper_limit="65535" upper_limit_type="inclusive"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="MinimumAllowedBitRate" optional="true" interpretation="kilobits per
second" field_type="unsigned short integer" field_units="one"/>
                        <fixed_field name="MaximumAllowedBitRate" optional="true" interpretation="kilobits per
second" field_type="unsigned short integer" field_units="one"/>
```

```
                    <bit_field name="SupportedDigitalFormats" optional="true" field_type_unsigned="unsigned
integer">
                        <sub_field name="LPCM_PCM">
                            <bit_range from_index="0" to_index="0" interpretation="LPCM_PCM"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="AIFF">
                            <bit_range from_index="1" to_index="1" interpretation="AIFF"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="WAV">
                            <bit_range from_index="2" to_index="2" interpretation="WAV"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="ALAC">
                            <bit_range from_index="3" to_index="3" interpretation="ALAC"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="FLAC">
                            <bit_range from_index="4" to_index="4" interpretation="FLAC"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="RealAudio">
                            <bit_range from_index="5" to_index="5" interpretation="RealAudio"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="WMA9_Lossless">
                            <bit_range from_index="6" to_index="6" interpretation="WMA9_Lossless"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="TrueAudio">
                            <bit_range from_index="7" to_index="7" interpretation="TrueAudio"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="DolbyDigital">
                            <bit_range from_index="8" to_index="8" interpretation="DolbyDigital"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="DTS">
                            <bit_range from_index="9" to_index="9" interpretation="DTS"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="MP2">
                            <bit_range from_index="10" to_index="10" interpretation="MP2"/>
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                <value_enum enum_index="1" enum_const="SUPPORTED"/>
                            </value_set>
                        </sub_field>
                        <sub_field name="MP3">
                            <bit_range from_index="11" to_index="11" interpretation="MP3"/>
```

```
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="AAC_MPEG2">
                                <bit_range from_index="12" to_index="12" interpretation="AAC_MPEG2"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="AAC_MPEG4">
                                <bit_range from_index="13" to_index="13" interpretation="AAC_MPEG4"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="VORBIS">
                                <bit_range from_index="14" to_index="14" interpretation="VORBIS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="WMA">
                                <bit_range from_index="15" to_index="15" interpretation="WMA"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Speex">
                                <bit_range from_index="16" to_index="16" interpretation="Speex"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                        </bit_field>
                        <fixed_field name="MinSampleRate" optional="true" interpretation="Samples per second"
field_type="unsigned integer" field_units="one"/>
                        <fixed_field name="MaxSampleRate" optional="true" interpretation="Samples per second"
field_type="unsigned integer" field_units="one"/>
                        <bit_field name="SupportedBitDepths" optional="true" field_type_unsigned="unsigned short
integer">
                            <sub_field name="EIGHT_BITS">
                                <bit_range from_index="0" to_index="0" interpretation="EIGHT_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="TEN_BITS">
                                <bit_range from_index="1" to_index="1" interpretation="TEN_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="TWELVE_BITS">
                                <bit_range from_index="2" to_index="2" interpretation="TWELVE_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="FOURTEEN_BITS">
                                <bit_range from_index="3" to_index="3" interpretation="FOURTEEN_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="SIXTEEN_BITS">
                                <bit_range from_index="4" to_index="4" interpretation="SIXTEEN_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
```

```
                            </sub_field>
                            <sub_field name="TWENTY_BITS">
                                <bit_range from_index="5" to_index="5" interpretation="TWENTY_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="TWENTY_FOUR_BITS">
                                <bit_range from_index="6" to_index="6" interpretation="TWENTY_FOUR_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="THIRTY_TWO_BITS">
                                <bit_range from_index="7" to_index="7" interpretation="THIRTY_TWO_BITS"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                        </bit_field>
                        <bit_field name="SupportedQualityLevels" optional="true" field_type_unsigned="unsigned
byte">
                            <sub_field name="Best">
                                <bit_range from_index="0" to_index="0" interpretation="Best"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Better">
                                <bit_range from_index="1" to_index="1" interpretation="Better"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Good">
                                <bit_range from_index="2" to_index="2" interpretation="Good"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Average">
                                <bit_range from_index="3" to_index="3" interpretation="Average"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Adequate">
                                <bit_range from_index="4" to_index="4" interpretation="Adequate"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Poor">
                                <bit_range from_index="5" to_index="5" interpretation="Poor"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Worst">
                                <bit_range from_index="6" to_index="6" interpretation="Worst"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NOT_SUPPORTED"/>
                                    <value_enum enum_index="1" enum_const="SUPPORTED"/>
                                </value_set>
                            </sub_field>
                        </bit_field>
                    </record>
                </list>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="ReportDigitalAudioSensorConfiguration" message_id="480F" is_command="false">
```

```
            <description xml:space="preserve">This message is used to report the current configuration for one or
more audio sensors.</description>
            <header name="JAUSApplicationLayerHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="DigitalAudioSensorConfigurationList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <record name="DigitalAudioSensorConfigurationRec" optional="false">
                        <presence_vector field_type_unsigned="unsigned byte"/>
                        <fixed_field name="SensorID" optional="false" field_units="one" field_type="unsigned short
integer" interpretation="ID of the sensor. Zero is not a valid ID in this message.">
                            <value_set offset_to_lower_limit="false">
                              <value_range lower_limit_type="inclusive" lower_limit="1"
                                  upper_limit="65535" upper_limit_type="inclusive"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="Sensitivity" optional="true" interpretation="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." field_type="unsigned short integer" field_units="one">
                            <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="MinimumBitRate" optional="true" interpretation="kilobits per second.
Different min and max bitrates may result in a Variable Bit Rate (VBR) stream, if supported." field_type="unsigned
short integer" field_units="one"/>
                        <fixed_field name="MaximumBitRate" optional="true" interpretation="kilobits per second.
Different min and max bitrates may result in a Variable Bit Rate (VBR) stream, if supported." field_type="unsigned
short integer" field_units="one"/>
                        <fixed_field name="DigitalFormat" optional="true" field_type="unsigned byte"
field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="LPCM_PCM"/>
                                <value_enum enum_index="1" enum_const="AIFF"/>
                                <value_enum enum_index="2" enum_const="WAV"/>
                                <value_enum enum_index="3" enum_const="ALAC"/>
                                <value_enum enum_index="4" enum_const="FLAC"/>
                                <value_enum enum_index="5" enum_const="RealAudio"/>
                                <value_enum enum_index="6" enum_const="WMA9_LOSSLESS"/>
                                <value_enum enum_index="7" enum_const="TrueAudio"/>
                                <value_enum enum_index="8" enum_const="DolbyDigital"/>
                                <value_enum enum_index="9" enum_const="DTS"/>
                                <value_enum enum_index="10" enum_const="MP2"/>
                                <value_enum enum_index="11" enum_const="MP3"/>
                                <value_enum enum_index="12" enum_const="AAC_MPEG2"/>
                                <value_enum enum_index="13" enum_const="AAC_MPEG4"/>
                                <value_enum enum_index="14" enum_const="VORBIS"/>
                                <value_enum enum_index="15" enum_const="WMA"/>
                                <value_enum enum_index="16" enum_const="Speex"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="SampleRate" optional="true" interpretation="Samples per second"
field_type="unsigned integer" field_units="one"/>
                        <fixed_field name="BitDepth" optional="true" field_type="unsigned byte" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="EIGHT_BITS"/>
                                <value_enum enum_index="1" enum_const="TEN_BITS"/>
                                <value_enum enum_index="2" enum_const="TWELVE_BITS"/>
                                <value_enum enum_index="3" enum_const="FOURTEEN_BITS"/>
                                <value_enum enum_index="4" enum_const="SIXTEEN_BITS"/>
                                <value_enum enum_index="5" enum_const="TWENTY_BITS"/>
                                <value_enum enum_index="6" enum_const="TWENTY_FOUR_BITS"/>
                                <value_enum enum_index="7" enum_const="THIRTY_TWO_BITS"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="EncodingQuality" optional="true" field_type="unsigned byte"
field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="Best"/>
                                <value_enum enum_index="1" enum_const="Better"/>
                                <value_enum enum_index="2" enum_const="Good"/>
                                <value_enum enum_index="3" enum_const="Average"/>
                                <value_enum enum_index="4" enum_const="Adequate"/>
                                <value_enum enum_index="5" enum_const="Poor"/>
                                <value_enum enum_index="6" enum_const="Worst"/>
                            </value_set>
                        </fixed_field>
                    </record>
                </list>
```

```
                </body>
                <footer name="Footer"/>
            </message_def>
        </output_set>
    </message_set>
    <internal_events_set/>
    <protocol_behavior is_stateless="false">
        <start state_machine_name="AccessControl.events.transport.ReceiveFSM"
state_name="Receiving.Ready.NotControlled"/>
        <start state_machine_name="AccessControl.events.transport.SendFSM" state_name="Sending"/>
        <state_machine name="AccessControl.events.transport.ReceiveFSM">
            <state name="Receiving" initial_state="Ready">
                <state name="Ready" initial_state="NotControlled">
                    <state name="NotControlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalAudioSensorCapabilities" value="msg"
interpretation="enveloped Query Digital Audio Sensor Capbilities message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <internal/>
                                <action name="sendReportDigitalAudioSensorCapabilities" interpretation="Send a
ReportDigitalAudioSensorCapabilities message to querying client">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalAudioSensorConfiguration" value="msg"
interpretation="enveloped Query Digital Audio Sensor Configuration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <internal/>
                                <action name="sendReportDigitalAudioSensorConfiguration" interpretation="Send a
ReportDigitalAudioSensorConfiguration message to querying client">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                            </state>
                    <state name="Controlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalAudioSensorCapabilities" value="msg"
interpretation="enveloped Query Digital Audio Sensor Capbilities message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <internal/>
                                <action name="sendReportDigitalAudioSensorCapabilities" interpretation="Send a
ReportDigitalAudioSensorCapabilities message to querying client">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalAudioSensorConfiguration" value="msg"
interpretation="enveloped Query Digital Audio Sensor Configuration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <internal/>
                                <action name="sendReportDigitalAudioSensorConfiguration" interpretation="Send a
ReportDigitalAudioSensorConfiguration message to querying client">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="SetDigitalAudioSensorConfiguration" value="msg"
interpretation="enveloped Set Digital Audio Sensor Configuration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                                <guard condition="isControllingClient( transportData )" interpretation="True if the
```

