THE JOINT ARCHITECTURE FOR UNMANNED SYSTEMS

Reference Architecture Specification

Volume II, Part 3

Message Set

Version 3.3

June 27, 2007

TABLE OF CONTENTS

Title **Page**

TABLE OF FIGURES

Title **Page**

TABLE OF TABLES

Title **Page**

1. INTRODUCTION

This document, Part 3 of the Joint Architecture for Unmanned Systems (JAUS) Reference Architecture Specification, specifies the JAUS message set.

This part presents the details of command code usage for each message defined by JAUS. All command codes refer to Field #2 of the message header specified in Part 2. Messages that require additional data will specify the data and format in a table immediately following the description. Messages not requiring additional data present the command code and brief discussion only. Optional data fields in messages are indicated by the use of a presence vector (represented as a flag bit field). The mapping of the presence vector to the optional data fields follows the message definition table. A one (1) in the bit field indicates that the message field will be included in the message. A zero (0) in the bit field indicates that the message field will not be included in the message. Reserved bits in the Presence Vector shall be set to zero (0).

2. MESSAGE SET

2.1 Command Class

Command class messages are used to precipitate actions within a component.

2.1.1 Core Subgroup – Codes 0001-01FF

The following set of messages is defined for all JAUS components. If a component does not specify the implementation of a core message, then that core message can be ignored in implementations of that component. The Code is represented in hexadecimal notation with the letter h used as the hex indicator.

- Code 0001h: Set Component Authority
- Code 0002h: Shutdown
- Code 0003h: Standby
- Code 0004h: Resume
- Code 0005h: Reset
- Code 0006h: Set Emergency
- Code 0007h: Clear Emergency
- Code 0008h: Create Service Connection
- Code 0009h: Confirm Service Connection
- Code 000Ah: Activate Service Connection
- Code 000Bh: Suspend Service Connection
- Code 000Ch: Terminate Service Connection
- Code 000Dh: Request Component Control
- Code 000Eh: Release Component Control
- Code 000Fh: Confirm Component Control
- Code 0010h: Reject Component Control
- Code 0011h: Set Time
- Code 2001h: Query Component Authority
- Code 2002h: Query Component Status

- Code 2011h: Query Time
- Code 4001h: Report Component Authority
- Code 4002h: Report Component Status
- Code 4011h: Report Time

Code 0001h: Set Component Authority

This message shall set the command authority of the receiving component. The authority bits range in value from 0 to 255 with 255 being the highest.

The System Commander reserves the authority of 255. All Subsystem Commander components default to the authority of 8 and can be set higher by the System Commander. All other components default to the authority of 0 and can be changed by their respective Subsystem Commander component or by the system designer.

Field #	Name	Type	Units	Interpretation
1	Authority Code	Byte	N/A	Authority range 0-255

Code 0002h: Shutdown

This message shall cause the receiving component to free all of the resources allocated to its process by the system and then to shutdown.

Code 0003h: Standby

This message shall transition the receiving component to the Standby state. In the Standby state, the component shall cease internal functions until the Resume command is received. This command is only effective when the receiving component is in the Ready state.

Code 0004h: Resume

This message shall transition the receiving component out of the Standby state. This command is only effective when the receiving component is in Standby state.

Code 0005h: Reset

This message shall cause the receiving component to reinitialize.

Code 0006h: Set Emergency

This message shall alert the component to a safety critical situation. The component that sends the emergency command shall set the message priority to the safety critical priority range as described in Part 2. Receipt of the emergency command shall result in the component transitioning into the emergency state.

Field #	Name	Type	Units	Interpretation
1	Emergency Code	Unsigned Short Integer	N/A	Bit 0 = Perform transition to emergency state Bits 1 – 15 Reserved

Code 0007h: Clear Emergency

This message shall notify the receiving component that the current emergency condition is to be reset and that the component shall transition out of the emergency state, provided that all emergency conditions have been cleared.

JAUS currently defines only one emergency condition, the "Stop" condition. Future versions of this document could define other emergency conditions. If multiple emergency conditions exist, then all conditions that have been set must be specifically cleared before the component can transition out of the emergency state. This command is only effective when the receiving component is in Emergency state.

Field #	Name	Type	Units	Interpretation
1	Emergency Code	Unsigned Short Integer	N/A	Bit 0 = Stop Bits 1 – 15 Reserved

Code 0008h: Create Service Connection (DEPRECATE IN V4.0)

This message shall create a service connection (SC) between two components at the requested rate. The Requested Periodic Update Rate scaled integer limits are from 0 to 1092. A Requested Periodic Updated Rate of zero (0) is not valid. The range allowed is

once per minute (.016666667Hz) to 1092Hz. The Scaled Integer Lower and Upper Limits shown for the Requested Periodic Update Rate are for conversion purposes only.

SCs created for messages that support selection of optional data fields use field #3 to specify the presence vector. It is specified as an unsigned integer type, which is the maximum size for a presence vector. Unused bytes of the presence vector, as required by the command code in field #1, shall be set to zero.

Field #	Name	Type	Units	Interpretation
1	Command Code	Unsigned Short Integer	N/A	Command code of message to be sent on this SC
2	Requested Periodic Update Rate	Unsigned Short Integer	Hertz	Scaled Integer: Lower Limit = 0 Upper Limit = 1092
3	Presence Vector	Unsigned Integer	N/A	As defined by the specification of the command used in field #1

The command code and presence vector in the Create Service Connection message provide ample data to support the start of a Service Connection.

Code 0009h: Confirm Service Connection (DEPRECATE IN V4.0)

This message shall notify the SC creator of the result of the Create request. This message shall provide the creator with an instance ID of the specific SC being created; the confirmed periodic data update rate at which the SC is established, and a response code.

The instance ID shall distinguish this instance of the SC from other instances which might exist for the same message ID, but whose data content may differ due to the presence vector. Inform type SCs, which use the same command code and presence vector, shall assign the same instance ID. Command type SCs shall be always unique to a specific commander/provider pair of components.

The confirmed periodic data update rate informs the creator of the actual rate of the SC. Normally, the confirmed periodic data update rate of the SC will equal the requested rate. Performance and/or message transport issues (e.g. bandwidth) might prevent a

provider from establishing the SC at the requested rate. In this instance, the provider shall respond with a confirmed rate less than the requested rate. It is the responsibility of the creator to determine whether the confirmed periodic data update rate is sufficient. If the confirmed periodic data update rate is insufficient for proper and/or safe operation, then the creator shall terminate the connection.

The response code provides SC status information back to the creator. The response code shall be set as follows:

0 = SC created successfully

1 = Node does not support SCs.

2 = Component does not support SCs.

3 = Unused.

4 = Service Connection is refused due to internal component restrictions.

5 = The Create SC message used one or more invalid parameters.

6 = The Create SC Command is not supported by the receiving Component.

Field #	Name	Type	Units	Interpretation
1	Command Code	Unsigned Short Integer	N/A	Command code of message to be sent on this SC
2	Instance ID	Byte	N/A	The specific SC for the message indicated in field #1
3	Confirmed Periodic Update Rate	Unsigned Short Integer	Hertz	Scaled Integer: Lower Limit = 0 Upper Limit = 1092
4	Response Code	Byte	N/A	Bits 0 – 3: 0 = Successful 1 = Node not SC capable 2 = Component not SC capable 3 = Insufficient authority 4 = Connection refused 5 = Invalid create parameter(s) 6 = Command not supported Bits 4 – 7: Available

Code 000Ah: Activate Service Connection (DEPRECATE IN V4.0)

This message shall notify the provider that the SC should be activated. The SC commanders or the node only shall send this command. It has no effect if sent by Inform requesters. This message shall only be sent once a SC has been created and confirmed.

Field #	Name	Type	Units	Interpretation
1	Command Code	Unsigned Short Integer	N/A	Command code of message to be sent on this SC
2	Instance ID	Byte	N/A	The specific SC for the message indicated in field #1

Code 000Bh: Suspend Service Connection (DEPRECATE IN V4.0)

The SC creator, or the node, uses this message to notify the provider that the service provided on this connection shall be suspended. . This message shall only be sent once a SC has been created and confirmed.

Field #	Name	Type	Units	Interpretation
1	Command Code	Unsigned Short Integer	N/A	Command code of message to be sent on this SC
2	Instance ID	Byte	N/A	The specific SC for the message indicated in field #1

Code 000Ch: Terminate Service Connection (DEPRECATE IN V4.0)

The SC creator, or the node, uses this message to notify the service provider that the service connection shall be terminated. . This message shall only be sent once a SC has been created and confirmed. Once all components have requested a SC be terminated, the SC shall close and will not be available for future use. If the data from a closed SC is needed in the future, a new SC shall have to be created (see RA Part 2 for full description of SC protocol).

Field #	Name	Type	Units	Interpretation
1	Command Code	Unsigned Short Integer	N/A	Command code of message to be sent on this SC
2	Instance ID	Byte	N/A	The specific SC for the message indicated in field #1

Code 000Dh: Request Component Control

This message is used to request uninterruptible control of the receiving component. Once the uninterruptible control is established, the receiving component shall only execute commands from the sending component. The authority code parameter is to be set equal to that of the sending component. The receiving component must always accept the control of the highest authority component that is requesting uninterruptible control. Commands from all other components are ignored unless from a component with higher authority.

Field #	Name	Type	Units	Interpretation
1	Authority Code	Byte	N/A	Authority 0-255

Code 000Eh: Release Component Control

This message is used to relinquish uninterruptible control of the receiving component. . This message shall only be sent once control has been granted and shall be ignored if received from any component other than the component that has control. A component with higher authority can take control from a lower authority component by using the Code 000Dh: Request Component Control message.

Code 000Fh: Confirm Component Control

The Confirm Component Control message is used to notify a component that it accepts control from that component. When control has been granted, response code of 0, the component under control will only execute messages from the controlling component

until control is released or interrupted. When the requesting component has lower authority than the current controlling entity, the response will be 2, Control Not Accepted. For components not supporting interruptible control, the response code value of 1 can be used.

Field #	Name	Type	Units	Interpretation
1	Response Code	Byte	N/A	Bit 0 and 1: 0 = Control Accepted 1 = Uninterruptible control not supported 2 = Control Not Accepted Bits 2-7: Reserved

Code 0010h: Reject Component Control

The Reject Command message is used to notify a component that it does not accept control from that component. Control shall be rejected when a component with a higher authority requests control or when a service connection with a component with higher authority is active.

Code 0011h: Set Time

Time is configured within a JAUS system using the following message. Accuracy of the time may be dependent on latencies in the transmission of the message. Proper systems engineering procedures should be used to insure the accuracy of the time messages are within the system tolerance. All times are in Coordinated Universal Time (UTC).

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	Bits 0 – 1 correspond to fields 2 thru 3 respectively
2	Time	Unsigned Integer	N/A	Bits 0-9: milliseconds, range 0...999 Bits 10-15: Seconds, range 0...59 Bits 16 – 21: Minutes, range 0...59 Bits 22-26: Hour (24 hour clock), range 0..23 Bits 27-31: Day, range 1...31
3	Date	Unsigned Short Integer	N/A	Bits 0-4: Day, range 1...31 Bits 5-8: Month, range 1...12 Bits 9 – 15: Year, range 2000...2127 Where 0 is 2000, 1 is 2001, etc.

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	R	3	2

"R" indicates that the bit is reserved.

2.1.2 Event Setup and Control – Codes 01F0-01FF

Code 01F0h: Create Event

This message is used to set up an event. Required fields are 1, 2, 3, and 4. Field 1 provides a mapping of which optional fields are included. Field 2 is a local request ID that the event provider returns in the Confirm or Reject message. Field 3 contains the JAUS Message Code of the requested Event (Report) message. Field 4 is the Event Type, which allows the requester to specify the type of event – Periodic specifies that the event is a service connection request and should not be queued, in which case field 14 (Requested periodic rate) must be included. Event type of Every Change specifies that the corresponding Report message should be sent every time the data associated with that message changes, subject to the optional boundary conditions. Event type of First Change specifies that the Report message should be sent only the first time the data associated with that message changes, subject to the optional boundary conditions. Event type of First Change In and Out of Boundary specifies that the Report message should be sent the first time the data associated with that messages changes, subject to the boundary conditions, and again when the boundary conditions are no longer satisfied (as an exit report). Event type of Periodic without Replacement specifies that the report should be generated at the given periodic rate, but should be treated as a regular message and not subject to existing service connection replacement rules. Event type of One Time-On Demand specifies that an event is a one-time event that should be immediately triggered; this is equivalent to Query/Report pair.

Fields 5 through 12 are semantically linked. If none of these fields are specified and the Event Type is Every Change, then any and all changes to that message should trigger the event. If Field 4 is included, Fields 6 and either Fields 7 and 8, Fields 9 and 10, or Fields 11 and 12 should also be included.

Field 5 allows the requester to specify triggering conditions for events. An event could be triggered when a condition is:

Exactly met, most likely used for discrete fields, such as Status or waypoint number;

value = trigger condition

Not met, often used for discrete fields like Status

value != trigger condition

Between two values (such as a payload arm position)

trigger low <= value <= trigger high

trigger low < value < trigger high

Outside of two values (such as a temperature reading being too high or too low),

value <= trigger low OR value >= trigger high

value < trigger low OR value > trigger high

Above a given value (a temperature is too high, speed is too fast, notification when a robot has come up to speed)

value > trigger condition

value >= trigger condition

Below a given value (low fuel or battery).

value < trigger condition

value <= trigger condition

When used in conjunction with "Periodic" event type, a service connection is activated when the trigger condition is met and suspended when it is not met. When used in conjunction with the event type "Every Change", an event will be triggered when that triggering condition is met and every value that meets that triggering criteria should cause an event notification to occur. Used with "First Change" event type, an event would be triggered once per trigger event – that is, if the watched value crosses the

trigger boundary, it triggers an event, then does not trigger another event until the value goes back outside the triggering boundary and back in.[1]

Field 6 specifies the field number of the field in the report message that corresponds to the boundary condition and Fields 7, 9, and 11 specifies its data type. It should be included if and only if a boundary condition is specified (field 5). Field 8 should be used to specify the lower boundary value for Inside, Outside, and Low Boundary Types. Field10 should be used to specify the upper boundary value for Inside, Outside, and High Boundary Types. Field 12 should be used if and only if the Event Boundary type is Equal.

Field 13 is used for throttling updates in periodic messages. If a Periodic Event is created for position, and the vehicle is sitting still, it allows the requester to tell the provider to throttle-back the flow during that time. If the Requested Update Rate (field 14) is 5 and the Requested Minimum Update Rate (field 13) is 1, the events will be generated at a rate of 5 HZ if it is changing, but only 1 HZ when the content of the message (i.e., position) is not changing. Field 14, as specified earlier, is the requested rate for periodic events.

Field 15 contains the size of the Query message that is to specify the contents of the Report. Field 16 contains the Query message body.

Note, there is no message code for the Query message; this requires that the Report message have at most one corresponding Query message and that no other Query message be used to set up this Report.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Request ID	Byte	N/A	Local request ID for use in confirm event

[1] This assumes that if the requester wants to know when the trigger condition ceases, a separate "first change" event will be issued with the opposite trigger.