```
command message was received from the client currently controlling this component."/>
                                <internal/>
                                <action name="setConfiguration" interpretation="Update the settings for any sensor IDs
with valid and supported configurations specified in the message">
                                    <argument value="msg"/>
                                </action>
                                <action name="sendConfirmSensorConfiguration" interpretation="Send
sendConfirmSensorConfiguration message with confirmation or error code for each specified sensor ID">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                        </state>
                    <default_state>
                                        </default_state>
                                    </state>
                            </state>
                                </state_machine>
        <state_machine name="AccessControl.events.transport.SendFSM">
            <state name="Sending">
                            </state>
                                </state_machine>
    </protocol_behavior>
</service_def>
```

A.4 DIGITALRESOURCEDISCOVERY

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="DigitalResourceDiscovery" id="urn:jaus:jss:environmentSensing:DigitalResourceDiscovery"
version="2.0" xmlns="urn:jaus:jsidl:1.1" xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">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).</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost, or reordered.</assumptions>
    <references>
        <inherits_from name="Events" id="urn:jaus:jss:core:Events" version="1.1"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="QueryDigitalResourceEndpoint" message_id="2816" is_command="false">
                <description xml:space="preserve">Queries for a list of known digital resource endpoints.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body"/>
                <footer name="Footer"/>
            </message_def>
            <message_def name="RegisterDigitalResourceEndpoint" message_id="0808" is_command="true">
                <description xml:space="preserve">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.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="RegisterDigitalResourceSeq" optional="false">
                        <record name="RequestIDRec" optional="false">
                            <fixed_field name="RequestID" optional="false" interpretation="Client provided ID to link
the response to the request" field_type="unsigned byte" field_units="one"/>
                        </record>
                        <record name="DigitalResourceEndpointRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned byte"/>
                            <fixed_field name="ServerType" optional="false" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="RTSP"/>
                                    <value_enum enum_index="1" enum_const="MPEG2_TS"/>
```

```
                                    <value_enum enum_index="2" enum_const="FTP"/>
                                    <value_enum enum_index="3" enum_const="SFTP"/>
                                    <value_enum enum_index="4" enum_const="FTP_over_SSH"/>
                                    <value_enum enum_index="5" enum_const="HTTP"/>
                                    <value_enum enum_index="6" enum_const="HTTPS"/>
                                    <value_enum enum_index="7" enum_const="SCP"/>
                                    <value_enum enum_index="8" enum_const="CCSI" interpretation="where URL is of the
form 'ip address:port number' (no quotes)"/>
                                    <value_range lower_limit="9" lower_limit_type="inclusive" upper_limit="200"
upper_limit_type="inclusive" interpretation="Reserved for future use"/>
                                    <value_range lower_limit="201" lower_limit_type="inclusive" upper_limit="255"
upper_limit_type="inclusive" interpretation="Reserved for program or implementation specific use"/>
                                </value_set>
                            </fixed_field>
                            <variable_length_string name="ServerURL" optional="false" interpretation="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.">
                                <count_field min_count="0" max_count="255" field_type_unsigned="unsigned byte"/>
                            </variable_length_string>
                            <bit_field name="JAUS_ID" optional="true" field_type_unsigned="unsigned integer">
                                <sub_field name="ComponentID">
                                    <bit_range from_index="0" to_index="7" interpretation="ComponentID"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="0" lower_limit_type="inclusive" upper_limit="255"
upper_limit_type="inclusive"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="NodeID">
                                    <bit_range from_index="8" to_index="15" interpretation="NodeID"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="0" lower_limit_type="inclusive" upper_limit="255"
upper_limit_type="inclusive"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="SubsystemID">
                                    <bit_range from_index="16" to_index="31" interpretation="SubsystemID"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="0" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                    </value_set>
                                </sub_field>
                            </bit_field>
                            <fixed_field name="ResourceID" optional="true" interpretation="The ID used by the
configuration and control service to identify this source. This is the SensorID for visual sensors."
field_type="unsigned short integer" field_units="one"/>
                        </record>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="RemoveDigitalResourceEndpoint" message_id="0809" is_command="true">
                <description xml:space="preserve">Remove a previously registered digital resource transfer server from
the service.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="RemoveDigitalResourceEndpointRec" optional="false">
                        <fixed_field name="ID" optional="false" interpretation="Unique ID that was returned in the
confirm message" field_type="unsigned byte" field_units="one"/>
                        <fixed_field name="RequestID" optional="false" interpretation="Client provided ID to link the
response to the request" field_type="unsigned byte" field_units="one"/>
                    </record>
                </body>
                <footer name="Footer"/>
            </message_def>
        </input_set>
        <output_set>
            <message_def name="ReportDigitalResourceEndpoint" message_id="4816" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
```

```
                    <list name="DigitalResourceEndpointList" optional="false">
                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned byte"/>
                        <record name="DigitalResourceEndpointRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned byte"/>
                            <fixed_field name="ServerType" optional="false" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="RTSP"/>
                                    <value_enum enum_index="1" enum_const="MPEG2_TS"/>
                                    <value_enum enum_index="2" enum_const="FTP"/>
                                    <value_enum enum_index="3" enum_const="SFTP"/>
                                    <value_enum enum_index="4" enum_const="FTP_over_SSH"/>
                                    <value_enum enum_index="5" enum_const="HTTP"/>
                                    <value_enum enum_index="6" enum_const="HTTPS"/>
                                    <value_enum enum_index="7" enum_const="SCP"/>
                                    <value_enum enum_index="8" enum_const="CCSI" interpretation="where URL is of the
form 'ip address:port number' (no quotes)"/>
                                    <value_range lower_limit="9" lower_limit_type="inclusive" upper_limit="200"
upper_limit_type="inclusive" interpretation="Reserved for future use"/>
                                    <value_range lower_limit="201" lower_limit_type="inclusive" upper_limit="255"
upper_limit_type="inclusive" interpretation="Reserved for program or implementation specific use"/>
                                </value_set>
                            </fixed_field>
                            <variable_length_string name="ServerURL" optional="false" interpretation="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.">
                                <count_field min_count="0" max_count="255" field_type_unsigned="unsigned byte"/>
                            </variable_length_string>
                            <bit_field name="JAUS_ID" optional="true" field_type_unsigned="unsigned integer">
                                <sub_field name="ComponentID">
                                    <bit_range from_index="0" to_index="7" interpretation="ComponentID"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="0" lower_limit_type="inclusive" upper_limit="255"
upper_limit_type="inclusive"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="NodeID">
                                    <bit_range from_index="8" to_index="15" interpretation="NodeID"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="0" lower_limit_type="inclusive" upper_limit="255"
upper_limit_type="inclusive"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="SubsystemID">
                                    <bit_range from_index="16" to_index="31" interpretation="SubsystemID"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="0" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                    </value_set>
                                </sub_field>
                            </bit_field>
                            <fixed_field name="ResourceID" optional="true" interpretation="The ID used by the
configuration and control service to identify this source. This is the SensorID for visual sensors."
field_type="unsigned short integer" field_units="one"/>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="ConfirmDigitalResourceEndpoint" message_id="080A" is_command="false">
                <description xml:space="preserve">Confirm a digital resource endpoint registration or removal.
Provides a unique ID for referencing the server in the future.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="ConfirmDigitalResourceEndpointRec" optional="false">
                        <fixed_field name="ID" optional="false" interpretation="Unique ID identifying this resource
endpoint" field_type="unsigned byte" field_units="one"/>
                        <fixed_field name="RequestID" optional="false" interpretation="Client provided ID to link the
response to the request" field_type="unsigned byte" field_units="one"/>
                    </record>
                </body>
                <footer name="Footer"/>
            </message_def>
        </output_set>
    </message_set>
    <internal_events_set/>
```

```
    <protocol_behavior is_stateless="false">
        <start state_machine_name="Events.transport.ReceiveFSM" state_name="Receiving.Ready"/>
        <start state_machine_name="Events.transport.SendFSM" state_name="Sending"/>
        <state_machine name="Events.transport.ReceiveFSM">
            <state name="Receiving" initial_state="Ready">
                <state name="Ready">
                    <default_state>
                        <transition name="events.transport.Receive">
                            <parameter type="QueryDigitalResourceEndpoint" value="msg" interpretation="enveloped Query
Digital Resource Endpoint message"/>
                            <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                            <internal/>
                            <action name="sendReportDigitalResourceEndpoint" interpretation="Send a
ReportDigitalResourceEndpoint message to querying client">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                        </transition>
                        <transition name="events.transport.Receive">
                            <parameter type="RegisterDigitalResourceEndpoint" value="msg" interpretation="enveloped
Register Digital Resource Endpoint message"/>
                            <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                            <internal/>
                            <action name="AddDigitalResourceEndpoint" interpretation="Adds the specified endpoint to
the list of known endpoints">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                            <action name="sendConfirmDigitalResourceEndpoint" interpretation="Send a
ConfirmDigitalResourceEndpoint message to querying client">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                        </transition>
                        <transition name="events.transport.Receive">
                            <parameter type="RemoveDigitalResourceEndpoint" value="msg" interpretation="enveloped
Remove Digital Resource Endpoint message"/>
                            <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transportData"/>
                            <internal/>
                            <action name="RemoveDigitalResourceEndpoint" interpretation="Removes the specified endpoint
from the list of known endpoints">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                            <action name="sendConfirmDigitalResourceEndpoint" interpretation="Send a
ConfirmDigitalResourceEndpoint message to querying client">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                        </transition>
                    </default_state>
                                </state>
                            </state>
                                </state_machine>
        <state_machine name="Events.transport.SendFSM">
            <state name="Sending">
                            </state>
                                </state_machine>
    </protocol_behavior>
</service_def>
```


A.5 DIGITALVIDEO

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="DigitalVideo" id="urn:jaus:jss:environmentSensing:DigitalVideo" version="2.0"
xmlns="urn:jaus:jsidl:1.1" xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">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.</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost, or reordered.</assumptions>
    <references>
        <inherits_from name="visualSensor" id="urn:jaus:jss:environmentSensing:VisualSensor" version="2.0"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="ControlDigitalVideoSensorStream" message_id="0805" is_command="true">
                <description xml:space="preserve">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 redunant mechanism and a service client may choose to use either the native protocol or this
message for stream control.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="ControlDigitalVideoSensorStreamRec" optional="false">
                        <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is not a
valid ID in this message." field_type="unsigned short integer" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="StreamState" optional="false" field_type="unsigned byte" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="Play"/>
                                <value_enum enum_index="1" enum_const="Pause"/>
                                <value_enum enum_index="2" enum_const="Stop"/>
                            </value_set>
                        </fixed_field>
                    </record>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="ControlDigitalVideoSensorRecording" message_id="0810" is_command="true">
                <description xml:space="preserve">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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="ControlDigitalVideoSensorRecordingRec" optional="false">
                        <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is not a
valid ID in this message." field_type="unsigned short integer" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="RequestedRecordingState" optional="false" field_type="unsigned byte"
field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NotRecording"/>
                                <value_enum enum_index="1" enum_const="Recording"/>
                            </value_set>
                        </fixed_field>
                    </record>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="SetDigitalVideoSensorConfiguration" message_id="0804" is_command="true">
                <description xml:space="preserve">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).</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
```

```
            </header>
            <body name="Body">
                <sequence name="DigitalVideoSensorConfigurationSequence" optional="false">
                    <record name="RequestIdRec" optional="false">
                        <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                    </record>
                    <list name="DigitalVideoSensorList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="DigitalVideoSensorConfigurationRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned byte"/>
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is
not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="MinimumBitRate" optional="true" interpretation="measured kilobits
per second" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="MaximumBitRate" optional="true" interpretation="measured kilobits
per second" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="FrameRate" optional="true" field_type="unsigned byte"
field_units="one"/>
                            <fixed_field name="FrameSize" optional="true" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="128x96 (sqcif)"/>
                                    <value_enum enum_index="1" enum_const="176x144 (qcif)"/>
                                    <value_enum enum_index="2" enum_const="352x288 (cif)"/>
                                    <value_enum enum_index="3" enum_const="704x576 (4cif)"/>
                                    <value_enum enum_index="4" enum_const="1408x1152 (16cif)"/>
                                    <value_enum enum_index="5" enum_const="160x120 (qqvga)"/>
                                    <value_enum enum_index="6" enum_const="320x240 (qvga)"/>
                                    <value_enum enum_index="7" enum_const="640x480 (vga)"/>
                                    <value_enum enum_index="8" enum_const="800x600 (svga)"/>
                                    <value_enum enum_index="9" enum_const="1024x768 (xga)"/>
                                    <value_enum enum_index="10" enum_const="1600x1200 (uxga)"/>
                                    <value_enum enum_index="11" enum_const="2048x1536 (qxga)"/>
                                    <value_enum enum_index="12" enum_const="1280x1024 (sxga)"/>
                                    <value_enum enum_index="13" enum_const="2560x2048 (qsxga)"/>
                                    <value_enum enum_index="14" enum_const="5120x4096 (hsxga)"/>
                                    <value_enum enum_index="15" enum_const="852x480 (wvga)"/>
                                    <value_enum enum_index="16" enum_const="1366x768 (wxga)"/>
                                    <value_enum enum_index="17" enum_const="1600x1024 (wsxga)"/>
                                    <value_enum enum_index="18" enum_const="1920x1200 (wuxga)"/>
                                    <value_enum enum_index="19" enum_const="2560x1600 (woxga)"/>
                                    <value_enum enum_index="20" enum_const="3200x2048 (wqsxga)"/>
                                    <value_enum enum_index="21" enum_const="3840x2400 (wquxga)"/>
                                    <value_enum enum_index="22" enum_const="6400x4096 (whsxga)"/>
                                    <value_enum enum_index="23" enum_const="7680x4800 (whuxga)"/>
                                    <value_enum enum_index="24" enum_const="320x200 (cga)"/>
                                    <value_enum enum_index="25" enum_const="640x350 (ega)"/>
                                    <value_enum enum_index="26" enum_const="852x480 (hd480)"/>
                                    <value_enum enum_index="27" enum_const="1280x720 (hd720)"/>
                                    <value_enum enum_index="28" enum_const="1920x1080 (hd1080)"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="DigitalFormat" optional="true" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AVI"/>
                                    <value_enum enum_index="1" enum_const="MJPEG"/>
                                    <value_enum enum_index="2" enum_const="MPEG-2"/>
                                    <value_enum enum_index="3" enum_const="H.263"/>
                                    <value_enum enum_index="4" enum_const="H.263+"/>
                                    <value_enum enum_index="5" enum_const="MPEG-4 Visual (MPEG-4 Part 2)"/>
                                    <value_enum enum_index="6" enum_const="MPEG-4 AVC"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </sequence>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="QueryDigitalVideoSensorConfiguration" message_id="2809" is_command="false">
            <description xml:space="preserve">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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="QueryDigitalVideoConfigurationList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="QueryDigitalVideoConfigurationRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="QueryPresenceVector" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryDigitalVideoSensorCapabilitiesExt" message_id="281D" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="QueryDigitalVideoSensorCapabilitiesList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="QueryDigitalVideoSensorCapabilitiesRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="QueryPresenceVector" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryDigitalVideoSensorStatus" message_id="281E" is_command="false">
                <description xml:space="preserve">This message shall cause the
   receiving service to reply to the requestor with a Report
   Digital Video Sensor Status message.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="SensorIdList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="SensorIDQueryRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                                <value_set offset_to_lower_limit="false">
```

```
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
        </input_set>
        <output_set>
            <message_def name="ReportDigitalVideoSensorCapabilitiesExt" message_id="481D" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="DigitalVideoSensorList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="DigitalVideoSensorCapabilitiesRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned byte"/>
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is not
a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="MinimumBitRate" optional="true" interpretation="measured kilobits per
second" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="MaximumBitRate" optional="true" interpretation="measured kilobits per
second" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="MinimumFrameRate" optional="true" field_type="unsigned byte"
field_units="one"/>
                            <fixed_field name="MaximumFrameRate" optional="true" field_type="unsigned byte"
field_units="one"/>
                            <bit_field name="SupportedFrameSizes" optional="true" field_type_unsigned="unsigned
integer">
                                <sub_field name="sqcif_128x96">
                                    <bit_range from_index="0" to_index="0"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="qcif_176x144">
                                    <bit_range from_index="1" to_index="1"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="cif_352x288">
                                    <bit_range from_index="2" to_index="2"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="cif4_704x576">
                                    <bit_range from_index="3" to_index="3"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="cif16_1408x1152">
                                    <bit_range from_index="4" to_index="4"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="qqvga_160x120">
                                    <bit_range from_index="5" to_index="5"/>
```

```
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="qvga_320x240">
        <bit_range from_index="6" to_index="6"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="vga_640x480">
        <bit_range from_index="7" to_index="7"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="svga_800x600">
        <bit_range from_index="8" to_index="8"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="xga_1024x768">
        <bit_range from_index="9" to_index="9"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="uxga_1600x1200">
        <bit_range from_index="10" to_index="10"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="qxga_2048x1536">
        <bit_range from_index="11" to_index="11"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="sxga_1280x1024">
        <bit_range from_index="12" to_index="12"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="qsxga_2560x2048">
        <bit_range from_index="13" to_index="13"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="hsxga_5120x4096">
        <bit_range from_index="14" to_index="14"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="wvga_852x480">
        <bit_range from_index="15" to_index="15"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="wxga_1366x768">
        <bit_range from_index="16" to_index="16"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="Unsupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
```

```
</sub_field>
<sub_field name="wsxga_1600x1024">
    <bit_range from_index="17" to_index="17"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="wuxga_1920x1200">
    <bit_range from_index="18" to_index="18"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="woxga_2560x1600">
    <bit_range from_index="19" to_index="19"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="wqsxga_3200x2048">
    <bit_range from_index="20" to_index="20"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="wquxga_3840x2400">
    <bit_range from_index="21" to_index="21"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="whsxga_6400x4096">
    <bit_range from_index="22" to_index="22"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="whuxga_7680x4800">
    <bit_range from_index="23" to_index="23"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="cga_320x200">
    <bit_range from_index="24" to_index="24"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="ega_640x350">
    <bit_range from_index="25" to_index="25"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="hd480_852x480">
    <bit_range from_index="26" to_index="26"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="hd720_1280x720">
    <bit_range from_index="27" to_index="27"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="Unsupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="hd1080_1920x1080">
    <bit_range from_index="28" to_index="28"/>
    <value_set offset_to_lower_limit="false">
```

```
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                        </bit_field>
                        <bit_field name="SupportedDigitalFormats" optional="true" field_type_unsigned="unsigned
byte">
                            <sub_field name="AVI">
                                <bit_range from_index="0" to_index="0"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="MJPEG">
                                <bit_range from_index="1" to_index="1"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="MPEG2">
                                <bit_range from_index="2" to_index="2"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="h263">
                                <bit_range from_index="3" to_index="3"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="h263plus">
                                <bit_range from_index="4" to_index="4"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="MPEG4_Visual">
                                <bit_range from_index="5" to_index="5"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="MPEG4_AVC_h264">
                                <bit_range from_index="6" to_index="6"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                        </bit_field>
                        <fixed_field name="RecordingSupported" optional="true" interpretation="Indicates if the
implementation supports local recording of video for the sensor" field_type="unsigned byte" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="Unsupported"/>
                                <value_enum enum_index="1" enum_const="Supported"/>
                            </value_set>
                        </fixed_field>
                    </record>
                </list>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="ReportDigitalVideoSensorConfiguration" message_id="4809" is_command="false">
            <description xml:space="preserve">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.</description>
            <header name="AppHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="DigitalVideoSensorConfigurationList" optional="false">
```

```
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="DigitalVideoSensorConfigurationRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned byte"/>
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is not
a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="MinimumBitRate" optional="true" interpretation="measured kilobits per
second" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="MaximumBitRate" optional="true" interpretation="measured kilobits per
second" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="FrameRate" optional="true" field_type="unsigned byte"
field_units="one"/>
                            <fixed_field name="FrameSize" optional="true" field_type="unsigned byte" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="128x96 (sqcif)"/>
                                    <value_enum enum_index="1" enum_const="176x144 (qcif)"/>
                                    <value_enum enum_index="2" enum_const="352x288 (cif)"/>
                                    <value_enum enum_index="3" enum_const="704x576 (4cif)"/>
                                    <value_enum enum_index="4" enum_const="1408x1152 (16cif)"/>
                                    <value_enum enum_index="5" enum_const="160x120 (qqvga)"/>
                                    <value_enum enum_index="6" enum_const="320x240 (qvga)"/>
                                    <value_enum enum_index="7" enum_const="640x480 (vga)"/>
                                    <value_enum enum_index="8" enum_const="800x600 (svga)"/>
                                    <value_enum enum_index="9" enum_const="1024x768 (xga)"/>
                                    <value_enum enum_index="10" enum_const="1600x1200 (uxga)"/>
                                    <value_enum enum_index="11" enum_const="2048x1536 (qxga)"/>
                                    <value_enum enum_index="12" enum_const="1280x1024 (sxga)"/>
                                    <value_enum enum_index="13" enum_const="2560x2048 (qsxga)"/>
                                    <value_enum enum_index="14" enum_const="5120x4096 (hsxga)"/>
                                    <value_enum enum_index="15" enum_const="852x480 (wvga)"/>
                                    <value_enum enum_index="16" enum_const="1366x768 (wxga)"/>
                                    <value_enum enum_index="17" enum_const="1600x1024 (wsxga)"/>
                                    <value_enum enum_index="18" enum_const="1920x1200 (wuxga)"/>
                                    <value_enum enum_index="19" enum_const="2560x1600 (woxga)"/>
                                    <value_enum enum_index="20" enum_const="3200x2048 (wqsxga)"/>
                                    <value_enum enum_index="21" enum_const="3840x2400 (wquxga)"/>
                                    <value_enum enum_index="22" enum_const="6400x4096 (whsxga)"/>
                                    <value_enum enum_index="23" enum_const="7680x4800 (whuxga)"/>
                                    <value_enum enum_index="24" enum_const="320x200 (cga)"/>
                                    <value_enum enum_index="25" enum_const="640x350 (ega)"/>
                                    <value_enum enum_index="26" enum_const="852x480 (hd480)"/>
                                    <value_enum enum_index="27" enum_const="1280x720 (hd720)"/>
                                    <value_enum enum_index="28" enum_const="1920x1080 (hd1080)"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="DigitalFormat" optional="true" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AVI"/>
                                    <value_enum enum_index="1" enum_const="MJPEG"/>
                                    <value_enum enum_index="2" enum_const="MPEG-2"/>
                                    <value_enum enum_index="3" enum_const="H.263"/>
                                    <value_enum enum_index="4" enum_const="H.263+"/>
                                    <value_enum enum_index="5" enum_const="MPEG-4 Visual (MPEG-4 Part 2)"/>
                                    <value_enum enum_index="6" enum_const="MPEG-4 AVC"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="ReportDigitalVideoSensorStatus" message_id="481E" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="DigitalVideoSensorStatusList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="DigitalVideoSensorStatusRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is not
```

```
a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="StreamingState" optional="false" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Play"/>
                                    <value_enum enum_index="1" enum_const="Pause"/>
                                    <value_enum enum_index="2" enum_const="Stop"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="RecordingState" optional="false" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NotRecording"/>
                                    <value_enum enum_index="1" enum_const="Recording"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
        </output_set>
    </message_set>
    <internal_events_set/>
    <protocol_behavior is_stateless="false">
        <start state_machine_name="visualSensor.accessControl.events.transport.ReceiveFSM"
state_name="Receiving.Ready.NotControlled"/>
        <start state_machine_name="visualSensor.accessControl.events.transport.SendFSM" state_name="Sending"/>
        <state_machine name="visualSensor.accessControl.events.transport.ReceiveFSM">
            <state name="Receiving" initial_state="Ready">
                <state name="Ready" initial_state="NotControlled">
                    <state name="NotControlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalVideoSensorCapabilitiesExt" value="msg"
interpretation="enveloped QueryDigitalVideoSensorCapabilitiesExt message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportDigitalVideoSensorCapabilitiesExt" interpretation="Send a
ReportDigitalVideoSensorCapabilitiesExt message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalVideoSensorConfiguration" value="msg"
interpretation="enveloped QueryDigitalVideoSensorConfiguration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportDigitalVideoSensorConfiguration" interpretation="Send a
ReportDigitalVideoSensorConfiguration message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalVideoSensorStatus" value="msg" interpretation="enveloped
QueryDigitalVideoSensorStatus message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportDigitalVideoSensorStatus" interpretation="Send a
ReportDigitalVideoSensorStatus message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                        </state>
                    <state name="Controlled" initial_state="Available">
```

```
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="SetDigitalVideoSensorConfiguration" value="msg"
interpretation="enveloped SetDigitalVideoSensorConfiguration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <guard condition="accessControl.isControllingClient( transportData )"
interpretation="True if the message that triggered the          transition is received from the client that is in
control          of this service."/>
                                <internal/>
                                <action name="sendConfirmSensorConfiguration" interpretation="Send
sendConfirmSensorConfiguration message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                                <action name="updateDigitalVideoSensorConfiguration" interpretation="Update the sensor
user controllable          configuration parameters according to the ones specified.">
                                    <argument value="msg"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="ControlDigitalVideoSensorStream" value="msg" interpretation="enveloped
ControlDigitalVideoSensorStream message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <guard condition="accessControl.isControllingClient( transportData )"
interpretation="True if the message that triggered the          transition is received from the client that is in
control of this service."/>
                                <internal/>
                                <action name="modifyDigitalVideoSensorStream" interpretation="Modify the video stream
according to the specified message.">
                                    <argument value="msg"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="ControlDigitalVideoSensorRecording" value="msg"
interpretation="enveloped ControlDigitalVideoSensorRecording message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <guard condition="accessControl.isControllingClient( transportData )  &amp;amp;&amp;amp;
isSupported( msg )" interpretation="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."/>
                                <internal/>
                                <action name="updateDigitalVideoSensorRecordingState" interpretation="Update the video
recording state according to the specified message.">
                                    <argument value="msg"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalVideoSensorCapabilitiesExt" value="msg"
interpretation="enveloped QueryDigitalVideoSensorCapabilitiesExt message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportDigitalVideoSensorCapabilitiesExt" interpretation="Send a
ReportDigitalVideoSensorCapabilitiesExt message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalVideoSensorConfiguration" value="msg"
interpretation="enveloped QueryDigitalVideoSensorConfiguration message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportDigitalVideoSensorConfiguration" interpretation="Send a
ReportDigitalVideoSensorConfiguration message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="accessControl.events.transport.Receive">
                                <parameter type="QueryDigitalVideoSensorStatus" value="msg" interpretation="enveloped
QueryDigitalVideoSensorStatus message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
```