Field #	Name	Type	Units	Interpretation
3	Message Code	Unsigned Short Integer	N/A	Message code of the Event message that the receiving component will generate for this message stream.
4	Event Type	Byte	N/A	Type of event, enumeration: 0: Periodic (SC) 1: Every Change 2: First Change 3: First change in and out of boundary 4: Periodic w/o replacement 5: One time, on demand
5	Event Boundary	Byte	N/A	Boundary condition on event trigger, enumeration: 0: Equal 1: Not Equal 2: Inside Inclusive 3: Inside Exclusive 4: Outside Inclusive 5: Outside Exclusive 6: Greater than or Equal 7: Strictly Greater than 8: Less than or Equal 9: Strictly Less than
6	Limit Data Field	Byte	N/A	Field from Event message to base trigger limit on
7	Lower Limit Data Field Type	Byte	N/A	Enumeration 0: byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
8	Lower Limit	Variable-type	varies	Lower limit for trigger condition, used for Inside, Outside, and Low Event Boundary Types

Field #	Name	Type	Units	Interpretation
9	Upper Limit Data Field Type	Byte	N/A	Enumeration 0: byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
10	Upper Limit	Variable-type	varies	Upper limit for trigger condition, used for Inside, Outside, and High Event Boundary Types
11	State Data Field Type	Byte	N/A	Enumeration 0: byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
12	State	Variable-type	varies	Trigger value used for Equal Event Boundary Type. Typically used for discrete-type events.
13	Requested Minimum Periodic Rate	Unsigned Short Integer	Hertz	For Periodic events, used to throttle messages if the value is not changing. Desired update rate: Scaled Integer Lower Limit: 0 Upper Limit: 1092
14	Requested Periodic Update Rate	Unsigned Short Integer	Hertz	Desired update rate: Scaled Integer Lower Limit: 0 Upper Limit: 1092
15	Query Message Size	Unsigned Integer	bytes	The size of the following JAUS Query message body
16	Query Message Body	JAUS Message	N/A	The JAUS Query message body to be used by the receiving component to generate the Report message(s).

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	15-16	14	13	11/12	9/10	7/8	6	5

"R" indicates that the bit is reserved.

Code 01F1h: Update Event

The Update Event message allows the requester to request a rate or threshold change. The format is the same as the Create Event, only with the addition of Event ID field to specify the given event.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Request ID	Byte	N/A	Local request ID for use in confirm event
3	Message Code	Unsigned Short Integer	N/A	Message code of the Event message
4	Event Type	Byte	N/A	Type of event, enumeration: 0: Periodic (SC) 1: Every Change 2: First Change 3: First change in and out of boundary 4: Periodic w/o replacement 5: One time, on demand
5	Event Boundary	Byte	N/A	Boundary condition on event trigger, enumeration: 0: Equal 1: Not Equal 2: Inside Inclusive 3: Inside Exclusive 4: Outside Inclusive 5: Outside Exclusive 6: Greater than or Equal 7: Strictly Greater than 8: Less than or Equal 9: Strictly Less than
6	Limit Data Field	Byte	N/A	Field from Report message to base trigger limit on

Field #	Name	Type	Units	Interpretation
7	Lower Limit Data Field Type	Byte	N/A	Enumeration 0: byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
8	Lower Limit	Variable-type	varies	Lower limit for trigger condition, used for Inside, Outside, and Low Event Boundary Types
9	Upper Limit Data Field Type	Byte	N/A	Enumeration 0: byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
10	Upper Limit	Variable-type	varies	Upper limit for trigger condition, used for Inside, Outside, and High Event Boundary Types
11	State Data Field Type	Byte	N/A	Enumeration 0: byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
12	State	Variable-type	varies	Trigger value used for Equal Event Boundary Type. Typically used for discrete-type events.

Field #	Name	Type	Units	Interpretation
13	Requested Minimum Periodic Rate	Unsigned Short Integer	Hertz	For Periodic events, used to throttle messages if the value is not changing. Desired update rate: Scaled Integer Lower Limit: 0 Upper Limit: 1092
14	Requested Periodic Update Rate	Unsigned Short Integer	Hertz	Desired update rate: Scaled Integer Lower Limit: 0 Upper Limit: 1092
15	Event ID	Byte	N/A	Unique Identifier to of existing event to update
16	Query Message Size	Unsigned Integer	bytes	The size of the following JAUS Query message body
17	Query Message Body	JAUS Message	N/A	The JAUS Query message body to be used by the receiving component to generate the Report message(s).

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	16-17	14	13	11/12	9/10	7/8	6	5

"R" indicates that the bit is reserved.

Code 01F2h: Cancel Event

The Cancel Event message is used by the requester to cancel and/or request deletion of the specified event.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Request ID	Byte	N/A	Local Request ID for use in confirm/reject message
3	Message Code	Unsigned Short	N/A	Message Code of the event message to be stopped
4	Event ID	Byte	N/A	Unique ID of the event to be removed

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0

Vector to Data Field Mapping for Above Command								
Data Field	R	R	R	R	R	R	4	3

"R" indicates that the bit is reserved.

Code 01F3h: Confirm Event Request

The Confirm Event message is used to confirm an Event has been created/updated/or cancelled. Field 2 represents the Request ID from the Create, Update, or Cancel message that initiated this message. The Request ID's scope is local to the requesting client only. Field 4, Event ID, is a globally unique ID that is established for the event.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Request ID	Byte	N/A	ID of the event maintenance request (Create, Update, or Cancel)
3	Message Code	Unsigned Short	N/A	Message Code of the Event message
4	Event ID	Byte	N/A	The identifier of the specific event
5	Confirmed Periodic Update Rate	Unsigned Short	Hertz	Scaled Integer: Lower Limit = 0 Upper Limit = 1092
6	Response Code	Byte	N/A	Enumeration: 0 = successful 1 = Periodic events not supported 2= Change-based events not supported 4 = Connection Refused 5 = Invalid event setup 6 = Message not supported 7 = Error, invalid event ID for update event request

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	R	R	5

"R" indicates that the bit is reserved.

Code 01F4h: Reject Event Request

The Reject Event Request message is used to reject an Event creation, update or cancellation. Field 2 represents the Request ID from the Create, Update, or Cancel message that initiated this message. The Request ID's scope is local to the requesting client only. Field 4, Event ID, is a globally unique ID that is established for the event.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Request ID	Byte	N/A	ID of the event maintenance request (Create, Update, or Cancel)
3	Response Code	byte	N/A	Enumeration: 1 = Periodic events not supported 2= Change-based events not supported 4 = Connection Refused 5 = Invalid event setup 6 = Message not supported 7 = Error, invalid event ID for update or cancel event request
4	Error Message	String	N/A	Optional string for additional information

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	R	R	4

"R" indicates that the bit is reserved.

2.1.3 Communications Subgroup – Codes 0200-03FF

Code 0200h: Set Data Link State

This message sets the state of the Communicator's desired data link to other subsystems. The data link will remain in the selected state until commanded to change or the data link fails.

Field #	Name	Type	Units	Interpretation
1	State	Byte	N/A	0 = Off (disconnect link) 1 = On (send and receive messages) 2 = Standby (maintain link, do not transmit) 3-255 Reserved

Code 0201h: Set Data Link Select

This message sets the desired data link to an alternate communication device if more than one exists. This data link shall be used until another Set Data Link Select message is sent changing it.

Field #	Name	Type	Units	Interpretation
1	Data Link ID	Byte	N/A	0 … 255

Code 0202h: Set Selected Data Link State

This message sets the state of the Communicator's selected data link to other subsystems. The data link will remain in the selected state until commanded to change or the data link fails.

Field #	Name	Type	Units	Interpretation
1	Data Link ID	Byte	N/A	0..255
2	State	Byte	N/A	0 = Off (disconnect link) 1 = On (send and receive messages) 2 = Standby (maintain link, do not transmit) 3-255 Reserved

2.1.4 Platform Subgroup – Codes 0400-05FF

Code 0405h: Set Wrench Effort

This message controls platform mobility actuators by mapping the twelve command elements to the specific mobility controls of a vehicle. The command consists of a six element propulsive wrench and a six element resistive wrench. The six elements of each wrench break down into three linear elements and three rotational elements, which are mapped to the three axis orthogonal coordinate frame of the vehicle.

All elements of the Wrench message are not necessarily applicable to a particular platform. For example, a typical wheeled vehicle can be controlled with only three elements of the wrench command: Propulsive Linear Effort X (throttle), Propulsive Rotational Effort Z (steering), and Resistive Linear Effort X (brake).

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See mapping table that follows.
2	Propulsive Linear Effort X	Short Integer	Percent	Scaled Integer Lower Limit = -100 Upper Limit = 100
3	Propulsive Linear Effort Y			
4	Propulsive Linear Effort Z			
5	Propulsive Rotational Effort X			
6	Propulsive Rotational Effort Y			
7	Propulsive Rotational Effort Z			
8	Resistive Linear Effort X	Byte	Percent	Scaled Integer Lower Limit = 0 Upper Limit = 100
9	Resistive Linear Effort Y			
10	Resistive Linear Effort Z			
11	Resistive Rotational Effort X			
12	Resistive Rotational Effort Y			
13	Resistive Rotational Effort Z			

Vector to Data Field Mapping for Above Command								
Vector Bit	15	14	13	12	11	10	9	8
Data Field	R	R	R	R	13	12	11	10
Vector Bit	7	6	5	4	3	2	1	0
Data Field	9	8	7	6	5	4	3	2

"R" indicates that the bit is reserved.

Code 0406h: Set Discrete Devices

This message controls discrete platform functions.

Field #2: The Main Propulsion bit field controls various functions associated with power source state control. The main propulsion unit on an unmanned system could be

gasoline internal combustion, diesel internal combustion, electric motor, or some other power system. An explanation of each control bit follows:

Bit 0 – controls and reports the main power plant ON/OFF state. For a gasoline engine, this bit corresponds to the engine ignition control. For a diesel engine, this bit controls the engines run/stop mechanism, which could be a vacuum switch, electrical solenoid, or other control device. Application to an electric motor could involve the engaging and disengaging of a safety type device.

- Bit 1 - controls and reports the power source of the main propulsion unit. For a gasoline or diesel engine, this could be a fuel pump, and/or fuel cut-off solenoid valve. For an electric motor, this could control a main power breaker that supplies electrical power to the motor. This bit shall be reported as 1 if the main power plant is online and ready for operation, or it is executing a normal or aborted starting or stopping sequence, and shall be reported as 0 only if the main power plant is fully shutdown.
- Bit 2 - controls and reports an auxiliary power source, if one exists. This could be an auxiliary fuel supply for an internal combustion engine.
- Bit 3 - controls and reports the power to auxiliary devices on the vehicle. Battery power to some devices might be temporarily suspended while the engine starter is engaged. It might also be desirable to enable power to some devices while the engine is off.
- Bit 4 - controls the starting device, if one exists. For an internal combustion engine, this is the starter motor.
- Bit 5 - allows special controls to be engaged if the engine requires them for starting in a cold state. A manual choke is one example of such a device.
- Bits 6 and 7 - are for use in vehicles that are capable of automatic startup and shutdown. For a vehicle with this capability, all controls needed to startup and control the shutdown of the main propulsion unit would use these bits. During automatic starting sequence, bit 6 shall be reported as 1 until the main power plant is online and ready for operation or until the startup sequence has failed or been aborted and ended with the main power plant returned to shutdown. During automatic stopping sequences, bit 7 shall be reported as 1 until the main power plant is fully shutdown.

The devices controlled by bits 0 – 5 are binary devices. When a command is received, if the command is not the current state, then the state toggles. If a command is received with a commanded state equal to the current state, no state change occurs.

Bits 6 and 7 control automated startup or shutdown procedures. They are mutually exclusive from each other and bit 4. Bit 6 is ignored if either bit 0 or 1 or both are off. Bit 7 allows the orderly normal shutdown of the main propulsion system. If the main propulsion unit is online and this message is received with either bit 0 or 1 or both off, the main propulsion system should be shutdown as quickly as possible without damage. If this message is received with either bit 0 or 1 or both off during a start sequence, the start sequence should be aborted and the main propulsion system returned to shutdown as quickly as possible without damage. If this message is received with either bit 0 or 1 or both off during a normal stop sequence, the stop sequence should be aborted and the main propulsion system shutdown as quickly as possible without damage. When a command is received, the startup or shutdown procedure runs to completion. If main propulsion is online and the startup command is received, it shall be ignored. If the main propulsion is offline and the shutdown command is received, it shall be ignored.

Field #3 controls the vehicle's parking brake and horn.

Fields #4-5 control the vehicle transmission gearing.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table that follows.
2	Main Propulsion	Byte	N/A	Bit 0, 1/0 = On/Off Bit 1, 1/0 = Main Energy /Fuel Supply On/Off Bit 2, 1/0 = Auxiliary Energy /Fuel Supply On/Off Bit 3, 1/0 = Power to Auxiliary Devices On/Off Bit 4, 1/0 = Starting Device On/Off Bit 5, 1/0 = Cold Start True/False Bit 6, 1 = Commence Automatic Start Sequence Bit 7, 1 = Commence Automatic Shutdown Sequence
3	Parking Brake, and Horn	Byte	N/A	Bit 0, 1/0 = Parking Brake Set/Release Bit 1, 1/0 = Horn On/Off Bits 2 - 7, Reserved

Field #	Name	Type	Units	Interpretation
4	Gear	Byte	N/A	0 = Park 1,2, ... 127 = Forward 128 = Neutral 129,130, ... 255 = Reverse
5	Transfer Case	Byte	N/A	0,1, …127 = Low 128 = Neutral 129,130, ... 255 = High

Presence Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	5	4	3	2

"R" indicates that the bit is reserved.

Code 0407h: Set Global Vector

Field #2 sets the desired speed of the platform. The desired heading angle is set in field #4 and is defined in a right hand sense about the Z axis of the global coordinate system (the Z axis points downward) where North is defined as zero degrees. Field #3 sets the desired Altitude above sea level (ASL). The altitude is in accordance with the WGS 84 standard. The desired roll angle is set in field #5 and is also defined in a right hand sense about the X axis of the global coordinate system. The desired pitch angle is set in field #6 in a right hand sense about the Y axis.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See Mapping table that follows.
2	Speed	Unsigned Short Integer	Meters per Second	Scaled Integer Lower Limit = 0 Upper Limit = 10,000
3	Altitude (ASL)	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
4	Heading	Short Integer	Radians	Scaled Integer Lower Limit = -π Upper Limit = π
5	Roll	Short Integer	Radians	Scaled Integer Lower Limit = -π Upper Limit = π

Field #	Name	Type	Units	Interpretation
6	Pitch	Short Integer	Radians	Scaled Integer Lower Limit = -π Upper Limit = π

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	6	5	4	3	2

"R" indicates that the bit is reserved.

Code 0408h: Set Local Vector

Field #1 sets the desired speed of the platform. The desired heading angle is set in field #2 and is defined in a right hand sense about the Z axis of the local coordinate system (the Z axis points downward) where zero degrees defines a heading that is parallel to the X axis of the local coordinate system.

Field #	Name	Type	Units	Interpretation
1	Speed	Integer	Meters per Second	Scaled Integer Lower Limit = 0 Upper Limit = 10,000
2	Heading	Short Integer	Radians	Scaled Integer Lower Limit = -π Upper Limit = π

Code 040Ah: Set Travel Speed

Field #1 sets the desired speed of the platform.

Field #	Name	Type	Units	Interpretation
1	Speed	Unsigned Short Integer	Meter per Second	Scaled Integer Lower Limit = 0 Upper Limit = 10,000

Code 040Ch: Set Global Waypoint

A global waypoint can have up to six fields to describe it completely. The waypoint is defined in the global coordinate system using the latitude, longitude, and altitude, fields.

The desired orientation of the platform at the waypoint is defined using the roll, pitch, and yaw fields. Only fields 1-4 (presence vector, waypoint number, latitude, and longitude) are required for each waypoint. The presence vector is used to specify if the remaining fields, i.e., altitude, roll, pitch, and yaw, are used to further describe the waypoint and the desired orientation at that point. The waypoint number starts at zero for the first waypoint, and must be incremented by one for each additional waypoint.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table that follows. The presence vector applies to all waypoints.
2	Waypoint Number	Unsigned Short Integer	N/A	Lower Limit = 0 Upper Limit = 65,535
3	Latitude	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
4	Longitude	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
5	Altitude, (ASL)	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
6	ϕ (Roll)	Short Integer	Radians	Scaled Integer Lower Limit = $-\pi$ Upper Limit = π
7	θ (Pitch)			
8	ψ (Yaw)			

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	8	7	6	5

"R" indicates that the bit is reserved.

Code 040Dh: Set Local Waypoint

A local waypoint can have up to six fields to describe it completely. The waypoint is defined in a local coordinate system using the X, Y, and Z fields. The desired orientation of the platform at the waypoint is defined using the roll, pitch, and yaw fields. Only fields 1-4 (presence vector, waypoint number, the X, and Y) are required for each waypoint. The presence vector is used to specify if the remaining fields, i.e., Z,

roll, pitch, and yaw, are used to further describe each waypoint and the desired orientation at that point. The waypoint number starts at zero for the first waypoint, and must be incremented by one for each additional waypoint.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table that follows. The presence vector applies to all waypoints.
2	Waypoint Number	Unsigned Short Integer	N/A	Lower Limit = 0 Upper Limit = 65,535
3	X	Integer	Meters	Scaled Integer Lower Limit = -100,000 Upper Limit = 100,000
4	Y	Integer	Meters	Scaled Integer Lower Limit = -100,000 Upper Limit = 100,000
5	Z	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
6	ϕ (Roll)	Short Integer	Radians	Scaled Integer Lower Limit = $-\pi$ Upper Limit = π
7	θ (Pitch)			
8	ψ (Yaw)			

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	8	7	6	5

"R" indicates that the bit is reserved.

Code 040Fh: Set Global Path Segment

A global path segment is defined in this message using three points, $\underline{\mathbf{P}}_0$, $\underline{\mathbf{P}}_1$ and $\underline{\mathbf{P}}_2$ and a weighting factor. The path segment number starts at zero for the first path segment, and must be incremented by one for each additional path segment. For the first path segment, i.e. where the path segment number is equal to zero, $\underline{\mathbf{P}}_0$ is assumed to be the current location of the platform as defined by Report Global Pose. For each successive path segments, i.e. where the path segment number is greater than zero, $\underline{\mathbf{P}}_0$ is equal to the previous path segment's $\underline{\mathbf{P}}_2$. Therefore, for each message, only $\underline{\mathbf{P}}_1$, $\underline{\mathbf{P}}_2$, and a weighting factor must be set in order to define a path segment. Each point is defined in the Global

Coordinate System by setting its Latitude, Longitude and Altitude. Both the Latitude and Longitude are required fields, but the Altitude, field is optional. Bits 0 and 1 in the presence vector are used to determine if the Altitude, fields are used.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table that follows.
2	Path Segment Number	Unsigned Short Integer	N/A	Lower Limit = 0 Upper Limit = 65,535
3	$P_{1,latitude}$	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
4	$P_{1,longitude}$	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
5	$P_{1,Altitude,(ASL)}$	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
6	$P_{2,latitude}$	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
7	$P_{2,longitude}$	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
8	$P_{2,Altitude,(ASL)}$	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
9	Weighting Factor	Unsigned Short Integer	N/A	Scaled Integer Range 0 to 500. Zero = a straight line

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	R	8	5

"R" indicates that the bit is reserved.

Code 0410h: Set Local Path Segment

A local path segment is defined in this message using three points, $\underline{\mathbf{P}}_0$, $\underline{\mathbf{P}}_1$ and $\underline{\mathbf{P}}_2$ and a weighting factor. The path segment number starts at zero for the first path segment, and

must be incremented by one for each additional path segment. For the first path segment, i.e. where the path segment number is equal to zero, $\underline{\mathbf{P}}_0$ is assumed to be the current location of the platform as defined by Report Local Pose. For each successive path segments, i.e. where the path segment number is greater than zero, $\underline{\mathbf{P}}_0$ is equal to the previous path segment's $\underline{\mathbf{P}}_2$. Therefore, for each message, only $\underline{\mathbf{P}}_1$, $\underline{\mathbf{P}}_2$, and a weighting factor must be set in order to define a path segment. Each point is defined in the Local Coordinate System by setting its X, Y, and Z. Both the X and Y are required fields, but the Z field is optional. Bit 0 in the presence vector is used to determine if the Z field is used.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table that follows.
2	Path Segment Number	Unsigned Short Integer	N/A	Lower Limit = 0 Upper Limit = 65,535
3	$P_{1,X}$	Integer	Meters	Scaled Integer Lower Limit = -100,000 Upper Limit = 100,000
4	$P_{1,Y}$	Integer	Meters	Scaled Integer Lower Limit = -100,000 Upper Limit = 100,000
5	$P_{1,Z}$	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
6	$P_{2,X}$	Integer	Meters	Scaled Integer Lower Limit = -100,000 Upper Limit = 100,000
7	$P_{2,Y}$	Integer	Meters	Scaled Integer Lower Limit = -100,000 Upper Limit = 100,000
8	$P_{2,Z}$	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
9	Weighting Factor	Unsigned Short Integer	N/A	Scaled Integer Range 0 to 500. Zero = a straight line

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	R	8	5

"R" indicates that the bit is reserved.

2.1.5 Manipulator Subgroup – Codes 0600-07FF

Code 0601h: Set Joint Efforts

Field #1 indicates the number of joint effort commands contained in this message. This message sets the desired joint effort values.

Field #	Name	Type	Units	Interpretation
1	Num Joints	Byte	N/A	1 ... 255 0 is Reserved
2	Joint 1 –effort	Short Integer	Percent	Scaled integer Lower Limit = -100% Upper Limit = +100%
3 … n	…			
n + 1	Joint n –effort	Short Integer	Percent	Scaled integer Lower Limit = -100% Upper Limit = +100%

Code 0602h: Set Joint Positions

Field #1 indicates the number of joint position commands contained in this message. This message sets the desired joint position values.

Field #	Name	Type	Units	Interpretation
1	Number of Joints	Byte	N/A	1 ... 255 0 is Reserved
2	Joint 1 – position	Integer	Radians or Meters	Scaled integer: If revolute joint, Lower limit = -8π rad Upper limit = +8π rad If prismatic joint, Lower limit = -10 m Upper limit = +10 m
3 ... n	…			
N + 1	Joint n – position	Integer	Radians or Meters	see field 2

Code 0603h: Set Joint Velocities

Field #1 indicates the number of joint velocity commands contained in this message. This message sets the desired joint velocity values.

Field #	Name	Type	Units	Interpretation
1	Num Joints	Byte	N/A	1 ... 255, 0 is Reserved
2	Joint 1 –velocity	Integer	Radians/ Second or Meters/ Second	Scaled Integer If revolute joint, Lower limit = -10π rad/sec Upper limit = +10π rad/sec If prismatic joint, Lower limit = -5 m/sec Upper limit = +5 m/sec
3 ... n	…			
N + 1	Joint n –velocity	Integer	Radians/ Second or Meters/ Second	see field 2

Code 0604h: Set Tool Point

This message specifies the coordinates of the end-effector tool point in terms of the coordinate system attached to the end-effector. For a six-axis robot, this coordinate system is defined by having its origin located at the intersection of the $\mathbf{S}_6$ joint axis vector and the user defined link vector $\mathbf{a}_{67}$. The Z axis of the coordinate system is along $\mathbf{S}_6$ and the X axis is along the $\mathbf{a}_{67}$ vector.

Field #	Name	Type	Units	Interpretation
1	X coordinate of tool point	Integer	Meters	Scaled Integer Lower limit = -15 m Upper limit = +15 m
2	Y coordinate of tool point	Integer	Meters	see field 1
3	Z coordinate of tool point	Integer	Meters	see field 1

Code 0605h: Set End-Effector Pose

This message defines the desired end-effector position and orientation. The coordinates of the tool point are defined in terms of the vehicle coordinate system. The orientation of the end-effector is defined by a unit quaternion (d ; a, b, c) which specifies the axis and angle of rotation that was used to establish the orientation of the end-effector coordinate system with respect to the vehicle coordinate system.

Field #	Name	Type	Units	Interpretation
1	X component of tool point	Integer	Meters	Scaled Integer Lower limit = -30 m Upper limit = +30 m
2	Y component of tool point	Integer	Meters	see field 1
3	Z component of tool point	Integer	Meters	see field 1
4	d component of unit quaternion q	Integer	N/A	Scaled Integer Lower limit = -1 Upper limit = +1
5	a component of unit quaternion q	Integer	N/A	see field 4
6	b component of unit quaternion q	Integer	N/A	see field 4
7	c component of unit quaternion q	Integer	N/A	see field 4

Code 0606h: Set End-Effector Velocity State

The velocity state of body B measured with respect to body A is defined as the angular velocity of body B with respect to body A, ${}^{A}\boldsymbol{\omega}^{B}$, and the linear velocity of the point in body B that is coincident with the origin of the reference frame measured with respect to body A, ${}^{A}\mathbf{v}_{0}^{B}$. From these parameters, the velocity of any point in body B can be determined from the equation ${}^{A}\mathbf{v}_{P}^{B} = {}^{A}\mathbf{v}_{0}^{B} + {}^{A}\boldsymbol{\omega}^{B} \times \mathbf{r}_{0\to P}$. Here ${}^{A}\mathbf{v}_{P}^{B}$ is the velocity of some point P, that is embedded in body B, measured with respect to body A and $\mathbf{r}_{0\to P}$ represents the vector from the origin of the reference frame to point P, i.e. the

coordinates of point P. In this application, body B is the end-effector, and body A is ground. The reference coordinate system is embedded in ground, but is aligned with either the end-effector coordinate system or the vehicle coordinate system at this instant.

Field #	Name	Type	Units	Interpretation
1	Coordinate system definition	Byte	N/A	1 = reference coord. system aligned with vehicle coord. sys. 2 = reference coord. system aligned with end-effector coord. sys.
2	X component of angular velocity ${}^{A}\boldsymbol{\omega}^{B}$	Integer	Radians/ Second	Scaled Integer Lower limit = -20 π rad/sec Upper limit = +20 π rad/sec
3	Y component of angular velocity ${}^{A}\boldsymbol{\omega}^{B}$	Integer	Radians/ Second	see field 2
4	Z component of angular velocity ${}^{A}\boldsymbol{\omega}^{B}$	Integer	Radians/ Second	see field 2
5	X component of the linear velocity ${}^{A}\mathbf{v}_{0}^{B}$	Integer	Meters/ Second	Scaled Integer Lower limit = -10 m/sec Upper limit = +10 rad/sec
6	Y component of the linear velocity ${}^{A}\mathbf{v}_{0}^{B}$	Integer	Meters/ Second	see field 5
7	Z component of the linear velocity ${}^{A}\mathbf{v}_{0}^{B}$	Integer	Meters/ Second	see field 5

Code 0607h: Set Joint Motion

Field #1 indicates the number of joint motion commands contained in this message. This message sets the desired joint motion values.

Field #	Name	Type	Units	Interpretation
1	Num Joints, n	Byte	N/A	1 ... 255 0 is Reserved

Field #	Name	Type	Units	Interpretation
2	number of poses, p	Byte	N/A	1…255 0 is Reserved
3	pose 1 time	Integer	Seconds	Scaled Integer Lower limit = 0 sec Upper limit = 6000 sec
4	Joint 1 –position at pose 1	Integer	Radians or Meters	Scaled Integer If revolute joint, Lower limit = -8π rad Upper limit = +8π rad If prismatic joint, Lower limit = -10 m Upper limit = +10 m
5	Joint 1 – max velocity	Integer	Radians/ Second or Meters/ Second	Scaled Integer If revolute joint, Lower limit = -10π rad/sec Upper limit = +10π rad/sec If prismatic joint, Lower limit = -5 m/sec Upper limit = +5 m/sec
6	Joint 1 – max acceleration	Integer	Radians/ Second^2 or Meters/ Second^2	Scaled Integer If revolute joint, Lower limit = -10π rad/sec^2 Upper limit = +10π rad/sec^2 If prismatic joint, Lower limit = -20 m/sec^2 Upper limit = +20 m/sec^2
7	Joint 1 – max deceleration	Integer	Radians/ Second^2 or Meters/ Second^2	Scaled Integer If revolute joint, Lower limit = -10π rad/sec^2 Upper limit = +10π rad/sec^2 If prismatic joint, Lower limit = -20 m/sec^2 Upper limit = +20 m/sec^2
…				
4n	Joint n – position at pose 1	Integer	Radians or Meters	see field 4
4n+1	Joint n – max velocity	Integer	Radians/ Second or Meters/ Second	see field 5

Field #	Name	Type	Units	Interpretation
4n+2	Joint n – max acceleration	Integer	Radians/ Second^2 or Meters/ Second^2	see field 6
4n+3	Joint n – max deceleration	Integer	Radians/ Second^2 or Meters/ Second^2	see field 7
…				
(p-1) ×4n+ 4	pose p time	Integer	Seconds	see field 3
(p-1) ×4n+ 5	Joint 1 – position at pose p	Integer	Radians or Meters	see field 4
(p-1) ×4n+6	Joint 1 – max velocity	Integer	Radians/ Second or Meters/ Second	see field 5
(p-1) ×4n+7	Joint 1 – max acceleration	Integer	Radians/ Second^2 or Meters/ Second^2	see field 6
(p-1) ×4n+8	Joint 1 – max deceleration	Integer	Radians/ Second^2 or Meters/ Second^2	see field 7
…				
(p-1) ×4n+4n	Joint n – position at pose p	Integer	Radians or Meters	see field 4
(p-1) ×4n+4n +1	Joint n – max velocity	Integer	Radians/ Second or Meters/ Second	see field 5
(p-1) ×4n+4n +2	Joint n – max acceleration	Integer	Radians/ Second^2 or Meters/ Second^2	see field 6
(p-1) ×4n+4n +3	Joint n – max deceleration	Integer	Radians/ Second^2 or Meters/ Second^2	see field 7

Code 0608h: Set End-Effector Path Motion

A series of end-effector poses are defined in terms of the vehicle coordinate system at various times. The time is a relative time and is defined in seconds where time=0 is the moment that the motion towards the first pose begins.