```
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportDigitalVideoSensorStatus" interpretation="Send a
ReportDigitalVideoSensorStatus message">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                            </state>
                    <default_state>
                                            </default_state>
                                </state>
                            </state>
                                </state_machine>
        <state_machine name="visualSensor.accessControl.events.transport.SendFSM">
            <state name="Sending">
                            </state>
                                </state_machine>
    </protocol_behavior>
</service_def>
```

A.6 FORCETORQUESENSOR

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="ForceTorqueSensor" id="urn:jaus:jss:environmentSensing:ForceTorqueSensor" version="2.0"
xmlns="urn:jaus:jsidl:1.1" xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">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).</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost, or reordered.</assumptions>
    <references>
        <inherits_from name="events" id="urn:jaus:jss:core:Events" version="1.1"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="QueryForceTorqueCapabilities" message_id="280B" is_command="false">
                <description xml:space="preserve">This message is used to query the capabilities for one or more
force/torque devices.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="QueryForceTorqueCapabilitiesList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="QueryForceTorqueCapabilitiesRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="The ID of the sensor.  A
value of zero means all sensors" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="QueryPresenceVector" optional="false" interpretation="See
ReportForceTorqueCapabilities" field_type="unsigned short integer" field_units="one"/>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryForceTorque" message_id="280C" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="QueryForceTorqueList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="QueryForceTorqueRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="The ID of the sensor.  A
value of zero means all sensors" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="CoordinateSystem" optional="false" field_type="unsigned byte"
field_units="one">
```

```
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="NativeCoordinateSystems"/>
                                    <value_enum enum_index="1" enum_const="VehicleCoordinateSystems"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="QueryPresenceVector" optional="false" interpretation="See
ReportForceTorqueRec" field_type="unsigned byte" field_units="one"/>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
        </input_set>
        <output_set>
            <message_def name="ReportForceTorqueCapabilities" message_id="480B" is_command="false">
                <description xml:space="preserve">This message is used to report the capabilities of one or more
force/torque sensing devices.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="ReportForceTorqueCapabilitiesList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="ReportForceTorqueCapabilitiesRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned short integer"/>
                            <fixed_field name="SensorID" optional="false" interpretation="The ID of the sensor.  A
value of zero means all sensors" field_type="unsigned short integer" field_units="one"/>
                            <variable_length_string name="SensorName" optional="false">
                                <count_field min_count="0" max_count="255" field_type_unsigned="unsigned byte"/>
                            </variable_length_string>
                            <fixed_field name="CoordinateTransformSupported" optional="true" interpretation="True or
false depending on if the service supports transforming data into the vehicle coordinate system." field_type="unsigned
byte" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="False"/>
                                    <value_enum enum_index="1" enum_const="True"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="MinForceX" optional="true" interpretation="Most negative force
measurable in the X-direction." field_type="unsigned short integer" field_units="newton">
                                <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MaxForceX" optional="true" interpretation="Most positive force
measurable in the X-direction." field_type="unsigned short integer" field_units="newton">
                                <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MinForceY" optional="true" interpretation="Most negative force
measurable in the Y-direction." field_type="unsigned short integer" field_units="newton">
                                <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MaxForceY" optional="true" interpretation="Most positive force
measurable in the Y-direction." field_type="unsigned short integer" field_units="newton">
                                <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MinForceZ" optional="true" interpretation="Most negative force
measurable in the Z-direction." field_type="unsigned short integer" field_units="newton">
                                <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MaxForceZ" optional="true" interpretation="Most positive force
measurable in the Z-direction." field_type="unsigned short integer" field_units="newton">
                                <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MinTorqueX" optional="true" interpretation="Most negative torque
measurable around the X-axis." field_type="unsigned short integer" field_units="newton meter">
                                <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MaxTorqueX" optional="true" interpretation="Most positive torque
measurable around the X-axis." field_type="unsigned short integer" field_units="newton meter">
                                <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                            </fixed_field>
```

```
                        <fixed_field name="MinTorqueY" optional="true" interpretation="Most negative torque
measurable around the Y-axis." field_type="unsigned short integer" field_units="newton meter">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="MaxTorqueY" optional="true" interpretation="Most positive torque
measurable around the Y-axis." field_type="unsigned short integer" field_units="newton meter">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="MinTorqueZ" optional="true" interpretation="Most negative torque
measurable around the Z-axis." field_type="unsigned short integer" field_units="newton meter">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="MaxTorqueZ" optional="true" interpretation="Most positive torque
measurable around the Z-axis." field_type="unsigned short integer" field_units="newton meter">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                    </record>
                </list>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="ReportForceTorque" message_id="480C" is_command="false">
            <description xml:space="preserve">This message is used to report the measured force/torque for one or
more sensing devices.</description>
            <header name="JAUSApplicationLayerHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="ReportForceTorqueList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <record name="ReportForceTorqueRec" optional="false">
                        <presence_vector field_type_unsigned="unsigned byte"/>
                        <fixed_field name="SensorID" optional="false" interpretation="The ID of the sensor.  A
value of zero is not allowed" field_type="unsigned short integer" field_units="one"/>
                        <fixed_field name="CoordinateSystem" optional="false" field_type="unsigned byte"
field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="NativeCoordinateSystems"/>
                                <value_enum enum_index="1" enum_const="VehicleCoordinateSystems"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="ForceX" optional="true" interpretation="Current force measured in the X-
direction" field_type="unsigned short integer" field_units="newton">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="ForceY" optional="true" interpretation="Current force measured in the Y-
direction" field_type="unsigned short integer" field_units="newton">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="ForceZ" optional="true" interpretation="Current force measured in the Z-
direction" field_type="unsigned short integer" field_units="newton">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="TorqueX" optional="true" interpretation="Current torque measured around
the X-axis" field_type="unsigned short integer" field_units="newton meter">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="TorqueY" optional="true" interpretation="Current torque measured around
the Y-axis" field_type="unsigned short integer" field_units="newton meter">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                        <fixed_field name="TorqueZ" optional="true" interpretation="Current torque measured around
the Z-axis" field_type="unsigned short integer" field_units="newton meter">
                            <scale_range real_lower_limit="-5000" real_upper_limit="5000"
integer_function="round"/>
                        </fixed_field>
                    </record>
                </list>
            </body>
```

```
                <footer name="Footer"/>
            </message_def>
        </output_set>
    </message_set>
    <internal_events_set/>
    <protocol_behavior is_stateless="false">
        <start state_machine_name="events.transport.ReceiveFSM" state_name="Receiving.Ready"/>
        <start state_machine_name="events.transport.SendFSM" state_name="Sending"/>
        <state_machine name="events.transport.ReceiveFSM">
            <state name="Receiving" initial_state="Ready">
                <state name="Ready">
                    <default_state>
                        <transition name="events.transport.Receive">
                            <parameter type="QueryForceTorque" value="msg"/>
                            <parameter type="Receive.Body.ReceiveRec" value="transportData"/>
                            <internal/>
                            <action name="sendReportForceTorque" interpretation="Send a ReportForceTorque message">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                        </transition>
                        <transition name="events.transport.Receive">
                            <parameter type="QueryForceTorqueCapabilities" value="msg"/>
                            <parameter type="Receive.Body.ReceiveRec" value="transportData"/>
                            <internal/>
                            <action name="sendReportForceTorqueCapabilities" interpretation="Send a
ReportForceTorqueCapabilities message">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                        </transition>
                    </default_state>
                                </state>
                            </state>
                                </state_machine>
        <state_machine name="events.transport.SendFSM">
            <state name="Sending">
                            </state>
                                </state_machine>
    </protocol_behavior>
</service_def>
```

A.7 H264VIDEOENCODING

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="H264VideoEncoding" id="urn:jaus:jss:environmentSensing:H264VideoEncoding" version="2.0"
xmlns="urn:jaus:jsidl:1.1" xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">The H264 Video Encoding Service provides a mechanism for querying and configuring
the H264 encoding of one or more video sensors</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost, or reordered.</assumptions>
    <references>
        <inherits_from name="DigitalVideo" id="urn:jaus:jss:environmentSensing:DigitalVideo" version="2.0"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="SetH264VideoEncodingConfiguration" message_id="080D" is_command="true">
                <description xml:space="preserve">This message is used to set the active configuration for one or more
H264 encoded video streams.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="H264ConfigurationListSequence" optional="false">
                        <record name="RequestIdRec" optional="false">
                            <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                        <list name="H264ConfigurationList" optional="false">
                            <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                            <sequence name="H264ConfigurationSequence" optional="false">
                                <presence_vector field_type_unsigned="unsigned byte"/>
                                <record name="H264ConfigurationRec" optional="false">
                                    <presence_vector field_type_unsigned="unsigned byte"/>
                                    <fixed_field name="SensorID" optional="false" interpretation="Note: Zero is not a
```

```
valid VisualSensorID value" field_type="unsigned short integer" field_units="one"/>
                                    <fixed_field name="Profile" optional="true" interpretation="Enumeration for h264
profile setting." field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Baseline"/>
                                            <value_enum enum_index="1" enum_const="ConstrainedBaseline"/>
                                            <value_enum enum_index="2" enum_const="Main"/>
                                            <value_enum enum_index="3" enum_const="Extended"/>
                                            <value_enum enum_index="4" enum_const="High"/>
                                            <value_enum enum_index="5" enum_const="ProgressiveHigh"/>
                                            <value_enum enum_index="6" enum_const="ConstrainedHigh"/>
                                            <value_enum enum_index="7" enum_const="High10"/>
                                            <value_enum enum_index="8" enum_const="High422"/>
                                            <value_enum enum_index="9" enum_const="High444Predictive"/>
                                            <value_enum enum_index="10" enum_const="High10Intra"/>
                                            <value_enum enum_index="11" enum_const="High422Intra"/>
                                            <value_enum enum_index="12" enum_const="High444Intra"/>
                                            <value_enum enum_index="13" enum_const="CAVLC444"/>
                                            <value_enum enum_index="14" enum_const="ScalableBaseline"/>
                                            <value_enum enum_index="15" enum_const="ScalableConstrainedBaseline"/>
                                            <value_enum enum_index="16" enum_const="ScalableHigh"/>
                                            <value_enum enum_index="17" enum_const="ScalableConstrainedHigh"/>
                                            <value_enum enum_index="18" enum_const="ScalableHighIntra"/>
                                            <value_enum enum_index="19" enum_const="StereoHigh"/>
                                            <value_enum enum_index="20" enum_const="MultiviewHigh"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="Level" optional="true" interpretation="Enumeration for target
h264 level." field_type="unsigned integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Level_1"/>
                                            <value_enum enum_index="1" enum_const="Level_1B"/>
                                            <value_enum enum_index="2" enum_const="Level_11"/>
                                            <value_enum enum_index="3" enum_const="Level_12"/>
                                            <value_enum enum_index="4" enum_const="Level_13"/>
                                            <value_enum enum_index="5" enum_const="Level_2"/>
                                            <value_enum enum_index="6" enum_const="Level_21"/>
                                            <value_enum enum_index="7" enum_const="Level_22"/>
                                            <value_enum enum_index="8" enum_const="Level_3"/>
                                            <value_enum enum_index="9" enum_const="Level_31"/>
                                            <value_enum enum_index="10" enum_const="Level_32"/>
                                            <value_enum enum_index="11" enum_const="Level_4"/>
                                            <value_enum enum_index="12" enum_const="Level_41"/>
                                            <value_enum enum_index="13" enum_const="Level_42"/>
                                            <value_enum enum_index="14" enum_const="Level_5"/>
                                            <value_enum enum_index="15" enum_const="Level_51"/>
                                            <value_enum enum_index="16" enum_const="Level_52"/>
                                            <value_enum enum_index="17" enum_const="Level_6"/>
                                            <value_enum enum_index="18" enum_const="Level_61"/>
                                            <value_enum enum_index="19" enum_const="Level_62"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="Preset" optional="true" interpretation="Preconfigured settings
for hardware specific H264 settings.  The actual values for each preset are not specified by the standard and are up to
the implementation." field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="LowLatency"/>
                                            <value_enum enum_index="1" enum_const="SlowComms"/>
                                            <value_enum enum_index="2" enum_const="PersistentStare"/>
                                            <value_enum enum_index="3" enum_const="DriveVision"/>
                                            <value_enum enum_index="4" enum_const="Manipulation"/>
                                            <value_enum enum_index="5" enum_const="BestQuality"/>
                                            <value_enum enum_index="12" enum_const="ProgramSpecific_1"/>
                                            <value_enum enum_index="13" enum_const="ProgramSpecific_2"/>
                                            <value_enum enum_index="14" enum_const="ProgramSpecific_3"/>
                                            <value_enum enum_index="15" enum_const="ProgramSpecific_4"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="GroupOfPictures" optional="true" interpretation="Number of
interim frames before sending a key frame" field_type="unsigned byte" field_units="one"/>
                                    <fixed_field name="GradualDecoderRefreshOnOff" optional="true" interpretation="Turn
on/off Gradual Decoder Refresh, also called Periodic Intra Refresh." field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="OFF"/>
                                            <value_enum enum_index="1" enum_const="ON"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="RegionOfInterestOnOff" optional="true" interpretation="Turn
on/off region of interest (foveation) encoding, if supported.  The region of interest is specified by the
RegionOfInterestParams array." field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
```

```
                                        <value_enum enum_index="0" enum_const="OFF"/>
                                        <value_enum enum_index="1" enum_const="ON"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="IntraMacroRefresh" optional="true" interpretation="This
parameter is the random intra macroblock refresh; it provides the refresh rate in Macroblocks per picture."
field_type="unsigned short integer" field_units="one">
                                    <scale_range real_lower_limit="0" real_upper_limit="100"
integer_function="round"/>
                                </fixed_field>
                            </record>
                            <record name="RegionOfInterestParams" optional="true">
                                <fixed_field name="X_Start" optional="false" interpretation="Horizontal start
location of high bitrate window, measured as percent of total image width from the left" field_type="unsigned short
integer" field_units="percent">
                                    <scale_range real_lower_limit="0" real_upper_limit="100"
integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="Y_Start" optional="false" interpretation="Vertical start
location of high bitrate window, measured as percent of total image height from the top" field_type="unsigned short
integer" field_units="percent">
                                    <scale_range real_lower_limit="0" real_upper_limit="100"
integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="Width" optional="false" interpretation="Width of high bitrate
window, measured as percent of total image width" field_type="unsigned short integer" field_units="percent">
                                    <scale_range real_lower_limit="0" real_upper_limit="100"
integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="Height" optional="false" interpretation="Height of high bitrate
window , measured as percent of total image height" field_type="unsigned short integer" field_units="percent">
                                    <scale_range real_lower_limit="0" real_upper_limit="100"
integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="HighBitRate" optional="false" interpretation="Bitrate of ROI
window, measured in kilobits per second" field_type="unsigned short integer" field_units="one"/>
                                <fixed_field name="LowBitRate" optional="false" interpretation="Bitrate of ROI
window, measured in kilobits per second" field_type="unsigned short integer" field_units="one"/>
                            </record>
                            <record name="GradualDecoderRefreshParams" optional="true">
                                <fixed_field name="GDR_Duration" optional="false" interpretation="The number of
frames needed to completely refresh the image" field_type="unsigned short integer" field_units="one"/>
                                <fixed_field name="GDR_Interval" optional="false" interpretation="The number of
frames that need to be transmitted before starting a new GDR cycle. Note that the interval must be greater than or
equal to the duration" field_type="unsigned short integer" field_units="one"/>
                            </record>
                        </sequence>
                    </list>
                </sequence>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="QueryH264VideoEncodingCapabilities" message_id="280E" is_command="false">
            <description xml:space="preserve">This message must be used to query for the H264 encoding capabilities
of one or more visual sensors on the receiving service.</description>
            <header name="JAUSApplicationLayerHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="QueryH264VideoEncodingCapabilitiesList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <record name="QueryH264VideoEncodingRec" optional="false">
                        <fixed_field name="SensorID" optional="false" interpretation="The ID of the sensor for
which the capabilities are being queried.Note: A value of 0 shall be interpreted as all associated sensors."
field_type="unsigned short integer" field_units="one"/>
                    </record>
                </list>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="QueryH264VideoEncodingConfiguration" message_id="280D" is_command="false">
            <description xml:space="preserve">This message must be used to query for the currently configured H264
encoding parameters.</description>
            <header name="JAUSApplicationLayerHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                </record>
```

```
                </header>
                <body name="Body">
                    <list name="QueryH264VideoEncodingConfigurationList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="QueryH264VideoEncodingRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="The ID of the sensor for
which the capabilities are being queried.Note: A value of 0 shall be interpreted as all associated sensors."
field_type="unsigned short integer" field_units="one"/>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
        </input_set>
        <output_set>
            <message_def name="ConfirmSensorConfiguration" message_id="0801" is_command="false">
                <description xml:space="preserve">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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="ConfirmSensorConfigurationSequence" optional="false">
                        <record name="RequestIdRec" optional="false">
                            <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                        <list name="ConfirmSensorList" optional="false">
                            <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                            <variant name="ConfirmSensorConfigurationVariant" optional="false">
                                <vtag_field min_count="0" max_count="9" field_type_unsigned="unsigned byte"/>
                                <record name="SensorIdRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                </record>
                                <record name="RangeSensorErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="RangeSensorErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid Horizontal Field of View"/>
                                            <value_enum enum_index="2" enum_const="Invalid Vertical Field of View"/>
                                            <value_enum enum_index="3" enum_const="Invalid Update Rate"/>
                                            <value_enum enum_index="4" enum_const="Invalid Sensor Range"/>
                                            <value_enum enum_index="5" enum_const="Invalid Sensor State"/>
                                            <value_enum enum_index="6" enum_const="Multiple Invalid Parameters"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="VisualSensorErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
```

```
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="VisualSensorErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Sensor State"/>
                                        <value_enum enum_index="2" enum_const="Invalid Zoom Mode"/>
                                        <value_enum enum_index="3" enum_const="Invalid Zoom Value"/>
                                        <value_enum enum_index="4" enum_const="Invalid Focus Mode"/>
                                        <value_enum enum_index="5" enum_const="Invalid Focus Value"/>
                                        <value_enum enum_index="6" enum_const="Invalid White Balance"/>
                                        <value_enum enum_index="7" enum_const="Invalid Imaging Mode"/>
                                        <value_enum enum_index="8" enum_const="Invalid Exposure Mode"/>
                                        <value_enum enum_index="9" enum_const="Invalid Metering Mode"/>
                                        <value_enum enum_index="10" enum_const="Invalid Shutter Speed"/>
                                        <value_enum enum_index="11" enum_const="Invalid Aperture Value"/>
                                        <value_enum enum_index="12" enum_const="Invalid Light Sensitivity"/>
                                        <value_enum enum_index="13" enum_const="Invalid Image Stabilization"/>
                                        <value_enum enum_index="14" enum_const="Invalid Horizontal FOV"/>
                                        <value_enum enum_index="15" enum_const="Invalid Vertical FOV"/>
                                        <value_enum enum_index="16" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="DigitalVideoSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="DigitalVideoErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Minimum Bit Rate"/>
                                        <value_enum enum_index="2" enum_const="Invalid Maximum Bit Rate"/>
                                        <value_enum enum_index="3" enum_const="Requested Frame Rate Too Low"/>
                                        <value_enum enum_index="4" enum_const="Requested Frame Rate Too High"/>
                                        <value_enum enum_index="5" enum_const="Invalid Frame Size"/>
                                        <value_enum enum_index="6" enum_const="Invalid Format"/>
                                        <value_enum enum_index="7" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="AnalogVideoSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="AnalogVideoErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Format"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="StillImageSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
```

```
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="StillImageErrorCode" optional="false" field_type="unsigned byte"
field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid Frame Size"/>
                                            <value_enum enum_index="2" enum_const="Invalid Format"/>
                                            <value_enum enum_index="3" enum_const="Multiple Invalid Parameters"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="H264VideoEncodingErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="H264VideoEncodingErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid Profile"/>
                                            <value_enum enum_index="2" enum_const="Invalid Preset"/>
                                            <value_enum enum_index="3" enum_const="Invalid GroupOfPictures"/>
                                            <value_enum enum_index="4" enum_const="Invalid GDR"/>
                                            <value_enum enum_index="5" enum_const="Invalid RegionOfInterest"/>
                                            <value_enum enum_index="6" enum_const="Invalid IntraMacroRefresh"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="DigitalAudioSensorErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="DigitalAudioSensorErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid BitRate"/>
                                            <value_enum enum_index="2" enum_const="Invalid Format"/>
                                            <value_enum enum_index="3" enum_const="Invalid SampleRate"/>
                                            <value_enum enum_index="4" enum_const="Invalid BitDepth"/>
                                            <value_enum enum_index="5" enum_const="Invalid EncodingQuality"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="DigitalAudioOutputErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="DigitalAudioOutputErrorCode" optional="false"
```

```
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Stream Not Found"/>
                                            <value_enum enum_index="2" enum_const="Stream Not Supported"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="VideoIlluminatorErrorRec" optional="false">
                                    <fixed_field name="IlluminatorID" optional="false" field_type="unsigned short
integer" field_units="one"/>
                                    <fixed_field name="VideoIlluminatorErrorRecCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Unsupported Mode"/>
                                            <value_enum enum_index="2" enum_const="Unsupported Beam Width"/>
                                            <value_enum enum_index="3" enum_const="Unsupported Beam Height"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                            </variant>
                        </list>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="ReportH264VideoEncodingCapabilities" message_id="480E" is_command="false">
                <description xml:space="preserve">This message is used to report which configuration options are
supported on one or more H264 encoded video streams.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="H264CapabilitiesList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="H264CapabilitiesRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned byte"/>
                            <fixed_field name="SensorID" optional="false" interpretation="Note: Zero is not a valid
VisualSensorID value" field_type="unsigned short integer" field_units="one"/>
                            <bit_field name="SupportedProfiles" optional="true" interpretation="A high value (1) for a
bit means the profile is supported." field_type_unsigned="unsigned integer">
                                <sub_field name="Baseline">
                                    <bit_range from_index="0" to_index="0" interpretation="Baseline"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="ConstrainedBaseline">
                                    <bit_range from_index="1" to_index="1" interpretation="ConstrainedBaseline"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="Main">
                                    <bit_range from_index="2" to_index="2" interpretation="Main"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="Extended">
                                    <bit_range from_index="3" to_index="3" interpretation="Extended"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
```

```
    </value_set>
</sub_field>
<sub_field name="High">
    <bit_range from_index="4" to_index="4" interpretation="High"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="ProgressiveHigh">
    <bit_range from_index="5" to_index="5" interpretation="ProgressiveHigh"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="ConstrainedHigh">
    <bit_range from_index="6" to_index="6" interpretation="ConstrainedHigh"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="High10">
    <bit_range from_index="7" to_index="7" interpretation="High10"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="High422">
    <bit_range from_index="8" to_index="8" interpretation="High422"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="High444Predictive">
    <bit_range from_index="9" to_index="9" interpretation="High444Predictive"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="High10Intra">
    <bit_range from_index="10" to_index="10" interpretation="High10Intra"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="High422Intra">
    <bit_range from_index="11" to_index="11" interpretation="High422Intra"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="High444Intra">
    <bit_range from_index="12" to_index="12" interpretation="High444Intra"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="CAVLC444">
    <bit_range from_index="13" to_index="13" interpretation="CAVLC444"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="ScalableBaseline">
    <bit_range from_index="14" to_index="14" interpretation="ScalableBaseline"/>
    <value_set offset_to_lower_limit="false">
        <value_enum enum_index="0" enum_const="NotSupported"/>
        <value_enum enum_index="1" enum_const="Supported"/>
    </value_set>
</sub_field>
<sub_field name="ScalableConstrainedBaseline">
    <bit_range from_index="15" to_index="15"
```

```
interpretation="ScalableConstrainedBaseline"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="ScalableHigh">
                                    <bit_range from_index="16" to_index="16" interpretation="ScalableHigh"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="ScalableConstrainedHigh">
                                    <bit_range from_index="17" to_index="17" interpretation="ScalableConstrainedHigh"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="ScalableHighIntra">
                                    <bit_range from_index="18" to_index="18" interpretation="ScalableHighIntra"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="StereoHigh">
                                    <bit_range from_index="19" to_index="19" interpretation="StereoHigh"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="MultiviewHigh">
                                    <bit_range from_index="20" to_index="20" interpretation="MultiviewHigh"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                            </bit_field>
                            <bit_field name="SupportedLevels" optional="true" interpretation="A high value (1) for a
bit means the level is supported." field_type_unsigned="unsigned integer">
                                <sub_field name="Level_1">
                                    <bit_range from_index="0" to_index="0" interpretation="Level_1"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="Level_1B">
                                    <bit_range from_index="1" to_index="1" interpretation="Level_1B"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="Level_11">
                                    <bit_range from_index="2" to_index="2" interpretation="Level_11"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="Level_12">
                                    <bit_range from_index="3" to_index="3" interpretation="Level_12"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="Level_13">
                                    <bit_range from_index="4" to_index="4" interpretation="Level_13"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="NotSupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="Level_2">
                                    <bit_range from_index="5" to_index="5" interpretation="Level_2"/>
```

```
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_21">
        <bit_range from_index="6" to_index="6" interpretation="Level_21"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_22">
        <bit_range from_index="7" to_index="7" interpretation="Level_22"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_3">
        <bit_range from_index="8" to_index="8" interpretation="Level_3"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_31">
        <bit_range from_index="9" to_index="9" interpretation="Level_31"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_32">
        <bit_range from_index="10" to_index="10" interpretation="Level_32"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_4">
        <bit_range from_index="11" to_index="11" interpretation="Level_4"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_41">
        <bit_range from_index="12" to_index="12" interpretation="Level_41"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_42">
        <bit_range from_index="13" to_index="13" interpretation="Level_42"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_5">
        <bit_range from_index="14" to_index="14" interpretation="Level_5"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_51">
        <bit_range from_index="15" to_index="15" interpretation="Level_51"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
    </sub_field>
    <sub_field name="Level_52">
        <bit_range from_index="16" to_index="16" interpretation="Level_52"/>
        <value_set offset_to_lower_limit="false">
            <value_enum enum_index="0" enum_const="NotSupported"/>
            <value_enum enum_index="1" enum_const="Supported"/>
        </value_set>
```

```
                    </sub_field>
                    <sub_field name="Level_6">
                        <bit_range from_index="17" to_index="17" interpretation="Level_6"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="Level_61">
                        <bit_range from_index="18" to_index="18" interpretation="Level_61"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="Level_62">
                        <bit_range from_index="19" to_index="19" interpretation="Level_62"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                </bit_field>
                <bit_field name="SupportedPresets" optional="true" interpretation="A high value (1) for a
bit means the preset is supported." field_type_unsigned="unsigned short integer">
                    <sub_field name="LowLatency">
                        <bit_range from_index="0" to_index="0" interpretation="LowLatency"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="SlowComms">
                        <bit_range from_index="1" to_index="1" interpretation="SlowComms"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="PersistentStare">
                        <bit_range from_index="2" to_index="2" interpretation="PersistentStare"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="DriveVision">
                        <bit_range from_index="3" to_index="3" interpretation="DriveVision"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="Manipulation">
                        <bit_range from_index="4" to_index="4" interpretation="Manipulation"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="BestQuality">
                        <bit_range from_index="5" to_index="5" interpretation="BestQuality"/>
                        <value_set offset_to_lower_limit="false">
                            <value_enum enum_index="0" enum_const="NotSupported"/>
                            <value_enum enum_index="1" enum_const="Supported"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="Reserved">
                        <bit_range from_index="6" to_index="11" interpretation="Reserved"/>
                        <value_set offset_to_lower_limit="false">
                            <value_range lower_limit="0" lower_limit_type="inclusive" upper_limit="63"
upper_limit_type="inclusive"/>
                        </value_set>
                    </sub_field>
                    <sub_field name="ProgramSpecific">
                        <bit_range from_index="12" to_index="15" interpretation="ProgramSpecific"/>
                        <value_set offset_to_lower_limit="false">
                            <value_range lower_limit="0" lower_limit_type="inclusive" upper_limit="15"
upper_limit_type="inclusive"/>
                        </value_set>
                    </sub_field>
```