Field #	Name	Type	Units	Interpretation
1	number of poses, n	Byte	N/A	1 ... 255 0 is Reserved
2	time 1	Integer	Seconds	time for pose 1 Scaled integer: Lower limit = 0 sec Upper limit = 6000 sec
3	X component of tool point for pose 1	Integer	Meters	Scaled integer: Lower limit = -30 m Upper limit = +30 m
4	Y component of tool point for pose 1	Integer	Meters	see field 3
5	Z component of tool point for pose 1	Integer	Meters	see field 3
6	d component of unit quaternion q for pose 1	Integer	N/A	Scaled integer: Lower limit = -1 Upper limit = +1
7	a component of unit quaternion q for pose 1	Integer	N/A	see field 6
8	b component of unit quaternion q for pose 1	Integer	N/A	see field 6
9	c component of unit quaternion q for pose 1	Integer	N/A	see field 6
…				
8n-6	time n	Integer	Seconds	see field 2
8n-5	X component of tool point for pose n	Integer	Meters	see field 3

Field #	Name	Type	Units	Interpretation
8n-4	Y component of tool point for pose n	Integer	Meters	see field 3
8n-3	Z component of tool point for pose n	Integer	Meters	see field 3
8n-2	d component of unit quaternion q for pose n	Integer	N/A	see field 6
8n-1	a component of unit quaternion q for pose n	Integer	N/A	see field 6
8n	b component of unit quaternion q for pose n	Integer	N/A	see field 6
8n+1	c component of unit quaternion q for pose n	Integer	N/A	see field 6

2.1.6 Environment Sensor Subgroup – Codes 0800-09FF

Code 0801h: Set Camera Pose

This message is used to control the position and orientation of cameras with translational and/or rotational capabilities. An example is a camera mounted on a pan-tilt mechanism.

The translation and rotation of the camera are with respect to the camera's coordinate frame (see **Figure 2-1**). In this frame, panning is a rotation of the camera about its Y-axis. A positive Y-axis rotation will rotate the camera to the viewer's right. Tilt is a rotation of the camera about its X-axis. A positive X-axis rotation will rotate the camera up.

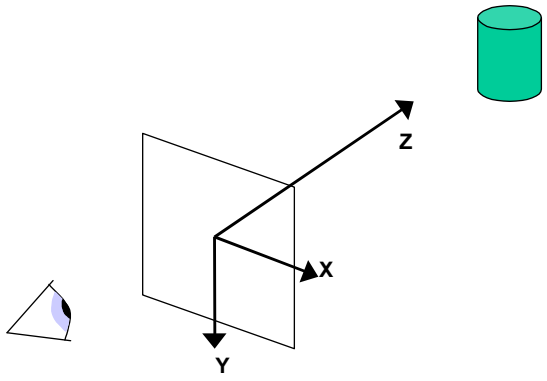

Figure 2-1: Camera Coordinate Frame

The scaled integer calculation for fields 4 – 9 is performed so that 0% will always correspond with 0 position/rate. For example, if a camera has a translational capability along its Z-axis and the range of displacement is from 0.0 to 0.5 meters, then the range used for the scaled integer calculation is ±0.5 meters. A command of +100% would correspond to +0.5 m and all commands less than or equal to 0% would result in a 0.0 displacement. Performing the scaled integer calculation using this method allows the "home" position of the camera (i.e. 0 angle, 0 position) to correspond to 0% angle and 0% position.

When this method of scaled integer calculation results in a range of commands that are beyond the range of the camera's positioning capability, the out-of-range values shall be clipped to the minimum or maximum achievable value.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	Bits 0 – 5 correspond to fields 4 thru 9 respectively.
2	Camera ID	Byte	N/A	1 … 255, 0 is Reserved
3	Mode Indicator	Byte	N/A	Bits 0 – 5, 1/0 = Rate/Position mode for field #s 4 – 9 respectively Bits 6 and 7 are Reserved.
4	X Displacement or Linear Rate	Short Integer	Percent	±100%, where full scale is

Field #	Name	Type	Units	Interpretation
5	Y Displacement or Linear Rate			determined by the greater of the positive or negative displacement and/or linear rate ranges. Scaled Integer Lower Limit = -100% Upper Limit = 100%
6	Z Displacement or Linear Rate			
7	X Angle or Angular Rotation Rate	Short Integer	Percent	±100%, where full scale is determined by the greater of the positive or negative angular displacement and/or rotation rate ranges. Scaled Integer Lower Limit = -100% Upper Limit = 100%
8	Y Angle or Angular Rotation Rate			
9	Z Angle or Angular Rotation Rate			

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	9	8	7	6	5	4

"R" indicates that the bit is reserved.

Code 0802h: Select Camera

The Select Camera message commands the Visual Component to use the selected camera for all video traffic. The messages Query Camera Count and Report Camera Count should be used to qualify the command data field.

Field #	Name	Type	Units	Interpretation
1	Camera ID	Byte	N/A	1 – 255, 0 - Reserved

Code 0805h: Set Camera Capabilities

The Set Camera Capabilities message commands the Visual Component to use the parameters specified for the currently selected camera.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See mapping table that follows.
2	Camera ID	Byte	N/A	1 … 255 0 is Reserved
3	Horizontal FOV	Unsigned Short Integer	Radians	Scaled Integer Lower Limit = 0 Upper Limit = π
4	Vertical FOV	Unsigned Short Integer	Radians	Scaled Integer Lower Limit = 0 Upper Limit = π
5	Horizontal Resolution	Unsigned Short Integer	N/A	The number of horizontal scan lines produced by the camera
6	Vertical Resolution	Unsigned Short Integer	N/A	The number of vertical scan lines produced by the camera
7	Focus	Byte	Percent	0 to 100 percent of possible focal length where 0 equates to the focal point nearest the camera. Scaled Integer Lower Limit = 0 Upper Limit = 100
8	Iris	Byte	Percent	0 to 100 percent of possible aperture width where 0 equates to a closed aperture and 100 is full open. Scaled Integer Lower Limit = 0 Upper Limit = 100
9	Gain	Byte	Percent	0 to 100 percent of possible gain where 0 equates to the lowest level of allowable visual noise and 100 is the maximum. Scaled Integer Lower Limit = 0 Upper Limit = 100
10	Frame Rate	Unsigned Short Integer	Frames per Second	For Motion Picture (Video) cameras this value represents the actual Frame Rate in Frames per Second.

Field #	Name	Type	Units	Interpretation
11	Shutter	Unsigned Short Integer	Seconds	For Still imagery this field is defined as the inverse of the field value. For example 500 would indicate a 1/500th of a second shutter speed. Speeds slower than one second are not available.
12	Imagery Control	Unsigned Short Integer	N/A	To turn a control on the indicated bit shall be set to one (1). To turn the control off the bit shall be set to zero (0). Bit 0 – Auto Focus ON Bit 1 – Auto Exposure/Iris ON Bit 2 – Image Stabilization ON Bit 3 – White Balance ON Bit 4 – Sync Flash/Strobe ON Bit 5 – Red Eye ON Bit 6 – Auto Shutter ON Bit 7 – Auto Gain ON Bit 8 – Interlaced 0, non-Interlaced 1 Bits 9– 15 Reserved
13	Audio Control	Unsigned Short Integer	N/A	To turn a control on the indicated bit shall be set to one (1). To turn the control off the bit shall be set to zero (0). Bit 0 – Audio ON Bit 1 – Auto Gain ON Bit 2 – Stereo ON Bit 3 – Directional ON Bit 4 – Front Microphone ON Bit 5 – Rear Microphone ON Bit 6 – Left Microphone ON Bit 7 – Right Microphone ON Bits 8 – 15 Reserved

Vector to Data Field Mapping for Above Command								
Vector Bit	15	14	13	12	11	10	9	8
Data Field	R	R	R	R	R	13	12	11
Vector Bit	7	6	5	4	3	2	1	0
Data Field	10	9	8	7	6	5	4	3

"R" indicates that the bit is reserved.

Code 0806h: Set Camera Format Options

The Set Camera Format Options message commands the Visual Component to use the format output as specified for the currently selected camera.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table that follows.
2	Camera ID	Byte	N/A	1 … 255 0 is Reserved
3	Audio Format	Byte	N/A	0-Unused 1-RAW 2-PCM 3-AU 4-WAV 5-MID 6-MPEG1 Layer 3 (MP3) 7-MP2 8-Advanced Streaming Format (ASF) 9-255 Reserved
4	Image Format	Byte	N/A	0 – Unused 1 – MPEG2 2 – MPEG4 3 – MJPEG 4 – NTSC 5 – PAL 6 – TIFF 7 – JPEG 8 – GIF 9 – H263 10 – H264 11 – PNG 12 – BMP 13 – RAW 14 – PPM 15 – PGM 16 – PNM 17 – 255 Reserved

Field #	Name	Type	Units	Interpretation
5	Format Option	Unsigned Integer	N/A	The value contained in this field shall not impact interoperability. Image format options beyond those published in this message may be necessary. One example use of this field is for the compression ratio for video.

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	5	4	3

"R" indicates that the bit is reserved.

2.1.7 World Model Subgroup – Codes 0A00-0AFF

Storing and sharing of vector-formatted geospatial data is supported by the World Model Vector Knowledge Store. The primary benefit of this world modeling method is that vector data typically require significantly less bandwidth to transmit as compared to raster data.

For the vector knowledge store, objects are represented as points, lines and polylines, and polygons. **Figure 2-2** shows the format of these vector objects. Polylines and polygons may consist of up to 65535 vertices. Rather than assigning the points that make up these objects Cartesian coordinates with respect to an arbitrarily chosen datum, all points are expressed as points of latitude and longitude (WGS84).

The vector objects on the right side of **Figure 2-2** have a region buffer parameter. The region buffer is defined as an offset distance in meters that establishes a radial region around each vector object vertex and connects the radial regions of two or more radial regions by drawing lines at their tangents. The area within these radial regions and tangent lines are considered to be within the vector object's buffer zone. This buffer

feature allows a region to be established in proximity to vector objects. For example, United States Geological Survey (USGS) digital line graph (DLG) road data is presented in vector form representing the center-line of such roads. It may be useful to search for objects within an area along a particular route defined in the digital line graph data. For simple cases, it may be possible to generate a polygonal representation of the area around the road. Establishing this polygon will require transmitting the coordinates of each of its vertices. As the problem scales up, this method becomes very inefficient. A better solution to this problem would be to determine the route using the DLG data and assign a region buffer to each line segment.

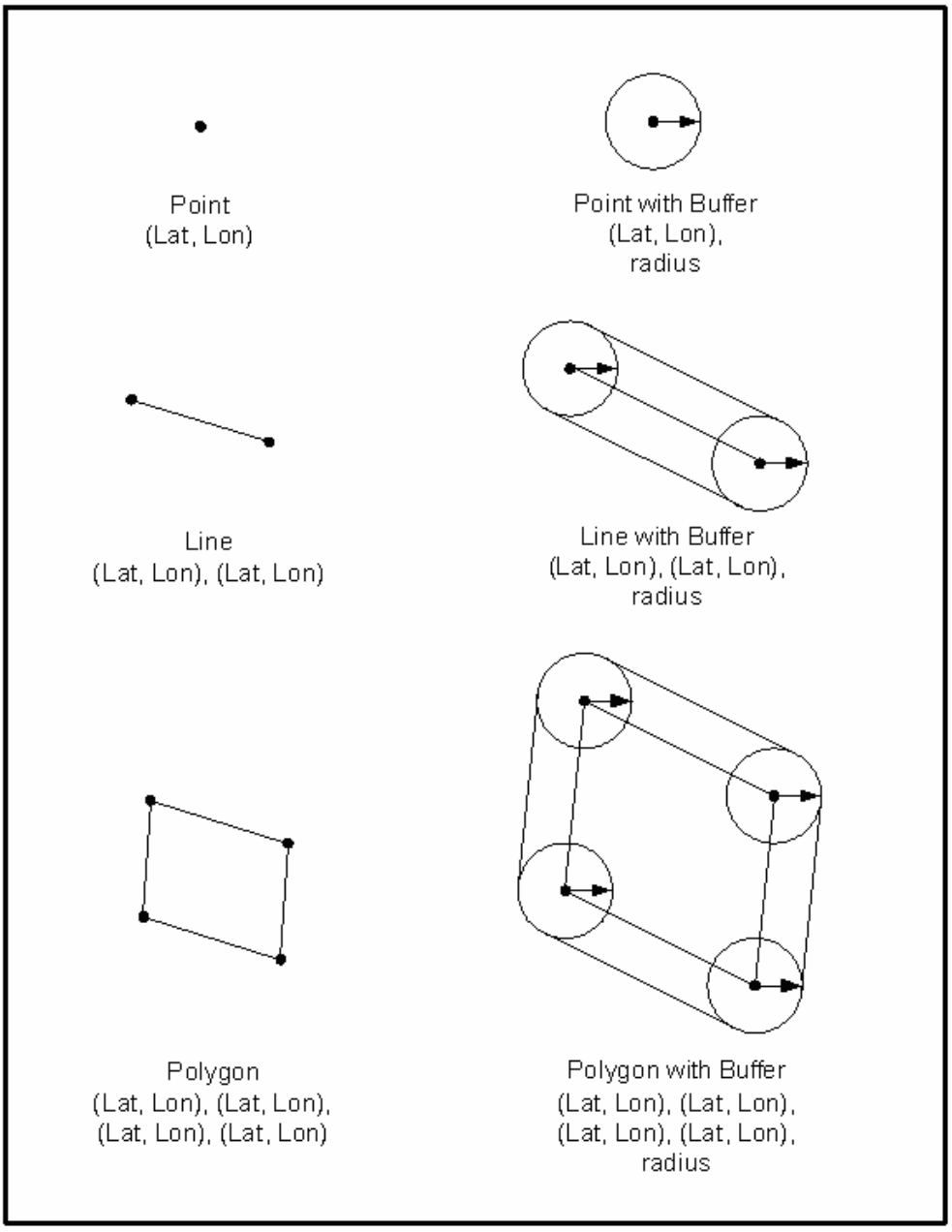

Figure 2-2: Vector Objects

Code 0A20h: Create Vector Knowledge Store Objects

The Code 0A20h: Create Vector Knowledge Store Objects message is used to add objects to the Vector Knowledge Store. This message allows multiple vector objects to be created using a single message.

Field 1 of this message is the presence vector. When multiple objects are created using the same message, the presence vector shall apply to all objects.

Field 2 of this message is the creation message properties. If bit zero is set, then the knowledge store shall return the Code 4A20h: Report Vector Knowledge Store Object(s) Creation message with the local request identifier specified in Field 3.

Field 4 indicates the number of vector objects included in the message.

Fields 5 is the beginning of the definition of a single vector object. The vector object is defined by its type (point, line, or polygon); the number of feature classes that it is assigned to; an attribute for each feature class; and the global coordinates of each of the vertices for the object. These fields are repeated for each object created using this message. Again, the presence vector applies to each vector object.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Message Properties	Byte	N/A	Bit Field 0: Request confirmation of object creation 1 – 7: Reserved
3	Local Request ID	Byte	N/A	Request identifier to be used when returning confirmation to requesting component
4	Number of Objects	Unsigned Short Integer		0, reserved 1 … 65,535
5	Object 1 Type	Byte	N/A	Enumeration 0: Point 1: Line 2: Polygon 3 – 255: Reserved
6	Object 1 Buffer	Float	Meters	
7	Object 1 Number of Feature Classes	Byte	N/A	
8	Object 1 Feature Class 1	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table

Field #	Name	Type	Units	Interpretation
				65,535 – Reserved
9	Object 1 Feature Class 1 Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
10	Object 1 Feature Class Attribute 1	Varies (see field 4)	Varies with Feature Class	
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
5+3m	Object 1 Feature Class m	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – Reserved
6+3m	Object 1 Feature Class m Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
7+3m	Object 1 Feature Class Attribute m	Varies (see previous field)	Varies with Feature Class	
8+3m	Number of Points for Object 1	Unsigned Short Integer	N/A	
9+3m	Object 1 Point 1 Latitude	Integer	Degrees	Scaled Integer Lower Limit = -90

Field #	Name	Type	Units	Interpretation
	(WGS84)			Upper Limit = 90
10+3m	Object 1 Point 1 Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
	…	…	…	…
	…	…	…	…
7+3m+2n	Object 1 Point n Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
8+3m+2n	Object 1 Point n Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
	Object p Type	Byte	N/A	Enumeration 0: Point 1: Line 2: Polygon 3 – 255: Reserved
	Object p Buffer	Float	Meters	
	Object p Number of Feature Classes	Byte	N/A	
	Object p Feature Class 1	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – Reserved
	Object p Feature Class 1 Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes)

Field #	Name	Type	Units	Interpretation
				10 – 255: Reserved
	Object p Feature Class Attribute 1	Varies (see previous field)	Varies with Feature Class	
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
	Object p Feature Class m	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – Reserved
	Object p Feature Class m Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
	Object p Feature Class Attribute m	Varies (see previous field)	Varies with Feature Class	
	Number of Points for Object p	Unsigned Short Integer	N/A	
	Object p Point 1 Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
	Object p Point 1 Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
	…	…	…	…
	…	…	…	…
	Object p Point r Latitude	Integer	Degrees	Scaled Integer Lower Limit = -90

Field #	Name	Type	Units	Interpretation
	(WGS84)			Upper Limit = 90
	Object p Point r Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	R	R	6

"R" indicates that the bit is reserved.

Code 0A21h: Set Vector Knowledge Store Feature Class Metadata

The Code 0A21h: Set Vector Knowledge Store Feature Class Metadata message allows the creation, modification, or deletion of feature class metadata. The format of these metadata is not specified. It is left to the system designer to develop a convention for doing this. Initially these data are to be used by the human operators. In the future a convention for the formatting of these metadata may be established in this document.

Field #	Name	Type	Units	Interpretation
1	Metadata Options	Byte	N/A	Enumeration 0: Append 1: Prepend 2: Overwrite 3 – 254: Reserved 255: Erase All
2	Feature Class	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – Reserved
3	Number of String Characters	Unsigned Short Integer	N/A	0 … 65,535 This field should be equal to zero only when Field 1 is equal to 255 (Erase All)
4	Feature Class Metadata	String	N/A	Variable length string

Field #	Name	Type	Units	Interpretation

Code 0A24h: Terminate Vector Knowledge Store Data Transfer

This Code 0A23h: Terminate Vector Knowledge Store Data Transfer message is a command class message that shall cause the vector knowledge store to immediately terminate the transfer of all current and outstanding data destined to the requesting component. Upon termination, the vector knowledge store shall send the requestor the Code 4A24h: Report Vector Knowledge Store Data Transfer Termination message.

Code 0A25h: Delete Vector Knowledge Store Objects

The Code 0A22h: Delete Vector Knowledge Store Objects message allows the removal of objects from the vector knowledge store. This message allows multiple vector objects to be deleted using a single message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Local Request ID	Byte	N/A	Request identifier to be used when returning data to requesting component
3	Number of Object IDs (p)	Unsigned Short Integer	N/A	
4	Object ID 1	Unsigned Integer	N/A	
…	…	…	…	…
3+p	Object ID p	Unsigned Integer	N/A	
4+p	Region Type	Byte	N/A	Enumeration 0: Point 1: Line

Field #	Name	Type	Units	Interpretation
				2: Polygon 3 – 255: Reserved
5+p	Region Buffer	Float	Meters	
6+p	Number of Feature Classes (n)	Byte	N/A	
7+p	Feature Class 1	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – All
8+p	Feature Class 1 Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
9+p	Feature Class Attribute 1	Varies (see field 4)	Varies with Feature Class	
	…	…	…	…
	…	…	…	…
	…	…	…	…
7+p+3n	Feature Class n	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – All
8+p+3n	Feature Class n Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer

Field #	Name	Type	Units	Interpretation
				3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
9+p+3n	Feature Class Attribute n	Varies (see previous field)	Varies with Feature Class	
10+p+3n	Number of Region Points (m)	Unsigned Short Integer	N/A	0, reserved 1 … 65,535
11+p+3n	Deletion Region Point 1 Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
12+p+3n	Deletion Region Point 1 Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
	…	…	…	…
	…	…	…	…
	Deletion Region Point m Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
	Deletion Region Point m Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	10+p+3n	8+3n	7+p	6+p	5+p	4+p	3

"R" indicates that the bit is reserved.

2.1.8 Dynamic Configuration Subgroup – Codes 0B00-0BFF

There are no Dynamic Configuration Command Class messages.

2.1.9 Payload Subgroup – Codes 0D00-0DFF

Code 0D01h: Set Payload Data Element

The Set Payload Data Element message allows the using component to set command and control interface fields published by the Report Payload Message Interface message. The Command Interface Number is the unique identifier derived from the placement of the interface in the Report Payload Interface message. The Command Interface Number starts at 1 for the first published interface.

Field #	Name	Type	Units	Interpretation
1	Number of Command Interfaces	Byte	Integer	The number (M) of the command and control data elements to be commanded in this message.
2	Command Interface Number	Byte	Integer	The number of the command and control data element as reported in the Report Payload Interface.
3	Value	Defined by Type Code	Defined by Units	Current/Reported Value
…				
2M	Command Interface Number	Byte	Integer	The number of the command and control data element as reported in the Report Payload Interface.

2M+1	Value	Defined by Type Code	Defined by Units	Current/Reported Value

2.1.10 Planning Subgroup – Codes 0E00 – 0EFF

Code 0E00h: Spool Mission

The Spool Mission message is used to send complex missions to lower level components (Mission Spoolers, Drivers, etc.). The amount of data in one Spool Mission message is determined by the receiving component's spooling preferences. For example, if a component can handle ten (10) JAUS messages at a time, then the spooling component shall send a Spool Mission message with no more than ten (10) messages. Spool Mission messages to lower level components (i.e. drivers) shall only contain JAUS messages that are specifically addressed to that lower level component. Hence, a Spool Mission message to a lower level component is reduced to a list, or queue, of JAUS commands addressed to that component.

The Mission ID field shall be a unique ID to allow for spooling and manipulating multiple missions at a time.

The Append Flag field indicates whether the mission is a new mission that shall replace the existing mission or if the mission shall be appended to the existing mission.

The remaining fields define the mission plan, where the mission plan *structure* is an N-ary tree. The Child Index field shall aide in parsing the N-ary tree. It is the byte index into this message where that child's information begins. A Child Task's information begins where its Task ID is defined.

An empty Spool Mission message can be used to indicate a paused, aborted, or completed mission. This technique can be used rather than the Pause and Abort messages to reduce the number of Mission Spooler messages a lower level component must implement. The Mission Spooler can determine which method to use by looking at the services the lower level components support.

Field #	Name	Type	Units	Interpretation
1	Mission ID	Unsigned Short Integer	N/A	Unique Mission ID
2	Append Flag	Byte	N/A	0 = Replace current mission with new mission 1 = Append new mission to current mission
3	Task (1) ID	Unsigned Short Integer	N/A	Unique Task ID
4	Number of Children (n)	Unsigned Short Integer	N/A	Number of children tasks for Task (1)
4+n	Child (1) Index	Unsigned Integer	N/A	The byte index into this message that Task (1)'s Child (1) begins (used for parsing)
…				
5+n	Number of Messages (m)	Unsigned Short Integer	N/A	Number of messages in Task (1)
3+n+3m	UID	Unsigned Short Integer	N/A	Unique identifier for each JAUS message.
4+n+3m	JAUS Message	JAUS Message	N/A	JAUS message (including the 16 byte message header) to be spooled. If Task (1) does not have any messages, this field does not exist
5+n+3m	Blocking	Byte	N/A	Indicates whether the above message is blocking. If Task (1) does not have any messages, this field does not exist. 0 = non-blocking 1 = blocking 2 – 255 Reserved
…				
6+n+3m	Child Task (1) ID	Unsigned Short Integer	N/A	Unique Child Task (1) ID (Field 4+n Child (1) Index contains the byte index to this field)
7+n+3m	Child Task (1)	Unsigned	N/A	Number of children tasks

Field #	Name	Type	Units	Interpretation
	Number of Children	Short Integer		for Task (1) Child Task (1)
8+n+3m	Child of Child Task (1) Index	Unsigned Integer	N/A	The byte index into this message that Task (1)'s Child Task (1) begins (used for parsing)
…	Recursive tree for N Child Tasks			

Code 0E01h: Run Mission

The Run Mission message tells the receiving component to begin execution of the mission with Mission ID.

Field #	Name	Type	Units	Interpretation
1	Mission ID	Unsigned Short Integer	N/A	Unique mission ID

Code 0E02h: Abort Mission

Receipt of the Abort Mission message tells the component to abort, or delete, the mission with Mission ID.

Field #	Name	Type	Units	Interpretation
1	Mission ID	Unsigned Short Integer	N/A	Unique mission ID
2	Task ID	Unsigned Short Integer	N/A	Unique Task ID – to abort an entire mission, Task ID will be 0

Code 0E03h: Pause Mission

Receipt of the Pause Mission message tells the component to pause the mission with Mission ID.

Field #	Name	Type	Units	Interpretation
1	Mission ID	Unsigned	N/A	Unique mission ID

		Short Integer		
2	Task ID	Unsigned Short Integer	N/A	Unique Task ID – to pause an entire mission, Task ID will be 0

Code 0E04h: Resume Mission

Receipt of the Resume Mission message tells the component to resume the mission with Mission ID.

Field #	Name	Type	Units	Interpretation
1	Mission ID	Unsigned Short Integer	N/A	Unique mission ID
2	Task ID	Unsigned Short Integer	N/A	Unique Task ID – to resume an entire mission, Task ID will be 0

Code 0E05h: Remove Messages

The Remove Messages message allows a component to remove any number of messages from an existing mission and task with Mission ID and Task ID. The message(s) to be removed are referenced by their Unique ID. The JAUS message corresponding to each unique ID in the mission shall be removed. If a message to be removed no longer exists in the mission, no action occurs as this indicates that the JAUS message has already been removed from the mission.

Field #	Name	Type	Units	Interpretation
1	Mission ID	Unsigned Short Integer	N/A	Unique ID of the mission the message to be removed is part of
2	Task ID	Unsigned Short Integer	N/A	Unique ID of the task the message to be removed is part of
3	Number of messages (N)	Unsigned Short Integer	N/A	The number of messages to remove from the mission
3+n	UID	Unsigned Short	NA	Unique ID of the message to be removed

		Integer		
…				

Code 0E06h: Replace Messages

The Replace Messages message allows a component to replace any number of messages from the mission and task with Mission ID and Task ID with new messages. The messages to be removed are referenced by their Unique ID. If a message to be removed no longer exists in the mission, this indicates that the JAUS message has already been removed from the mission and no action shall be taken to remove the message.

Field #	Name	Type	Units	Interpretation
1	Mission ID	Unsigned Short Integer	N/A	Unique mission ID
2	Task ID	Unsigned Short Integer	N/A	Unique task ID
3	Remove Number	Unsigned Short Integer	NA	The number of messages to remove
3+n	UID	Unsigned Short Integer	N/A	Unique ID of the message to be replaced
…				
4+n	Replace Number	Unsigned Short Integer	N/A	The number of messages to insert
1+n+3m	UID	Unsigned Short Integer	N/A	Unique ID of the message to be replaced
2+n+3m	JAUS Message	JAUS Message	N/A	Message to insert
3+n+3m	Blocking	Byte	N/A	0 = non-blocking 1 = blocking 2 – 255 Reserved
…				

2.2 Query Class

Query class messages are used to request information from a component. The response to a Query Class message is an Inform class message.

2.2.1 Core Subgroup – Codes 2000-21FF

Code 2001h: Query Component Authority

This message shall cause the receiving component to reply to the requestor with a Code 4001h: Report Component Authority message. All components must be able to respond to this message in any state.

Code 2002h: Query Component Status

This message shall cause the receiving component to reply to the requestor with a Code 4002h: Report Component Status message. All components must be able to respond to this message in any state.

Code 2011h: Query Time

This message shall cause the receiving component to reply to the requestor with a Code 4011h: Report Time message. All components must be able to respond to this message in any state. All times are in Coordinated Universal Time (UTC).

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	**N/A**	See Report Time 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.

Code 200Dh: Query Component Control

This message shall cause the receiving component to reply to the requestor with a Code 400Dh: Report Component Control message. All components must be able to respond to this message in any state.

2.2.2 Event Setup and Control – Codes 21F0-21FF

Code 21F0h: Query Events

The Query Events message is used to request detail on events:

Field 2 indicates the message code in question. If left out, all message codes should be returned. Field 3 indicates the event type to report on. If left out, all event types should be considered. Field 4 indicates a specific Event ID. If left out, all event IDs should be considered.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Message Code	Unsigned Short Integer	N/A	Message code of the Event message that the receiving component is inquiring about
3	Event Type	Byte	N/A	Type of event, enumeration: 0: Periodic 1: Every Change 2: First Change
4	Event ID	Byte	N/A	Event ID returned by Confirm Event for details on specific event.

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	4	3	2

"R" indicates that the bit is reserved.

2.2.3 Communications Subgroup – Codes 2200-23FF

Code 2200h: Query Data Link Status

This message shall cause the receiving component to reply to the requestor with a Code 4200h: Report Data Link Status message.

Field #	Name	Type	Units	Interpretation
1	Data Link ID	Byte	**N/A**	0 … 255

Code 2201h: Query Selected Data Link Status

This message shall cause the receiving component to reply to the requestor with a Code 4201h: Report Selected Data Link message.

Code 2202h: Query Heartbeat Pulse

This message shall cause the receiving component to reply to the sender with a Code 4202h: Report Heartbeat Pulse message.

2.2.4 Platform Subgroup – Codes 2400-25FF

Code 2400h: Query Platform Specifications

This message shall cause the receiving component to reply to the requestor with a Code 4400h: Report Platform Specifications message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Integer	N/A	See Report Platform Specifications 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.

Code 2401h: Query Platform Operational Data

This message shall cause the receiving component to reply to the requestor with a Code 4401h: Report Platform Operational Data message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See Report Platform Operational 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.

Code 2402h: Query Global Pose

This message shall cause the receiving component to reply to the requestor with a Code 4402h: Report Global Pose message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See Report Global Pose 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.

Code 2403h: Query Local Pose

This message shall cause the receiving component to reply to the requestor with a Code 4403h: Report Local Pose message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See Report Local Pose 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.

Code 2404h: Query Velocity State

This message shall cause the receiving component to reply to the requestor with a Code 4404h: Report Velocity State message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See Report Velocity State 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.

Code 2405h: Query Wrench Effort

This message shall cause the receiving component to reply to the requestor with a Code 4405h: Report Wrench Effort message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See Report Wrench Effort 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.

Code 2406h: Query Discrete Devices

This message shall cause the receiving component to reply to the requestor with a Code 4406h: Report Discrete Devices message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See Report Discrete Devices 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.

Code 2407h: Query Global Vector

This message shall cause the receiving component to reply to the requestor with a Code 4407h: Report Global Vector message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See Report Global Vector 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.

Code 2408h: Query Local Vector

This message shall cause the receiving component to reply to the requestor with a Code 4408h: Report Local Vector message.

Code 240Ah: Query Travel Speed

This message shall cause the receiving component to reply to the requestor with a Code 440Ah: Report Travel Speed message.

Code 240Bh: Query Waypoint Count

This message shall cause the receiving component to reply to the requestor with a Code 440Bh: Report Waypoint Count message.

Code 240Ch: Query Global Waypoint

This message shall cause the receiving component to reply to the requestor with a Code 440Ch: Report Global Waypoint message. Field #1 specifies the waypoint to be returned in the Report Global Waypoint message.