```
                        </bit_field>
                        <fixed_field name="GroupOfPicturesSupported" optional="true" interpretation="Boolean for
Group of Pictures support" field_type="unsigned byte" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="FALSE"/>
                                <value_enum enum_index="1" enum_const="TRUE"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="GradualDecoderRefreshSupported" optional="true" interpretation="Boolean
for Gradual Decoder Refresh (Periodic Intra Refresh) support" field_type="unsigned byte" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="FALSE"/>
                                <value_enum enum_index="1" enum_const="TRUE"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="RegionOfInterestSupported" optional="true" interpretation="Boolean for
Region of Interest (foveation) support" field_type="unsigned byte" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="FALSE"/>
                                <value_enum enum_index="1" enum_const="TRUE"/>
                            </value_set>
                        </fixed_field>
                    </record>
                </list>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="ReportH264VideoEncodingConfiguration" message_id="480D" is_command="false">
            <description xml:space="preserve">This message is used to report the active configuration for one or
more H264 encoded video streams.</description>
            <header name="JAUSApplicationLayerHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="H264ConfigurationList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <sequence name="H264ConfigurationSequence" optional="false">
                        <presence_vector field_type_unsigned="unsigned byte"/>
                        <record name="H264ConfigurationRec" optional="false">
                            <presence_vector field_type_unsigned="unsigned byte"/>
                            <fixed_field name="SensorID" optional="false" interpretation="Note: Zero is not a valid
VisualSensorID value" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="Profile" optional="true" interpretation="Enumeration for h264
profile setting." field_type="unsigned byte" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Baseline"/>
                                    <value_enum enum_index="1" enum_const="ConstrainedBaseline"/>
                                    <value_enum enum_index="2" enum_const="Main"/>
                                    <value_enum enum_index="3" enum_const="Extended"/>
                                    <value_enum enum_index="4" enum_const="High"/>
                                    <value_enum enum_index="5" enum_const="ProgressiveHigh"/>
                                    <value_enum enum_index="6" enum_const="ConstrainedHigh"/>
                                    <value_enum enum_index="7" enum_const="High10"/>
                                    <value_enum enum_index="8" enum_const="High422"/>
                                    <value_enum enum_index="9" enum_const="High444Predictive"/>
                                    <value_enum enum_index="10" enum_const="High10Intra"/>
                                    <value_enum enum_index="11" enum_const="High422Intra"/>
                                    <value_enum enum_index="12" enum_const="High444Intra"/>
                                    <value_enum enum_index="13" enum_const="CAVLC444"/>
                                    <value_enum enum_index="14" enum_const="ScalableBaseline"/>
                                    <value_enum enum_index="15" enum_const="ScalableConstrainedBaseline"/>
                                    <value_enum enum_index="16" enum_const="ScalableHigh"/>
                                    <value_enum enum_index="17" enum_const="ScalableConstrainedHigh"/>
                                    <value_enum enum_index="18" enum_const="ScalableHighIntra"/>
                                    <value_enum enum_index="19" enum_const="StereoHigh"/>
                                    <value_enum enum_index="20" enum_const="MultiviewHigh"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="Level" optional="true" interpretation="Enumeration for target h264
level." field_type="unsigned integer" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Level_1"/>
                                    <value_enum enum_index="1" enum_const="Level_1B"/>
                                    <value_enum enum_index="2" enum_const="Level_11"/>
                                    <value_enum enum_index="3" enum_const="Level_12"/>
                                    <value_enum enum_index="4" enum_const="Level_13"/>
                                    <value_enum enum_index="5" enum_const="Level_2"/>
                                    <value_enum enum_index="6" enum_const="Level_21"/>
```

```
                                    <value_enum enum_index="7" enum_const="Level_22"/>
                                    <value_enum enum_index="8" enum_const="Level_3"/>
                                    <value_enum enum_index="9" enum_const="Level_31"/>
                                    <value_enum enum_index="10" enum_const="Level_32"/>
                                    <value_enum enum_index="11" enum_const="Level_4"/>
                                    <value_enum enum_index="12" enum_const="Level_41"/>
                                    <value_enum enum_index="13" enum_const="Level_42"/>
                                    <value_enum enum_index="14" enum_const="Level_5"/>
                                    <value_enum enum_index="15" enum_const="Level_51"/>
                                    <value_enum enum_index="16" enum_const="Level_52"/>
                                    <value_enum enum_index="17" enum_const="Level_6"/>
                                    <value_enum enum_index="18" enum_const="Level_61"/>
                                    <value_enum enum_index="19" enum_const="Level_62"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="Preset" optional="true" interpretation="Preconfigured settings for
hardware specific H264 settings.  The actual values for each preset are not specified by the standard and are up to the
implementation." field_type="unsigned byte" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="LowLatency"/>
                                    <value_enum enum_index="1" enum_const="SlowComms"/>
                                    <value_enum enum_index="2" enum_const="PersistentStare"/>
                                    <value_enum enum_index="3" enum_const="DriveVision"/>
                                    <value_enum enum_index="4" enum_const="Manipulation"/>
                                    <value_enum enum_index="5" enum_const="BestQuality"/>
                                    <value_enum enum_index="12" enum_const="ProgramSpecific_1"/>
                                    <value_enum enum_index="13" enum_const="ProgramSpecific_2"/>
                                    <value_enum enum_index="14" enum_const="ProgramSpecific_3"/>
                                    <value_enum enum_index="15" enum_const="ProgramSpecific_4"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="GroupOfPictures" optional="true" interpretation="Number of interim
frames before sending a key frame" field_type="unsigned byte" field_units="one"/>
                            <fixed_field name="GradualDecoderRefreshOnOff" optional="true" interpretation="Turn
on/off Gradual Decoder Refresh, also called Periodic Intra Refresh." field_type="unsigned byte" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="OFF"/>
                                    <value_enum enum_index="1" enum_const="ON"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="RegionOfInterestOnOff" optional="true" interpretation="Turn on/off
region of interest (foveation) encoding, if supported.  The region of interest is specified by the
RegionOfInterestParams array." field_type="unsigned byte" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="OFF"/>
                                    <value_enum enum_index="1" enum_const="ON"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="IntraMacroRefresh" optional="true" interpretation="This parameter is
the random intra macroblock refresh; it provides the refresh rate in Macroblocks per picture." field_type="unsigned
short integer" field_units="one">
                                <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                            </fixed_field>
                        </record>
                        <record name="RegionOfInterestParams" optional="true">
                            <fixed_field name="X_Start" optional="false" interpretation="Horizontal start location
of high bitrate window, measured as percent of total image width from the left" field_type="unsigned short integer"
field_units="percent">
                                <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="Y_Start" optional="false" interpretation="Vertical start location of
high bitrate window, measured as percent of total image height from the top" field_type="unsigned short integer"
field_units="percent">
                                <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="Width" optional="false" interpretation="Width of high bitrate
window, measured as percent of total image width" field_type="unsigned short integer" field_units="percent">
                                <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="Height" optional="false" interpretation="Height of high bitrate
window , measured as percent of total image height" field_type="unsigned short integer" field_units="percent">
                                <scale_range real_lower_limit="0" real_upper_limit="100" integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="HighBitRate" optional="false" interpretation="Bitrate of ROI window,
measured in kilobits per second" field_type="unsigned short integer" field_units="one"/>
                            <fixed_field name="LowBitRate" optional="false" interpretation="Bitrate of ROI window,
measured in kilobits per second" field_type="unsigned short integer" field_units="one"/>
                        </record>
                        <record name="GradualDecoderRefreshParams" optional="true">
                            <fixed_field name="GDR_Duration" optional="false" interpretation="The number of frames
needed to completely refresh the image" field_type="unsigned short integer" field_units="one"/>
```

```
                                <fixed_field name="GDR_Interval" optional="false" interpretation="The number of frames
that need to be transmitted before starting a new GDR cycle. Note that the interval must be greater than or equal to
the duration" field_type="unsigned short integer" field_units="one"/>
                            </record>
                        </sequence>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
        </output_set>
    </message_set>
    <internal_events_set/>
    <protocol_behavior is_stateless="false">
        <start state_machine_name="digitalVideo.visualSensor.accessControl.events.transport.ReceiveFSM"
state_name="Receiving.Ready.NotControlled"/>
        <start state_machine_name="digitalVideo.visualSensor.accessControl.events.transport.SendFSM"
state_name="Sending"/>
        <state_machine name="digitalVideo.visualSensor.accessControl.events.transport.ReceiveFSM">
            <state name="Receiving" initial_state="Ready">
                <state name="Ready" initial_state="NotControlled">
                    <state name="NotControlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="digitalVideo.visualSensor.accessControl.events.transport.Receive">
                                <parameter type="QueryH264VideoEncodingCapabilities" value="msg"
interpretation="enveloped query message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportH264VideoEncodingCapabilities" interpretation="Send a
ReportH264VideoEncodingCapabilities message to the original requestor">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="digitalVideo.visualSensor.accessControl.events.transport.Receive">
                                <parameter type="QueryH264VideoEncodingConfiguration" value="msg"
interpretation="enveloped query message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportH264VideoEncodingConfiguration" interpretation="Send a
ReportH264VideoEncodingConfiguration message to the original requestor">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                            </state>
                    <state name="Controlled" initial_state="Available">
                        <state name="Available">
                                                </state>
                        <state name="NotAvailable">
                                                </state>
                        <default_state>
                            <transition name="digitalVideo.visualSensor.accessControl.events.transport.Receive">
                                <parameter type="SetH264VideoEncodingConfiguration" value="msg"
interpretation="enveloped set message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <guard condition="isControllingClient( transportData )" interpretation="True if the
command message was received from the client currently controlling this component"/>
                                <internal/>
                                <action name="setH264VideoEncodingConfig" interpretation="Update the H264 encoding
configuration if valid for the specified sensor ID">
                                    <argument value="msg"/>
                                </action>
                                <action name="sendConfirmSensorConfiguration" interpretation="Send
sendConfirmSensorConfiguration message with confirmation or error code for each specified sensor ID">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="digitalVideo.visualSensor.accessControl.events.transport.Receive">
                                <parameter type="QueryH264VideoEncodingCapabilities" value="msg"
interpretation="enveloped query message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
```