Field #	Name	Type	Units	Interpretation
1	Waypoint Number	Unsigned Short Integer	N/A	Lower Limit = 0 Upper Limit = 65,535

Code 240Dh: Query Local Waypoint

This message shall cause the receiving component to reply to the requestor with a Code 440Dh: Report Local Waypoint message. Field #1 specifies the waypoint to be returned in the Report Local Waypoint message.

Field #	Name	Type	Units	Interpretation
1	Waypoint Number	Unsigned Short Integer	N/A	Lower Limit = 0 Upper Limit = 65,535

Code 240Eh: Query Path Segment Count

This message shall cause the receiving component to reply to the requestor with a Code 440Eh: Report Path Segment Count message.

Code 240Fh: Query Global Path Segment

This message shall cause the receiving component to reply to the requestor with a Code 440Fh: Report Global Path Segment message. Field #1 specifies the path segment number to be returned in the Report Global Path Segment message.

Field #	Name	Type	Units	Interpretation
1	Path Segment Number	Unsigned Short Integer	N/A	Lower Limit = 0 Upper Limit = 65,535

2	Presence Vector	Byte	N/A	See Report Global Path Segment 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.

Code 2410h: Query Local Path Segment

This message shall cause the receiving component to reply to the requestor with a Code 4410h: Report Local Path Segment message. Field #2 specifies the path segment to be returned in the Report Local Path Segment message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See Report Local Path Segment Message.
2	Path Segment Number	Unsigned Short Integer	N/A	Lower Limit = 0 Upper Limit = 65,535

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.

2.2.5 Manipulator Subgroup – Codes 2600-27FF

Code 2600h: Query Manipulator Specifications

This message shall cause the primitive manipulator component to reply to the requestor with a Code 4600h: Report Manipulator Specifications message.

Code 2601h: Query Joint Efforts

This message shall cause the receiving component to reply to the requestor with a Code 4601h: Report Joint Efforts message.

Code 2602h: Query Joint Positions

This message shall cause the receiving component to reply to the requestor with a Code 4602h: Report Joint Positions message.

Code 2603h: Query Joint Velocities

This message shall cause the receiving component to reply to the requestor with a Code 4603h: Report Joint Velocities message.

Code 2604h: Query Tool Point

This message shall cause the receiving component to reply to the requestor with a Code 4604h: Report Tool Point message.

Code 2605h: Query Joint Force/Torques

This message shall cause the receiving component to reply to the requestor with a Code 4605h: Report Joint Force/Torques message.

2.2.6 Environment Sensor Subgroup – Codes 2800-29FF

Code 2800h: Query Camera Pose

This message shall cause the receiving component to reply to the requestor with a Code 4800h: Report Camera Pose message. Field #2 specifies which camera pose to return in the Report Camera Pose message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See Report Camera Pose Message
2	Camera ID	Byte	N/A	1 … 255, 0 is Reserved

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.

Code 2801h: Query Camera Count

This message shall cause the receiving component to reply to the requestor with a Code 4801h: Report Camera Count message.

Code 2802h: Query Relative Object Position

This message shall cause the receiving component to reply to the requestor with a Code 4802h: Report Relative Object Position message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See Report Relative Object Position 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.

Code 2804h: Query Selected Camera

This message shall cause the receiving component to reply to the requestor with a Code 4804h: Report Selected Camera message.

Code 2805h: Query Camera Capabilities

This message shall cause the receiving component to reply to the requestor with a Code 4805h: Report Camera Capabilities message to report the capabilities of the selected camera.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See Report Camera Capabilities Message.
2	Camera ID	Byte	N/A	1 … 255 0 is Reserved

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.

Code 2806h: Query Camera Format Options

This message shall cause the receiving component to reply to the requestor with a Code 4806h: Report Camera Format Options message to report the output options of the selected camera.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See Report Camera Format Options Message.
2	Camera ID	Byte	N/A	1 … 255 0 is Reserved

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.

Code 2807h: Query Image

The Query Image message can be used to retrieve one image at a time from the currently selected camera. For streaming video, it is recommended that the Service Connection be used. For streaming video outside of a JAUS controlled transmission such as a dedicated analog RF link, the JAUS message set shall be used to control the flow. JAUS messages for this control are provided by the core message set: Resume and Standby. The response to the Query Image message shall be a Code 4807h: Report Image message.

2.2.7 World Model Subgroup – Codes 2A00-2AFF

Code 2A21h: Query Vector Knowledge Store Feature Class Metadata

The Code 2A21h: Query Vector Knowledge Store Feature Class Metadata message shall cause the receiving Vector Knowledge Store to reply to the requestor with the Code 4A21h: Report Vector Knowledge Store Feature Class Metadata message with the requested data.

There is a single field associated with this message. This field specifies the feature class metadata to return in the reply. There is also an option to return metadata for all feature classes present in the queried vector knowledge store.

Field #	Name	Type	Units	Interpretation
1	Feature Class	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – All

Code 2A22h: Query Vector Knowledge Store Bounds

The Code 2A22h: Query Vector Knowledge Store Bounds message is used to request the spatial extents of a single feature class or of all feature classes within a vector

knowledge store. The knowledge store shall respond with the Code 4A23h: Report Vector Knowledge Store Bounds message.

Field #	Name	Type	Units	Interpretation
1	Local Request ID	Byte	N/A	Request identifier to be used when returning data to requesting component
2	Feature Class	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – All Feature Classes

Code 2A23h: Query Vector Knowledge Store Objects

The Code 2A20h: Query Vector Knowledge Store Objects message allows access to objects within the vector knowledge store. The message allows the requestor to ask for all data matching the query to be returned or to have the knowledge store simply return the number of object that match the query. This feature is controlled using Field 2 of the message below.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Presence Vector	Byte	N/A	Query Response Presence Vector. See Code 4A220h: Report Vector Knowledge Store Objects message for Presence Vector format.
3	Local Request ID	Byte	N/A	Request identifier to be used when returning data to requesting component
4	Number of Object IDs (p)	Unsigned Short Integer	N/A	
5	Object ID 1	Unsigned Integer	N/A	
…	…	…	…	…

Field #	Name	Type	Units	Interpretation
4+p	Object ID p	Unsigned Integer	N/A	
5+p	Region Type	Byte	N/A	Enumeration 0: Point 1: Line 2: Polygon 3 – 255: Reserved
6+p	Region Buffer	Float	Meters	
7+p	Number of Feature Classes (n)	Byte	N/A	
8+p	Feature Class 1	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – All
9+p	Feature Class 1 Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
10+p	Feature Class Attribute 1	Varies (see field 4)	Varies with Feature Class	
	…	…	…	…
	…	…	…	…
	…	…	…	…
8+p+3n	Feature Class n	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – All
9+p+3n	Feature Class n Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer

Field #	Name	Type	Units	Interpretation
				6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
10+p+3n	Feature Class Attribute n	Varies (see previous field)	Varies with Feature Class	
11+p+3n	Number of Region Points (m)	Unsigned Short Integer	N/A	0, reserved 1 … 65,535
12+p+3n	Query Region Point 1 Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
13+p+3n	Query Region Point 1 Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
	…	…	…	…
	…	…	…	…
	Query Region Point m Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
	Query Region Point m Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	11+p+3n	9+p	8+p	7+p	6+p	5+p	4

"R" indicates that the bit is reserved.

2.2.8 Dynamic Configuration Subgroup – Codes 2B00-2BFF

Dynamic Configuration is used by all components within a JAUS system. The dynamic configuration process can be broken down into the following: discovery, identifier allocation, and capability publication. Discovery is the process of determining the existence of other components, nodes, and subsystems within the system. Identifier

allocation is the process of assigning unique identifiers for each component, node, and subsystem within the system. And, capability publication is the process of conveying a component's capability to any other component within the system. In RA 3.3, only the discovery and capabilities problems are addressed.

2.2.6.1 Discovery

The process of discovery is conducted at both the node level and the subsystem level. At the node level, a Report Heartbeat Pulse message must be sent to all nodes within the subsystem. This message must be persistent at a 1 Hz rate from all nodes within the subsystem. The JAUS destination identifier must be set to X:255:1:1, where X is the subsystem identifier. The JAUS source identifier of the Report Heartbeat Pulse message must be set to the component's identifier that is capable of responding to the messages Query Identification and Query Configuration at the node level. When a Report Heartbeat Pulse message is received by another node within the subsystem, the receiving node may query for the node's identification and configuration information. **Figure 2-3** shows a typical flow of messages for discovering the configuration of a new node, where nodes A, B, and C are all on the same subsystem.

The subsystem level discovery is very similar to the node level discovery. A Report Heartbeat Pulse message must be sent to all nodes within the system, again at a 1Hz rate. The JAUS destination identifier must be set to 255:255:1:1, and the JAUS source identifier must be set to the component's identifier that is capable of responding to the messages Query Identification and Query Configuration at the subsystem level. When a Report Heartbeat Pulse message is received by a node from a different subsystem, the receiving node may query for the subsystem's identification and configuration information. **Figure 2-3** can be used again to show the typical flow of messages for discovering the configuration of a new subsystem and keeping it current, where nodes A, B, and C are all on different subsystems.

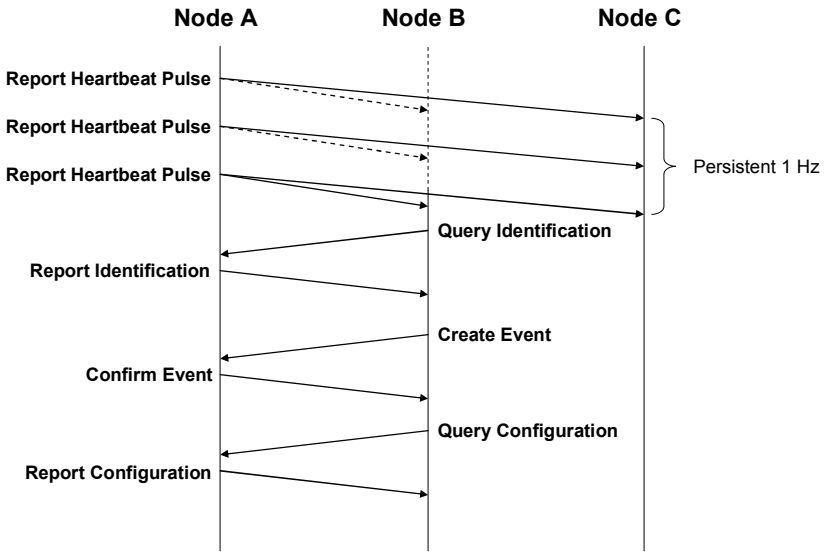

Figure 2-3: Configuration Discovery Message Flow

Code 2B00: Query Identification

This message shall request the identification summary of a Subsystem, Node, or Component.

Field #	Name	Type	Units	Interpretation
1	Query Type	Byte	N/A	0: Reserved 1: System Identification 2: SS Identification 3: Node Identification 4: Component Identification 5 – 255: Reserved

Code 2B01: Query Configuration

This message shall request the configuration summary of a subsystem or node. For example, to get the complete configuration of a subsystem, field 1 shall be set to 2.

Field #	Name	Type	Units	Interpretation
1	Query Field	Byte	N/A	0: Reserved 1: Reserved 2: Subsystem Configuration 3: Node Configuration 4 – 255: Reserved

Code 2B02: Query Subsystem List

This message shall request the Report Subsystem List message. There are no data fields associated with this message.

Code 2B03: Query Services

This message allows a component to request the capabilities of another component. Receipt of this message shall cause the receiving component to respond with Code 4B03 Report Services. There are no data fields associated with this query message.

2.2.9 Payload Subgroup – Codes 2D00-2DFF

Code 2D00h: Query Payload Interface Message

No message data beyond header.

Query Payload Interface must be responded to with either

NAK implies that the component/node is not a payload but is a Component/Node with standard JAUS message capabilities.

Or

Code 4D00: Report Payload Interface Message

Code 2D01h: Query Payload Data Element

Querying of a payload data element must specify the specific information interface number as reported in the Report Payload Interface Message. Querying of command and control data elements is not supported. The Information Interface Number is the unique identifier derived from the placement of the interface in the Report Payload

Interface message. The Information Interface Number starts at 1 for the first published informational interface.

Field #	Name	Type	Units	Interpretation
1	Number Information Interfaces	Byte	Integer	The number (M) of information data elements being requested by the Payload controller.
2	Information Interface Number	Byte	Integer	The number of the information data element as reported in the Report Payload Interface.
…				
M+1	Information Interface Number	Byte	Integer	The number of the information data element as reported in the Report Payload Interface.

2.2.10 Planning Subgroup – Codes 2E00 – 2EFF

Code 2E00h: Query Spooling Preference

The spooling component shall send a Query Spooling Preference message to every component utilized in a mission plan that is ready to be spooled. Spooling preferences indicate how components want the commands spooled to them. The receiving component shall respond with a Report Spooling Preference message if it supports the Query Spooling message.

Code 2E01h: Query Mission Status

Receipt of the Query Mission Status message shall result in the receiving component responding with the Report Mission Status message. The Query Type can be Mission, Task, or Message. The Presence Vector allows you to narrow your query to a particular Mission, Task, or Message by indicating the Mission, Task, or Message ID to query the status on. Examples:

1. Query the status of any/all Missions:
    a. Presence Vector: Bit 0, 1, 2 = 0
    b. Type = 0
2. Query the status of Mission 5:
    a. Presence Vector: Bit 0 = 1
    b. Type = 0

c. Mission ID = 5

3. Query the status of any/all Messages:
   a. Presence Vector: Bit 0, 1, 2 = 0
   b. Type = 2

Note: Multiple Report Mission Status messages may be sent in response to a single Query Mission Status message.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	Bits 0 – 2 correspond to fields 3 thru 5 respectively
2	Type	Byte	N/A	0 = Mission 1 = Task 2 = Message 3 – 255 Reserved
3	Mission ID	Unsigned Short Integer	N/A	Unique Mission ID
4	Task ID	Unsigned Short Integer	N/A	Unique Task ID (If Type = 0, Task ID = 0)
5	UID	Unsigned Short Integer	N/A	Unique Message ID (If Type = 0 or 1, Message ID = 0)

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	R	R	5	4	3

"R" indicates that the bit is reserved.

2.3 Inform Class

Inform class messages are used to report information from a component. The Inform Class message is in response to a Query Class message.

2.3.1 Core Subgroup – Codes 4001-41FF

Code 4001h: Report Component Authority

This message informs the receiver of the current command authority of the sending component.

Field #	Name	Type	Units	Interpretation
1	Authority Code	Byte	N/A	Authority 0-255

Code 4002h: Report Component Status

Field #1 of the status message indicates the primary status code of the component. This data field is segmented into two four-bit sections. The first four-bit section (bits 0 – 3) is interpreted as a four bit, unsigned integer. The second four-bit section is available for vendor specific use. JAUS defines six generic primary status conditions. The remaining primary status codes of the first four-bit segment (values 6 through 15) are reserved.

Field #2 of the status message is a secondary status code, which is used to further qualify the primary status code. This data field is segmented into two sixteen-bit sections. The secondary status code is interpreted on a bit-by-bit basis, as opposed to the primary status code, which is interpreted as an integer value. The bit-field interpretation allows multiple secondary status codes to be set within the same status message. For instance, in the emergency state, the secondary status bits could indicate specific emergency conditions that have been set. Secondary status bits are defined on a component-by-component basis.

Field #	Name	Type	Units	Interpretation
1	Primary Status Code	Byte	N/A	Bits 0 – 3: 0 = Initialize 1 = Ready 2 = Standby 3 = Shutdown 4 = Failure 5 = Emergency Bits 4 – 7: Available

2	Secondary Status Code	Unsigned Integer	N/A	Bits 0 – 15 Reserved Each bit corresponds to specific conditions for each primary code. Bits 16 – 31: Available

Reminder: Developers can only use the "Available" portions of the status message data for debug and testing. The JAUS specified codes must also be supported, and the vendor specific codes shall not replace information already specified within JAUS.

Code 4011h: Report Time

This message provides the receiver with the current time as managed by the sending component. The message data and mapping of the presence vector of this message are identical to Code 0011h: Set Time. All times are in Coordinated Universal Time (UTC).

Code 400Dh: Report Component Control

This message informs the receiver of the current component that has control. The ID fields shall be set to zero (0) if there is no component in control.

Field #	Name	Type	Units	Interpretation
1	Subsystem ID	Byte	N/A	0-254
2	Node ID	Byte	N/A	0-254
3	Component ID	Byte	N/A	0-254
4	Instance ID	Byte	N/A	0-254
5	Authority Code	Byte	N/A	Authority 0-255

2.3.2 Event Setup and Control – Codes 41F0-41FF

Code 41F0h: Report Events

Report Events

Field #	Name	Type	Units	Interpretation
1	Count	Byte	N/A	Number of reported events.
2	Presence Vector	Byte	N/A	See mapping table below

Field #	Name	Type	Units	Interpretation
3	Message Code	Unsigned short integer	N/A	Message code of the Event message.
4	Event Type	Byte	N/A	Type of event, enumeration: 0: Periodic (SC) 1: Every Change 2: First Change 3: First change in and out of boundary 4: Periodic w/o replacement 5: One time, on demand
5	Event Boundary	Byte	N/A	Boundary condition on event trigger, enumeration: 0: Equal 1: Not Equal 2: Inside Inclusive 3: Inside Exclusive 4: Outside Inclusive 5: Outside Exclusive 6: Greater than or Equal 7: Strictly Greater than 8: Less than or Equal 9: Strictly Less than
6	Limit Data Field	Byte	N/A	Field from Report message to base trigger limit on
7	Lower Limit Data Field Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
8	Lower Limit	Variable-type	varies	Lower limit for trigger condition, used for Inside, Outside, and Low Event Boundary Types

Field #	Name	Type	Units	Interpretation
9	Upper Limit Data Field Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
10	Upper Limit	Variable-type	varies	Upper limit for trigger condition, used for Inside, Outside, and High Event Boundary Types
11	State Data Field Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
12	State	Variable-type	varies	Trigger value used for Equal Event Boundary Type. Typically used for discrete-type events.
13	Event ID	Byte	N/A	Unique Identifier to of existing event to update
14	Query Message Size	Unsigned Integer	Bytes	The size of the following JAUS Query message body
15	Query Message	JAUS Message	N/A	The JAUS Query message body to be used by the receiving component to generate the Report message(s).

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0

Vector to Data Field Mapping for Above Command								
Data Field	R	14-15	13	11/12	9/10	7/8	6	5

"R" indicates that the bit is reserved.

Code 41F1h: Event

The Event message is sent when event is triggered. It includes the Event ID and a sequence number to allow the client to keep track of event processing.

Field #	Name	Type	Units	Interpretation
1	Event ID	Byte	N/A	Unique identifier of the enclosed event
2	Message Code	Unsigned short integer	N/A	Message code of the enclosed Event message.
3	Sequence Number	Byte	N/A	Sequential count of the number of times this event has been issued
4	Message Size	Unsigned Integer	Bytes	The size of the enclosed message
5	Report Message	JAUS Message	N/A	The JAUS Report message body

2.3.3 Communications Subgroup – Codes 4200-43FF

Code 4200h: Report Data Link Status

The Data Link Status message provides the requesting component the state of the requested data link.

Field #	Name	Type	Units	Interpretation
1	Data Link ID	Byte	N/A	0 … 255
2	Data Link State	Byte	N/A	0 = Off 1 = On 2 = Standby 3 = Loss of Command (link has been lost) 4-255 Reserved

Code 4201h: Report Selected Data Link Status

This message provides the requesting component with the current data link ID.

Field #	Name	Type	Units	Interpretation
1	Data Link ID	Byte	N/A	0 ... 255

Code 4202h: Report Heartbeat Pulse

This message notifies the receiver that the sender is alive.

2.3.4 Platform Subgroup – Codes 4400-45FF

Code 4400h: Report Platform Specifications

This message provides the receiver specifications of the platform including its name, dimensions, location of its center of gravity, turning radius, wheel base, track width, static pitch and roll over angles, and maximum velocities and angular rates the platform is capable of achieving.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Integer	N/A	See mapping table below
2	Mobility Platform Name	String	N/A	15 Character fixed length string
3	Front	Unsigned Short Integer	Meters	Distance from the vehicle origin to the bounding planes of the mobility platform. Scaled Integer Lower Limit = 0 Upper Limit = 32.767
4	Back			
5	Right			
6	Left			
7	Bottom			
8	Top			
9	Xcg	Unsigned Short	Meters	Center of Gravity coordinates with
10	Ycg			

Field #	Name	Type	Units	Interpretation
		Integer		respect to mobility platform origin.
11	Zcg			Scaled Integer Lower Limit = 0 Upper Limit = 32.767
12	Turning Radius	Unsigned Short Integer	Meters	Minimum turning radius of mobility platform. Scaled Integer Lower Limit= 0 Upper Limit = 65.535
13	Wheel Base	Unsigned Short Integer	Meters	The distance between the fore-most and aft-most axle. Scaled Integer Lower Limit = 0 Upper Limit = 65.535
14	Track Width	Unsigned Short Integer	Meters	The distance between the centerlines of the left-most and right-most driving devices. Scaled Integer Lower Limit = 0 Upper Limit = 65.535
15	Static Pitch-over	Unsigned Short Integer	Radians	Incline angle at which the mobility platform will pitch over backwards. Scaled Integer Lower Limit = 0 Upper Limit = 2.56
16	Static Roll-over	Unsigned Short Integer	Radians	Roll angle at which the mobility platform will roll over sideways at zero velocity. Scaled Integer Lower Limit = 0 Upper Limit = 2.56

Field #	Name	Type	Units	Interpretation
17	Maximum Velocity X	Unsigned Short Integer	Meters per Second	Scaled Integer Lower Limit = 0 Upper Limit = 65.534
18	Maximum Velocity Y			
19	Maximum Velocity Z			
20	Maximum Roll Rate	Unsigned Short Integer	Radians per Second	Scaled Integer Lower Limit = 0 Upper Limit = 32.767
21	Maximum Pitch Rate			
22	Maximum Yaw Rate			

Vector to Data Field Mapping for Above Command								
Vector Bit	31	30	29	28	27	26	25	24
Data Field	R	R	R	R	R	R	R	R
Vector Bit	23	22	21	20	19	18	17	16
Data Field	R	R	R	22	21	20	19	18
Vector Bit	15	14	13	12	11	10	9	8
Data Field	17	16	15	14	13	12	11	10
Vector Bit	7	6	5	4	3	2	1	0
Data Field	9	8	7	6	5	4	3	2

"R" indicates that the bit is reserved.

Code 4401h: Report Platform Operational Data

This message provides the receiver with a variety of operational data from the platform including engine temperature, odometer reading, battery voltage, fuel level, and oil pressure if these are available from the platform.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table that follows
2	Engine Temperature	Short Integer	Degrees Celsius	Scaled Integer Lower Limit = -75° C Upper Limit = 180° C
3	Odometer	Unsigned Integer	Meters	Cumulative distance traveled by vehicle
4	Battery Voltage	Byte	Percent	Scaled Integer Lower Limit = 0%

				Upper Limit = 127%
5	Percentage Fuel Level	Byte	Percent	Scaled Integer Lower Limit = 0% Upper Limit = 100%
6	Percentage Oil Pressure	Byte	Percent	Scaled Integer Lower Limit = 0% Upper Limit = 127%

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	6	5	4	3	2

"R" indicates that the bit is reserved.

Code 4402h: Report Global Pose

The Report Global Pose message provides the position and attitude of the platform. The position of the platform is given in latitude, longitude, and altitude, in accordance with the WGS 84 standard. All times are in Coordinated Universal Time (UTC).

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See mapping table that follows.
2	Latitude (WGS 84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
3	Longitude (WGS 84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
4	Altitude, (ASL)	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
5	Position RMS	Unsigned Integer	Meters	A RMS value indicating the validity of the position data. Scaled integer Lower Limit = 0 Upper Limit = 100
6	ϕ (Roll)	Short Integer	Radians	Scaled Integer Lower Limit = $-\pi$ Upper Limit = π
7	θ (Pitch)			
8	ψ (Yaw)			
9	Attitude RMS	Unsigned Short Integer	Radians	A RMS value indicating the validity of the Roll/Pitch/Yaw data

Field #	Name	Type	Units	Interpretation
				Scaled integer Lower Limit = 0 Upper Limit = π
10	Time Stamp	Unsigned Integer	N/A	Bits 0-9: milliseconds, range 0...999 Bits 10-15: Seconds, range 0...59 Bits 16 – 21: Minutes, range 0...59 Bits 22-26: Hour (24 hour clock), range 0..23 Bits 27-31: Day, range 1...31

Vector to Data Field Mapping for Above Command								
Vector Bit	15	14	13	12	11	10	9	8
Data Field	R	R	R	R	R	R	R	10
Vector Bit	7	6	5	4	3	2	1	0
Data Field	9	8	7	6	5	4	3	2

"R" indicates that the bit is reserved.

Code 4403h: Report Local Pose

The Local Pose message provides the position and attitude of the platform relative to a local reference frame. All times are in Coordinated Universal Time (UTC).

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See mapping table that follows.
2	X	Integer	Meters	Scaled Integer Lower Limit = -100,000 Upper Limit = 100,000
3	Y	Integer	Meters	Scaled Integer Lower Limit = -100,000 Upper Limit = 100,000
4	Z	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
5	Position RMS	Unsigned Integer	Meters	A RMS value indicating the validity of the position data. Scaled Integer Lower Limit = 0 Upper Limit = 100

Field #	Name	Type	Units	Interpretation
6	ϕ (Roll)	Short Integer	Radians	Scaled Integer Lower Limit = $-\pi$ Upper Limit = π
7	θ (Pitch)			
8	ψ (Yaw)			
9	Attitude RMS	Unsigned Short Integer	Radians	A RMS value indicating the validity of the Roll/Pitch/Yaw data Scaled Integer Lower Limit = 0 Upper Limit = π
10	Time Stamp	Unsigned Integer	N/A	Bits 0-9: milliseconds, range 0...999 Bits 10-15: Seconds, range 0...59 Bits 16 – 21: Minutes, range 0...59 Bits 22-26: Hour (24 hour clock), range 0..23 Bits 27-31: Day, range 1…31

Vector to Data Field Mapping for Above Command								
Vector Bit	15	14	13	12	11	10	9	8
Data Field	R	R	R	R	R	R	R	10
Vector Bit	7	6	5	4	3	2	1	0
Data Field	9	8	7	6	5	4	3	2

"R" indicates that the bit is reserved.

Code 4404h: Report Velocity State

The Report Velocity State message provides the linear velocity and rotational rate of the platform.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See mapping table that follows.
2	Velocity X	Integer	Meters per Second	Scaled Integer Lower Limit = -65.534 Upper Limit = 65.534
3	Velocity Y			
4	Velocity Z			

Field #	Name	Type	Units	Interpretation
5	Velocity RMS	Unsigned Integer	Meters per Second	A value indicating the validity of the velocity data. Scaled Integer Lower Limit = 0 Upper Limit = 100
6	Roll Rate	Short Integer	Radians per Second	Scaled Integer Lower Limit = -32.767 Upper Limit = 32.767
7	Pitch Rate			
8	Yaw Rate			
9	Rate RMS	Unsigned Short Integer	Radians per Second	A value indicating the validity of the rotational rate data. Scaled Integer Lower Limit = 0 Upper Limit = π
10	Time Stamp	Unsigned Integer	N/A	Bits 0-9: milliseconds, range 0..999 Bits 10-15: Seconds, range 0..59 Bits 16 – 21: Minutes, range 0..59 Bits 22-26: Hour (24 hour clock), range 0..23 Bits 27-31: Day, range 1..31

Vector to Data Field Mapping for Above Command								
Vector Bit	15	14	13	12	11	10	9	8
Data Field	R	R	R	R	R	R	R	10
Vector Bit	7	6	5	4	3	2	1	0
Data Field	9	8	7	6	5	4	3	2

"R" indicates that the bit is reserved.

Code 4405h: Report Wrench Effort

This message provides the receiver the current values of the commanded wrench effort. The message data and mapping of the presence vector for the Report Wrench Effort message are identical to Code 0405h: Set Wrench Effort.

Code 4406h: Report Discrete Devices

This message provides the receiver the current values of the commanded discrete devices. The data fields and presence vector mapping of this message are identical to Code 0406h: Set Discrete Devices.

Code 4407h: Report Global Vector

This message provides the receiver the current values of the commanded global vector. The message data of this message is identical to Code 0407h: Set Global Vector.

Code 4408h: Report Local Vector

This message provides the receiver the current values of the commanded local vector. The message data this message is identical to Code 0408h: Set Local Vector.

Code 440Ah: Report Travel Speed

This message provides the receiver the current values of the commanded travel speed. The message data of this message is identical to Code 040Ah: Set Travel Speed.

Code 440Bh: Report Waypoint Count

This message provides the receiver with the total number of waypoints reported by Report Global Waypoint or Report Local Waypoint messages.

Field #	Name	Type	Units	Interpretation
1	Waypoint Count	Unsigned Short Integer	N/A	0 … 65,535

Code 440Ch: Report Global Waypoint

This message provides the receiver with the current values of the requested waypoint as specified by the data in Code 240Ch: Query Global Waypoint. The message data and mapping of the presence vector of this message are identical to Code 040Ch: Set Global Waypoint.

Code 440Dh: Report Local Waypoint

This message provides the receiver with the current values of the requested waypoint as specified by the data in Code 240Dh: Query Local Waypoint. The message data and mapping of the presence vector of this message are identical to Code 040Dh: Set Local Waypoint.

Code 440Eh: Report Path Segment Count

This message provides the receiver with the total number of path segments reported by Report Global Path Segment or Report Local Path Segment messages.

Field #	Name	Type	Units	Interpretation
1	Path Segment Count	Unsigned Short Integer	N/A	0 … 65,535

Code 440Fh: Report Global Path Segment

This message provides the receiver the current values of the requested path segment specified by the data in Code 240Fh: Query Global Path Segment. The message data and mapping of the presence vector of this message are identical to Code 040Fh: Set Global Path Segment.

Code 4410h: Report Local Path Segment

This message provides the receiver the current values of the requested path segment specified by the data in Code 2410h: Query Local Path Segment. The message data and mapping of the presence vector of this message are identical to Code 0410h: Set Local Path Segment.

2.3.5 Manipulator Subgroup – Codes 4600-47FF

Code 4600h: Report Manipulator Specifications

This message provides the specifications of the manipulator including the number of joints, the link lengths, twist angles, offset or joint angles, minimum and maximum value for each joint, and minimum and maximum speed for each joint.

Field #	Name	Type	Units	Interpretation
1	Number of Joints	Byte	N/A	1 … 255 0 is Reserved For a value of n the message data area size in bytes is: (36 + 13(n-1)) bytes.