```
                                <internal/>
                                <action name="sendReportH264VideoEncodingCapabilities" interpretation="Send a
ReportH264VideoEncodingCapabilities message to the original requestor">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                            <transition name="digitalVideo.visualSensor.accessControl.events.transport.Receive">
                                <parameter type="QueryH264VideoEncodingConfiguration" value="msg"
interpretation="enveloped query message"/>
                                <parameter type="Receive.Body.ReceiveRec" value="transportData"
interpretation="transport data"/>
                                <internal/>
                                <action name="sendReportH264VideoEncodingConfiguration" interpretation="Send a
ReportH264VideoEncodingConfiguration message to the original requestor">
                                    <argument value="msg"/>
                                    <argument value="transportData"/>
                                </action>
                            </transition>
                        </default_state>
                                            </state>
                    <default_state>
                                            </default_state>
                                    </state>
                            </state>
                                </state_machine>
        <state_machine name="digitalVideo.visualSensor.accessControl.events.transport.SendFSM">
            <state name="Sending">
                            </state>
                                </state_machine>
    </protocol_behavior>
</service_def>
```

A.8 PORTMAPPER

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="PortMapper" id="urn:jaus:jss:environmentSensing:PortMapper" version="2.0" xmlns="urn:jaus:jsidl:1.1"
xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">A port mapping service is required to support streaming of data from a device
residing on a different subnet. For example, a digital resource such as a video or audio stream described by an RTSP
endpoint may be on a network internal to the host platform. However, a controller or other client may reside on a
different external network. The Port Mapper Service allows for the bridging of these two networks, such that the stream
is accessible on the external facing network. Effectively, the stream source uses to the Port Mapper to request port
forwarding, such that any traffic received on the external facing IP address and port is redirected by the service
implementation to the stream host on the internal IP address and port. The stream source can then use the external IP
and port in its discovery registration process. Note that the implementation of the Port Mapper must have access to
both networks.</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost, or reordered.</assumptions>
    <references>
        <inherits_from name="events" id="urn:jaus:jss:core:Events" version="1.1"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="CancelPortMapping" message_id="080B" is_command="false">
                <description xml:space="preserve">This message is sent to cancel any mapping to the endpoint specified
in the message.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="RequestPortMappingRec" optional="false">
                        <fixed_field name="SocketType" optional="false" field_type="unsigned byte" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="UDP"/>
                                <value_enum enum_index="1" enum_const="TCP"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="IPAddr" optional="false" interpretation="IPv4 address of endpoint port to
which PortMapper will map, in network byte order" field_type="unsigned integer" field_units="one"/>
                        <fixed_field name="Port" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </body>
                <footer name="Footer"/>
```

```
            </message_def>
            <message_def name="RequestPortMapping" message_id="080C" is_command="false">
                <description xml:space="preserve">This message is sent to request a port mapping at the network gateway
(node hosting the PortMapper) such that network packets arriving on the gateway's mapped port are forwarded to the
endpoint specified in this request.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <record name="RequestPortMappingRec" optional="false">
                        <fixed_field name="SocketType" optional="false" field_type="unsigned byte" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="UDP"/>
                                <value_enum enum_index="1" enum_const="TCP"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="IPAddr" optional="false" interpretation="IPv4 address of endpoint port to
which PortMapper will map, in network byte order" field_type="unsigned integer" field_units="one"/>
                        <fixed_field name="Port" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryPortMappings" message_id="2815" is_command="false">
                <description xml:space="preserve">This message is sent to request a list of all current
mappings</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body"/>
                <footer name="Footer"/>
            </message_def>
        </input_set>
        <output_set>
            <message_def name="GrantPortMapping" message_id="080E" is_command="false">
                <description xml:space="preserve">The GrantPortMapping message is sent in response to the
RequestPortMapping message. GrantPortMapping contains both the requested endpoint that was specified in the request,
and the mapped endpoint on the external-facing network.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="GrantPortMappingSeq" optional="false">
                        <record name="RequestPortMappingRec" optional="false">
                            <fixed_field name="SocketType" optional="false" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="UDP"/>
                                    <value_enum enum_index="1" enum_const="TCP"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="IPAddr" optional="false" interpretation="IPv4 address of endpoint port
to which PortMapper will map, in network byte order" field_type="unsigned integer" field_units="one"/>
                            <fixed_field name="Port" optional="false" field_type="unsigned short integer"
field_units="one"/>
                        </record>
                        <record name="GrantPortMappingRec" optional="false">
                            <fixed_field name="IPAddr" optional="false" interpretation="IPv4 address of client side
mapping of endpoint, in network byte order" field_type="unsigned integer" field_units="one"/>
                            <fixed_field name="Port" optional="false" field_type="unsigned short integer"
field_units="one"/>
                        </record>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="ReportPortMappings" message_id="4815" is_command="false">
                <description xml:space="preserve">This message is includes a list of all current
mappings.</description>
                <header name="JAUSApplicationLayerHeader">
                    <record name="HeaderRec" optional="false">
```

```
                        <fixed_field name="MessageID" optional="false" field_type="unsigned short integer"
field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="MappingSeq" optional="false">
                        <record name="RequestPortMappingRec" optional="false">
                            <fixed_field name="SocketType" optional="false" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="UDP"/>
                                    <value_enum enum_index="1" enum_const="TCP"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="IPAddr" optional="false" interpretation="IPv4 address of endpoint port
to which PortMapper will map, in network byte order" field_type="unsigned integer" field_units="one"/>
                            <fixed_field name="Port" optional="false" field_type="unsigned short integer"
field_units="one"/>
                        </record>
                        <record name="GrantPortMappingRec" optional="false">
                            <fixed_field name="IPAddr" optional="false" interpretation="IPv4 address of client side
mapping of endpoint, in network byte order" field_type="unsigned integer" field_units="one"/>
                            <fixed_field name="Port" optional="false" field_type="unsigned short integer"
field_units="one"/>
                        </record>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
        </output_set>
    </message_set>
    <internal_events_set/>
    <protocol_behavior is_stateless="false">
        <start state_machine_name="events.transport.ReceiveFSM" state_name="Receiving.Ready"/>
        <start state_machine_name="events.transport.SendFSM" state_name="Sending"/>
        <state_machine name="events.transport.ReceiveFSM">
            <state name="Receiving" initial_state="Ready">
                <state name="Ready">
                    <default_state>
                        <transition name="events.transport.Receive">
                            <parameter type="RequestPortMapping" value="msg"/>
                            <parameter type="Receive.Body.ReceiveRec" value="transportData"/>
                            <internal/>
                            <action name="createMappedPort" interpretation="Assign an external address and port to the
endpoint specified in the message and begin forwarding traffic received on that port">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                            <action name="sendGrantPortMapping" interpretation="Send a Grant Port Mapping message to
the requesting client">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                        </transition>
                        <transition name="events.transport.Receive">
                            <parameter type="CancelPortMapping" value="msg"/>
                            <parameter type="Receive.Body.ReceiveRec" value="transportData"/>
                            <internal/>
                            <action name="cancelMappedPort" interpretation="Remove the specified mapping from the list
of active maps">
                                <argument value="msg"/>
                            </action>
                        </transition>
                        <transition name="events.transport.Receive">
                            <parameter type="QueryPortMappings" value="msg"/>
                            <parameter type="Receive.Body.ReceiveRec" value="transportData"/>
                            <internal/>
                            <action name="sendReportPortMappings" interpretation="Send a Report Port Mappings message
to the requesting client">
                                <argument value="msg"/>
                                <argument value="transportData"/>
                            </action>
                        </transition>
                    </default_state>
                                </state>
                            </state>
                                </state_machine>
        <state_machine name="events.transport.SendFSM">
            <state name="Sending">
                            </state>
                                </state_machine>
```

```
    </protocol_behavior>
</service_def>
```

5.2 RangeSensor

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<service_def name="RangeSensor" id="urn:jaus:jss:environmentSensing:RangeSensor" version="2.0"
xmlns="urn:jaus:jsidl:1.1" xmlns:ns2="urn:jaus:jsidl:plus">
    <description xml:space="preserve">The function of the Range Sensor Service is to provide
  information from proximity sensors. This service will output the location of various
  Data Points with a certain measure of accuracy. A given Range Sensor service may be
  comprised of one to many actual physical sensors or technologies. Each sub-sensor
  can be assigned (by the developer) a unique Sensor ID. When appropriate, the
  reserved Sensor ID of 0 may be used to refer to all sensors attached to a given
  Range Sensor Service.
  The Data Points are measured in the sensor's native coordinate system and are
  expressed in terms of range, bearing and inclination. Range is the distance, in
  meters, along the line from the origin of the sensor's native coordinate system
  (sensor's origin) to the specified point. Bearing is the angle, in radians, that
  the line from the sensor's origin to the specified point makes about the sensor's
  z-axis in the right handed sense (Figure 2). Inclination is the the angle, in
  radians, that the line from the sensor origin to the specified point makes about
  the sensor's y-axis in the right handed sense (Figure 2).
  Each data point has an optional ID parameter. This parameter is provided for
  those sensor technologies which may assign and/or track entities based on unique
  ID values; however, such tracking capabilities are not required for a compliant
  Range Sensor Service. The behavior of the data point ID is not specified, i.e.,
  IDs may repeat in a given report and IDs may persist from one report to another.
  No semantic value should be placed on the ID values in a generalized way. Data
  Point ID behavior should be derived from the underlying sensor or algorithm
  technology and is mearly provided to be used in those situations where mutiple
  parties can agree upon the behavior and semantics of the ID values.
  Data from the range sensor can be reported in both a compressed and uncompressed
  format, different query and report messages are provided for each exchange and
  the kind of data compression supported by the service is reported in the
  Report Range Sensor Capabilities message. Requests for unsupported data
  compression algorithms will result in the generation of a Report Sensor Error
  message indicating an unsupported compression request.
  The range sensor can express the bearing, inclination and range terms with respect
  to either its native coordinate system or the vehicle coordinate system if coordinate
  transforms are supported. The Query Sensor Geometric Properties message can be
  used to determine the geometric relationship between the sensor and the vehicle
  coordinate system. Three possible coordinate responses are possible: (a)
  the service does not know the sensor's position, (b) the sensor coordinate
  system is fixed with respect to the vehicle and (c) the sensor is attached
  to some manipulator. These cases are supported by the Report Sensor Geometric
  Properties message and are described therein.</description>
    <assumptions xml:space="preserve">Messages may be delayed, lost, or reordered.</assumptions>
    <references>
        <inherits_from name="accessControl" id="urn:jaus:jss:core:AccessControl" version="1.1"/>
    </references>
    <message_set>
        <input_set>
            <message_def name="SetRangeSensorConfiguration" message_id="0802" is_command="true">
                <description xml:space="preserve">This message is used to set the range sensors' current configuration.
Configuration is based off of each range sensor's capabilities as described in the Report Range Sensor Capabilities
message. This message shall cause the receiving service to reply to the sender with a Confirm Range Sensor
Configuration message. If the configuration specified is invalid for a given sensor ID, the confirm message shall
contain an Range Sensor Error Record for the given Sensor ID however other, valid, configurations specified shall be
set (if they exist).</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <sequence name="RangeSensorConfigurationSequence" optional="false">
                        <record name="RequestIdRec" optional="false">
                            <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                        </record>
                        <list name="RangeSensorConfigurationList" optional="false">
                            <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                            <record name="RangeSensorConfigurationRec" optional="false">
                                <presence_vector field_type_unsigned="unsigned short integer"/>
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is
```

```
not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="HorizontalFieldOfViewStartAngle" optional="true"
field_type="unsigned integer" field_units="radian">
                                    <scale_range real_lower_limit="-3.141592653589793"
real_upper_limit="3.141592653589793" integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="HorizontalFieldOfViewStopAngle" optional="true" field_type="unsigned
integer" field_units="radian">
                                    <scale_range real_lower_limit="-3.141592653589793"
real_upper_limit="3.141592653589793" integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="VerticalFieldOfViewStartAngle" optional="true" field_type="unsigned
integer" field_units="radian">
                                    <scale_range real_lower_limit="-3.141592653589793"
real_upper_limit="3.141592653589793" integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="VerticalFieldOfViewStopAngle" optional="true" field_type="unsigned
integer" field_units="radian">
                                    <scale_range real_lower_limit="-3.141592653589793"
real_upper_limit="3.141592653589793" integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="UpdateRate" optional="true" field_type="unsigned short integer"
field_units="hertz">
                                    <scale_range real_lower_limit="0" real_upper_limit="1000"
integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="MinimumRange" optional="true" field_type="unsigned integer"
field_units="meter">
                                    <scale_range real_lower_limit="0" real_upper_limit="1000000.0"
integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="MaximumRange" optional="true" field_type="unsigned integer"
field_units="meter">
                                    <scale_range real_lower_limit="0" real_upper_limit="1000000.0"
integer_function="round"/>
                                </fixed_field>
                                <fixed_field name="SensorState" optional="true" field_type="unsigned byte"
field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Active"/>
                                        <value_enum enum_index="1" enum_const="Standby"/>
                                        <value_enum enum_index="2" enum_const="Off"/>
                                    </value_set>
                                </fixed_field>
                            </record>
                        </list>
                    </sequence>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QuerySensorGeometricProperties" message_id="2805" is_command="false">
                <description xml:space="preserve">This message shall cause the
   receiving component to reply to the requestor with a Report
   Sensor Geometric Properties message.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="SensorIdList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="SensorIDQueryRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </body>
```

```
            <footer name="Footer"/>
        </message_def>
        <message_def name="QueryRangeSensorCompressedData" message_id="2804" is_command="false">
            <description xml:space="preserve">This message shall cause the
   receiving service to reply to the requestor with a Report Range
   Sensor Compressed Data 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. The third field specifies which coordinate system
   the data should be reported in, either the sensor's native
   coordinate system or, if supported, a coordinate system specified
   by a Set Specified Sensor Coordinate System message. The fourth
   field specifies the data compression algorithm. Three compression
   algorithms are supported: DEFLATE, bzip2 and LZMA.</description>
            <header name="AppHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="QueryRangeSensorCompressedDataList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <record name="QueryRangeSensorCompressedDataRec" optional="false">
                        <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="AllSensors"/>
                                <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="ReportCoordinateSystem" optional="false" field_type="unsigned byte"
field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="Native Coordinate System"/>
                                <value_enum enum_index="1" enum_const="Vehicle Coordinate System"/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="DataCompression" optional="false" field_type="unsigned byte"
field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_enum enum_index="0" enum_const="None"/>
                                <value_enum enum_index="1" enum_const="DEFLATE"/>
                                <value_enum enum_index="2" enum_const="bzip2"/>
                                <value_enum enum_index="3" enum_const="LZMA "/>
                            </value_set>
                        </fixed_field>
                        <fixed_field name="QueryPresenceVector" optional="false" field_type="unsigned short
integer" field_units="one"/>
                    </record>
                </list>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="QueryRangeSensorData" message_id="2803" is_command="false">
            <description xml:space="preserve">This message shall cause the
   receiving service to reply to the requestor with a Report
   Range Sensor Data 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. The second field specifies which coordinate
   system the data should be reported in, either the sensor's
   native coordinate system or, if supported, a coordinate system
   specified by a Set Specified Sensor Coordinate System message.</description>
            <header name="AppHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="QueryRangeSensorDataList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <record name="QueryRangeSensorDataRec" optional="false">
                        <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
```

```
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="ReportCoordinateSystem" optional="false" field_type="unsigned byte"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Native Coordinate System"/>
                                    <value_enum enum_index="1" enum_const="Vehicle Coordinate System"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="QueryPresenceVector" optional="false" field_type="unsigned short
integer" field_units="one"/>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryRangeSensorCapabilities" message_id="2801" is_command="false">
                <description xml:space="preserve">This message shall cause the
   receiving service to reply to the requestor with a Report
   Range Sensor Capabilities 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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="RangeSensorCapabilitiesList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="QueryRangeSensorCapabilitiesRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
                            <fixed_field name="QueryPresenceVector" optional="false" field_type="unsigned short
integer" field_units="one"/>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="QueryRangeSensorConfiguration" message_id="2802" is_command="false">
                <description xml:space="preserve">This message shall cause the
   receiving service to reply to the requestor with a Report
   Range 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.</description>
                <header name="AppHeader">
                    <record name="HeaderRec" optional="false">
                        <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                    </record>
                </header>
                <body name="Body">
                    <list name="RangeSensorConfigurationList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <record name="QueryRangeSensorConfigurationRec" optional="false">
                            <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor being
queried. Zero is used to query all sensors associated with this service." field_type="unsigned short integer"
field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="AllSensors"/>
                                    <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive" interpretation="Specific ID to be queried"/>
                                </value_set>
                            </fixed_field>
```

```
                        <fixed_field name="QueryPresenceVector" optional="false" field_type="unsigned short
integer" field_units="one"/>
                    </record>
                </list>
            </body>
            <footer name="Footer"/>
        </message_def>
    </input_set>
    <output_set>
        <message_def name="ConfirmSensorConfiguration" message_id="0801" is_command="false">
            <description xml:space="preserve">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.</description>
            <header name="AppHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <sequence name="ConfirmSensorConfigurationSequence" optional="false">
                    <record name="RequestIdRec" optional="false">
                        <fixed_field name="RequestID" optional="false" field_type="unsigned byte"
field_units="one"/>
                    </record>
                    <list name="ConfirmSensorList" optional="false">
                        <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                        <variant name="ConfirmSensorConfigurationVariant" optional="false">
                            <vtag_field min_count="0" max_count="9" field_type_unsigned="unsigned byte"/>
                            <record name="SensorIdRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                            </record>
                            <record name="RangeSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="RangeSensorErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Horizontal Field of View"/>
                                        <value_enum enum_index="2" enum_const="Invalid Vertical Field of View"/>
                                        <value_enum enum_index="3" enum_const="Invalid Update Rate"/>
                                        <value_enum enum_index="4" enum_const="Invalid Sensor Range"/>
                                        <value_enum enum_index="5" enum_const="Invalid Sensor State"/>
                                        <value_enum enum_index="6" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="VisualSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="VisualSensorErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
```

```
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Sensor State"/>
                                        <value_enum enum_index="2" enum_const="Invalid Zoom Mode"/>
                                        <value_enum enum_index="3" enum_const="Invalid Zoom Value"/>
                                        <value_enum enum_index="4" enum_const="Invalid Focus Mode"/>
                                        <value_enum enum_index="5" enum_const="Invalid Focus Value"/>
                                        <value_enum enum_index="6" enum_const="Invalid White Balance"/>
                                        <value_enum enum_index="7" enum_const="Invalid Imaging Mode"/>
                                        <value_enum enum_index="8" enum_const="Invalid Exposure Mode"/>
                                        <value_enum enum_index="9" enum_const="Invalid Metering Mode"/>
                                        <value_enum enum_index="10" enum_const="Invalid Shutter Speed"/>
                                        <value_enum enum_index="11" enum_const="Invalid Aperture Value"/>
                                        <value_enum enum_index="12" enum_const="Invalid Light Sensitivity"/>
                                        <value_enum enum_index="13" enum_const="Invalid Image Stabilization"/>
                                        <value_enum enum_index="14" enum_const="Invalid Horizontal FOV"/>
                                        <value_enum enum_index="15" enum_const="Invalid Vertical FOV"/>
                                        <value_enum enum_index="16" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="DigitalVideoSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="DigitalVideoErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Minimum Bit Rate"/>
                                        <value_enum enum_index="2" enum_const="Invalid Maximum Bit Rate"/>
                                        <value_enum enum_index="3" enum_const="Requested Frame Rate Too Low"/>
                                        <value_enum enum_index="4" enum_const="Requested Frame Rate Too High"/>
                                        <value_enum enum_index="5" enum_const="Invalid Frame Size"/>
                                        <value_enum enum_index="6" enum_const="Invalid Format"/>
                                        <value_enum enum_index="7" enum_const="Multiple Invalid Parameters"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="AnalogVideoSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
                                <fixed_field name="AnalogVideoErrorCode" optional="false" field_type="unsigned
byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Invalid Format"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="StillImageSensorErrorRec" optional="false">
                                <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                    </value_set>
                                </fixed_field>
```

```
                                    <fixed_field name="StillImageErrorCode" optional="false" field_type="unsigned byte"
field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid Frame Size"/>
                                            <value_enum enum_index="2" enum_const="Invalid Format"/>
                                            <value_enum enum_index="3" enum_const="Multiple Invalid Parameters"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="H264VideoEncodingErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="H264VideoEncodingErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid Profile"/>
                                            <value_enum enum_index="2" enum_const="Invalid Preset"/>
                                            <value_enum enum_index="3" enum_const="Invalid GroupOfPictures"/>
                                            <value_enum enum_index="4" enum_const="Invalid GDR"/>
                                            <value_enum enum_index="5" enum_const="Invalid RegionOfInterest"/>
                                            <value_enum enum_index="6" enum_const="Invalid IntraMacroRefresh"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="DigitalAudioSensorErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="DigitalAudioSensorErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Invalid BitRate"/>
                                            <value_enum enum_index="2" enum_const="Invalid Format"/>
                                            <value_enum enum_index="3" enum_const="Invalid SampleRate"/>
                                            <value_enum enum_index="4" enum_const="Invalid BitDepth"/>
                                            <value_enum enum_index="5" enum_const="Invalid EncodingQuality"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                        </value_set>
                                    </fixed_field>
                                    <variable_length_string name="ErrorMessage" optional="false">
                                        <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                    </variable_length_string>
                                </record>
                                <record name="DigitalAudioOutputErrorRec" optional="false">
                                    <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor.
Zero is not a valid ID in this message." field_type="unsigned short integer" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_range lower_limit="1" lower_limit_type="inclusive"
upper_limit="65535" upper_limit_type="inclusive"/>
                                        </value_set>
                                    </fixed_field>
                                    <fixed_field name="DigitalAudioOutputErrorCode" optional="false"
field_type="unsigned byte" field_units="one">
                                        <value_set offset_to_lower_limit="false">
                                            <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                            <value_enum enum_index="1" enum_const="Stream Not Found"/>
                                            <value_enum enum_index="2" enum_const="Stream Not Supported"/>
                                            <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
```

```
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                            <record name="VideoIlluminatorErrorRec" optional="false">
                                <fixed_field name="IlluminatorID" optional="false" field_type="unsigned short
integer" field_units="one"/>
                                <fixed_field name="VideoIlluminatorErrorRecCode" optional="false"
field_type="unsigned byte" field_units="one">
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unknown Sensor ID"/>
                                        <value_enum enum_index="1" enum_const="Unsupported Mode"/>
                                        <value_enum enum_index="2" enum_const="Unsupported Beam Width"/>
                                        <value_enum enum_index="3" enum_const="Unsupported Beam Height"/>
                                        <value_enum enum_index="255" enum_const="Unknown Error / Fault"/>
                                    </value_set>
                                </fixed_field>
                                <variable_length_string name="ErrorMessage" optional="false">
                                    <count_field min_count="0" max_count="255" field_type_unsigned="unsigned
byte"/>
                                </variable_length_string>
                            </record>
                        </variant>
                    </list>
                </sequence>
            </body>
            <footer name="Footer"/>
        </message_def>
        <message_def name="ReportRangeSensorCapabilities" message_id="4801" is_command="false">
            <description xml:space="preserve">This message is used to report the range sensors' capabilities upon
receipt of a Query Range Sensor Capabilities message. Capabilities include both static sensor properties and valid
values and ranges for properties which can be modified by the Set Range Sensor Capabilities Message.</description>
            <header name="AppHeader">
                <record name="HeaderRec" optional="false">
                    <fixed_field name="MessageID" optional="false" interpretation="A two byte field to hold the
message ID of a message" field_type="unsigned short integer" field_units="one"/>
                </record>
            </header>
            <body name="Body">
                <list name="RangeSensorCapabilitiesList" optional="false">
                    <count_field min_count="0" max_count="65535" field_type_unsigned="unsigned short integer"/>
                    <record name="RangeSensorCapabilitiesRec" optional="false">
                        <presence_vector field_type_unsigned="unsigned short integer"/>
                        <fixed_field name="SensorID" optional="false" interpretation="ID of the sensor. Zero is not
a valid ID in this message." field_type="unsigned short integer" field_units="one">
                            <value_set offset_to_lower_limit="false">
                                <value_range lower_limit="1" lower_limit_type="inclusive" upper_limit="65535"
upper_limit_type="inclusive"/>
                            </value_set>
                        </fixed_field>
                        <variable_length_string name="SensorName" optional="false">
                            <count_field min_count="0" max_count="255" field_type_unsigned="unsigned byte"/>
                        </variable_length_string>
                        <bit_field name="SupportedStates" optional="true" field_type_unsigned="unsigned byte">
                            <sub_field name="Active">
                                <bit_range from_index="0" to_index="0"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Standby">
                                <bit_range from_index="1" to_index="1"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                            <sub_field name="Off">
                                <bit_range from_index="2" to_index="2"/>
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="Unsupported"/>
                                    <value_enum enum_index="1" enum_const="Supported"/>
                                </value_set>
                            </sub_field>
                        </bit_field>
                        <fixed_field name="MinimumHorizontalFieldOfViewStartAngle" optional="true"
field_type="unsigned integer" field_units="radian">
```

```
                                <scale_range real_lower_limit="-3.141592653589793" real_upper_limit="3.141592653589793"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MaximumHorizontalFieldOfViewStopAngle" optional="true"
field_type="unsigned integer" field_units="radian">
                                <scale_range real_lower_limit="-3.141592653589793" real_upper_limit="3.141592653589793"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MinimumVerticalFieldOfViewStartAngle" optional="true"
field_type="unsigned integer" field_units="radian">
                                <scale_range real_lower_limit="-3.141592653589793" real_upper_limit="3.141592653589793"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MaximumVerticalFieldOfViewStopAngle" optional="true"
field_type="unsigned integer" field_units="radian">
                                <scale_range real_lower_limit="-3.141592653589793" real_upper_limit="3.141592653589793"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MiniumumUpdateRate" optional="true" field_type="unsigned short integer"
field_units="hertz">
                                <scale_range real_lower_limit="0" real_upper_limit="1000" integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MaximumUpdateRate" optional="true" field_type="unsigned short integer"
field_units="hertz">
                                <scale_range real_lower_limit="0" real_upper_limit="1000" integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MinimumRange" optional="true" field_type="unsigned integer"
field_units="meter">
                                <scale_range real_lower_limit="0" real_upper_limit="1000000.0"
integer_function="round"/>
                            </fixed_field>
                            <fixed_field name="MaximumRange" optional="true" field_type="unsigned integer"
field_units="meter">
                                <scale_range real_lower_limit="0" real_upper_limit="1000000.0"
integer_function="round"/>
                            </fixed_field>
                            <bit_field name="SupportedCompression" optional="true" field_type_unsigned="unsigned byte">
                                <sub_field name="NoCompression">
                                    <bit_range from_index="0" to_index="0"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="DEFLATE">
                                    <bit_range from_index="1" to_index="1"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="bzip2">
                                    <bit_range from_index="2" to_index="2"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                                <sub_field name="LZMA">
                                    <bit_range from_index="3" to_index="3"/>
                                    <value_set offset_to_lower_limit="false">
                                        <value_enum enum_index="0" enum_const="Unsupported"/>
                                        <value_enum enum_index="1" enum_const="Supported"/>
                                    </value_set>
                                </sub_field>
                            </bit_field>
                            <fixed_field name="CoordinateTransformationSupported" optional="true" field_type="unsigned
byte" field_units="one">
                                <value_set offset_to_lower_limit="false">
                                    <value_enum enum_index="0" enum_const="False"/>
                                    <value_enum enum_index="1" enum_const="True"/>
                                </value_set>
                            </fixed_field>
                        </record>
                    </list>
                </body>
                <footer name="Footer"/>
            </message_def>
            <message_def name="ReportRangeSensorConfiguration" message_id="4802" is_command="false">
                <description xml:space="preserve">This message is sent in response to a Query Range Sensor
Configuration message. It is populated with the current sensor configuration (per sensor ID) as defined in the table
```