Field #	Name	Type	Units	Interpretation
2	Joint n Type	Byte	N/A	Joint type of the last joint of the manipulator. 1 = revolute, 2 = prismatic
3	Joint n – Offset/Joint Angle	Unsigned Short Integer	Radians or Millimeters	Prismatic joint, mm Revolute joint, value × 10^{-3} radians
4	Joint n – Min value	Unsigned Short Integer	Radians or Millimeters	Prismatic joint, mm Revolute joint, value × 10^{-3} radians.
5	Joint n – Max value			
6	Joint n – Max velocity	Unsigned Short Integer	Radians per Second or Millimeters per Second	Prismatic joint, mm/sec Revolute joint, value × 10^{-3} radians/sec.
7	manipulator coordinate sys. x	Integer	Meters	x coordinate of origin of manipulator coordinate system measured with respect to vehicle coordinate system Scaled integer: Lower limit = -30 m Upper limit = +30 m
8	manipulator coordinate sys. y	Integer	Meters	y coordinate of origin of manipulator coordinate system measured with respect to vehicle coordinate system Scaled integer: Lower limit = -30 m Upper limit = +30 m
9	manipulator coordinate sys. z	Integer	Meters	z coordinate of origin of manipulator coordinate system measured with respect to vehicle coordinate system Scaled integer: Lower limit = -30 m Upper limit = +30 m

Field #	Name	Type	Units	Interpretation
10	d component of unit quaternion q	Integer	N/A	quaternion q defines the orientation of the manipulator coordinate system measured with respect to the vehicle coordinate system Scaled integer: Lower limit = -1 Upper limit = +1
11	a component of unit quaternion q	Integer	N/A	see field 10
12	b component of unit quaternion q	Integer	N/A	see field 10
13	c component of unit quaternion q	Integer	N/A	see field 10
14	Joint 1 Type	Byte	N/A	1 = revolute, 2 = prismatic
14a	Link a_{12} – Link Length	Unsigned Short Integer	Millimeters	Link Length
14b	Link a_{12} – Twist Angle	Unsigned Short Integer	Radians	Twist Angle - value × 10^{-3} radians
14c	Joint 1 – Offset/Joint Angle	Unsigned Short Integer	Radians or Millimeters	Prismatic joint, mm Revolute joint, value × 10^{-3} radians
14d	Joint 1 – Min value	Unsigned Short Integer	Radians or Millimeters	Prismatic joint, mm Revolute joint, value × 10^{-3} radians.
14e	Joint 1 – Max value			
14f	Joint 1 – Max velocity	Unsigned Short Integer	Radians per Second or Millimeters per Second	Prismatic joint, mm/sec Revolute joint, value × 10^{-3} radians/sec.
…				
n-1	Joint (n-1) Type	Byte	N/A	1 = revolute, 2 = prismatic
(n-1)a	Link $a_{(n-1)n}$ – Link Length	Unsigned Short Integer	Millimeters	Link Length
(n-1)b	Link $a_{(n-1)n}$ – Twist Angle	Unsigned Short Integer	Radians	Twist Angle - value × 10^{-3} radians

Field #	Name	Type	Units	Interpretation
(n-1)c	Joint (n-1) – Offset/Joint Angle	Unsigned Short Integer	Radians or Millimeters	Prismatic joint, value × 10^{-3} radians. Revolute joint, mm
(n-1)d	Joint (n-1) – Min value	Unsigned Short Integer	Radians or Millimeters	Prismatic joint, mm Revolute joint, value × 10^{-3} radians.
(n-1)e	Joint (n-1) – Max value			
(n-1)f	Joint (n-1) – Max velocity	Unsigned Short Integer	Radians per Second or Millimeters per Second	Prismatic joint, mm/sec Revolute joint, value × 10^{-3} radians/sec.

Code 4601h: Report Joint Efforts

This message provides the receiver with the current values of the commanded joint effort. The message data for the Report Joint Efforts message are identical to Code 0601h: Set Joint Effort.

Code 4602h: Report Joint Positions

This message provides the receiver with the current values of the joint positions. The message data for the Report Joint Positions message is identical to Code 0602h: Set Joint Positions.

Code 4603h: Report Joint Velocities

This message provides the receiver with the current values of the joint velocities. The message data for the Report Joint Velocities message is identical to Code 0603h: Set Joint Velocities.

Code 4604h: Report Tool Point

This message provides the receiver with the current values of the joint positions. The message data for the Report Joint Positions message is identical to Code 0604h: Set Tool Point.

Code 4605h: Report Joint Force/Torques

Field #1 indicates the number of joint force/torques reported in this message. This message reports the joint force/torques values.

Field #	Name	Type	Units	Interpretation
1	Num Joints	Byte	N/A	1 ... 255 0 is Reserved
2	Joint 1 –force or torque	Integer	Newtons or Newton Meters	If revolute joint Scaled integer: Lower limit = -1000 Nm Upper limit = +1000 Nm If prismatic joint Scaled integer: Lower limit = -500 N Upper limit = +500 N
3 ... n	…			
n + 1	Joint n –force or torque	Integer	Newtons or Newton Meters	see field 2

2.3.6 Environment Sensor Subgroup – Codes 4800-49FF

Code 4800h: Report Camera Pose

This message provides the receiver with the current values of the camera pose.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short Integer	N/A	See mapping table below
2	Camera ID	Byte	N/A	1 … 255 0 is Reserved
3	Camera Name	String	N/A	15 character fixed length string.
4	X Camera Origin	Short Integer	Meters	Scaled Integer Lower Limit = -32.767 Upper Limit = 32.767 Location of the camera's coordinate frame origin with respect to the vehicle.
5	Y Camera Origin			
6	Z Camera Origin			
7	X Camera Axis Dir. Cosine X	Short Integer	N/A	The Camera Axis Direction Cosines

Field #	Name	Type	Units	Interpretation
8	X Camera Axis Dir. Cosine Y			define the orientation of the camera coordinate frame with respect to the vehicle coordinate frame. Scaled Integer Lower Limit = -1.0 Upper Limit = 1.0
9	X Camera Axis Dir. Cosine Z			
10	Z Camera Axis Dir. Cosine X			
11	Z Camera Axis Dir. Cosine Y			
12	Z Camera Axis Dir. Cosine Z			

Vector to Data Field Mapping for Above Command								
Vector Bit	15	14	13	12	11	10	9	8
Data Field	R	R	R	R	R	R	12	11
Vector Bit	7	6	5	4	3	2	1	0
Data Field	10	9	8	7	6	5	4	3

"R" indicates that the bit is reserved.

Code 4801h: Report Camera Count

Each camera under the control of the visual component has a unique ID. The camera IDs range from one to the number of cameras, up to a maximum of 255. For example, if the Camera Count message returns a value of 3, then the camera IDs are 1, 2, and 3. The location and orientation of each camera is determined by information returned by the camera position and orientation parameters message.

Field #	Name	Type	Units	Interpretation
1	Camera Count	Byte	N/A	1 … 255 Returns the number of cameras under control of the visual component.

Code 4802h: Report Relative Object Position

For the Report Relative Object Position message, the parameters Range, Bearing, and Inclination indicate the relative offset to the object in the current platform coordinate system. All times are in Coordinated Universal Time (UTC).

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See Mapping following this table.
2	Time Stamp	Unsigned Integer	N/A	Bits 0-9: milliseconds, range 0...999 Bits 10-15: Seconds, range 0...59 Bits 16 – 21: Minutes, range 0...59 Bits 22-26: Hour (24 hour clock), range 0...23 Bits 27-31: Day, range 1…31
3	Range	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 10,000
4	Range Error	Unsigned Integer	Meters	Scaled Integer Lower Limit = 0 Upper Limit = 1,000
5	Bearing	Integer	Radians	Scaled Integer Lower Limit = -π Upper Limit = π
6	Bearing Error	Unsigned Integer	Radians	Scaled Integer Lower Limit = 0 Upper Limit = π
7	Inclination	Integer	Radians	Scaled Integer Lower Limit = -π Upper Limit = π
8	Inclination Error	Unsigned Integer	Radians	Scaled Integer Lower Limit = 0 Upper Limit = π
9	Confidence	Byte	N/A	0 – Low, 255 - Highest
10	Object ID	Unsigned Short Integer	N/A	Object Identifier

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	10	9	8	7	6	5	4	3

"R" indicates that the bit is reserved.

Code 4804h: Report Selected Camera

The Report Selected Camera message allows the Visual Component to report the selected camera currently in use for all imagery.

Field #	Name	Type	Units	Interpretation
1	Camera ID	Byte	N/A	1 – 255, 0 - Reserved

Code 4805h: Report Camera Capabilities

The Report Camera Capabilities message allows the Visual Component to report the capabilities for the selected camera.

Field #	Name	Type	Units	Interpretation (Note)
1	Presence Vector	Unsigned Short Integer	N/A	See mapping table that follows.
2	Camera ID	Byte	N/A	1 … 255 0 is Reserved
3	Description	String	N/A	50 characters containing the human readable description of the Camera. This shall be a null terminated ASCII String.
4	Max Horizontal FOV	Unsigned Short Integer	Radians	Scaled Integer Lower Limit = 0 Upper Limit = π
5	Min Horizontal FOV	Unsigned Short Integer	Radians	Scaled Integer Lower Limit = 0 Upper Limit = π
6	Max Vertical FOV	Unsigned Short Integer	Radians	Scaled Integer Lower Limit = 0 Upper Limit = π
7	Min Vertical FOV	Unsigned Short Integer	Radians	Scaled Integer Lower Limit = 0 Upper Limit = π
8	Max Horizontal Resolution	Unsigned Short Integer	N/A	The maximum number of possible horizontal scan lines produced by the camera
9	Min Horizontal Resolution	Unsigned Short Integer	N/A	The minimum number of possible horizontal scan lines produced by the camera
10	Max Vertical Resolution	Unsigned Short Integer	N/A	The maximum number of possible vertical scan lines produced by the camera
11	Min Vertical Resolution	Unsigned Short Integer	N/A	The minimum number of possible vertical scan lines produced by the camera

Field #	Name	Type	Units	Interpretation (Note)
12	Min Frame Rate	Unsigned Short Integer	Frames per Second	For video imagery this field indicates the minimum frame rate possible by the camera system in frames per second. Frame rates slower than 1 frame per second are not available.
13	Max Frame Rate	Unsigned Short Integer	Frames per Second	For video imagery this field indicates the maximum frame rate possible by the camera system in frames per second.
14	Min Shutter	Unsigned Short Integer	Seconds	For Still imagery this field indicates the slowest possible shutter speed as the inverse of the field value. For example 500 would indicate a $1/500^{th}$ of a second shutter speed. Speeds slower than one second are not available.
15	Max Shutter	Unsigned Short Integer	Seconds	For Still imagery this field indicates the fastest possible shutter speed as the inverse of the field value. For example 500 would indicate a $1/500^{th}$ of a second shutter speed.
16	Imagery Control	Unsigned Short Integer	N/A	Bit 0 – Auto Focus Available Bit 1 – Auto Exposure/Iris Available Bit 2 – Image Stabilization Available Bit 3 – White Balance Available Bit 4 – Sync Flash/Strobe Available Bit 5 – Red Eye Available Bit 6 – Auto Shutter Available Bit 7 – Auto Gain Available Bit 8 – Interlaced 0, non-Interlaced 1 Bits 9– 15 Reserved
17	Audio Control	Unsigned Short Integer	N/A	Bit 0 – Audio Available Bit 1 – Auto Gain Available Bit 2 – Stereo Available Bit 3 – Directional Available Bit 4 – Front Microphone Available Bit 5 – Rear Microphone Available Bit 6 – Left Microphone Available Bit 7 – Right Microphone Available Bits 8 – 15 Reserved

Vector to Data Field Mapping for Above Command								
Vector Bit	15	14	13	12	11	10	9	8
Data Field	R	17	16	15	14	13	12	11
Vector Bit	7	6	5	4	3	2	1	0
Data Field	10	9	8	7	6	5	4	3

"R" indicates that the bit is reserved.

Code 4806h: Report Camera Format Options

The Report Camera Format Options message allows the Visual Component to report the output formats available for the selected camera. Note that up to four image formats and two audio formats can be supported.

Field #	Name	Type	Units	Interpretation (Note)
1	Presence Vector	Byte	N/A	See mapping table that follows.
2	Camera ID	Byte	N/A	1 … 255 0 is Reserved
3	Audio Format	Byte	N/A	0-Unused 1-RAW 2-PCM 3-AU 4-WAV 5-MID 6-MPEG1 Layer 3 (MP3) 7-MP2 8-Advanced Streaming Format (ASF) 9-255 Reserved
4	Audio Format	Byte	N/A	See Field #3

5	Image Format	Byte	N/A	0 – Unused 1 – MPEG2 2 – MPEG4 3 – MJPEG 4 – NTSC 5 – PAL 6 – TIFF 7 – JPEG 8 – GIF 9 – H263 10 – H264 11 – PNG 12 – BMP 13 – RAW 14 – PPM 15 – PGM 16 – PNM 17 – 255 Reserved
6	Image Format	Byte	N/A	See Field #5
7	Image Format	Byte	N/A	See Field #5
8	Image Format	Byte	N/A	See Field #5
9	Format Option	Unsigned Integer	N/A	RESERVED: The value contained in this field shall not impact interoperability. Image format options beyond those published in this message may be necessary. One example use of this field is for the compression ratio for video.

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	9	8	7	6	5	4	3

"R" indicates that the bit is reserved.

Code 4807h: Report Image

The Report Image message contains the data for transmission of one entire frame of visual data. This can be one still image, a frame from a video stream, audio data, or the

combination of audio and video when supported by the format definition. The message size restrictions within JAUS may require that multiple transmissions be used to convey one frame. Paragraph 7.5 of the JAUS Reference Architecture addresses this issue.

Field #	Name	Type	Units	Interpretation
1..n	Video Data/Format	Unrestricted	N/A	The data and format contained within a Report Video Frame Message is not defined by the JAUS. The JTA and best commercial practices dictate the content of this message.

2.3.7 World Model Subgroup – Codes 4A00-4AFF

Code 4A20h: Report Vector Knowledge Store Object(s) Creation

The Code 4A20h: Report Vector Knowledge Store Object Creation message is used to confirm creation of objects in the vector knowledge store. This message is sent only when an object creation message is requested by setting bit zero in the Code 0A20h: Create Vector Knowledge Store Object message. If this bit is set, this message will be transmitted and the local request identifier (field 1) is set to the value sent with the Code 0A20h: Create Vector Knowledge Store Object message. If there was an error in creating an object specified in the Code 0A20h: Create Vector Knowledge Store Object message, the object identifier associated with that object shall be set to 0x00000000, a reserved value indicating an error condition.

Field #	Name	Type	Units	Interpretation
1	Local Request ID	Byte	N/A	Local request identifier sent by creating component
2	Number of Object IDs (p)	Unsigned Short Integer	N/A	
3	Object ID 1	Unsigned Integer		0x00000000 Invalid Object ID This value is used to inform the remote component that, for some reason, the corresponding object could not be created.
…	…	…	…	…

Field #	Name	Type	Units	Interpretation
2+p	Object ID p	Unsigned Integer		0x00000000 Invalid Object ID This value is used to inform the remote component that, for some reason, the corresponding object could not be created.

Code 4A21h: Report Vector Knowledge Store Feature Class Metadata

The Code 4A21h: Report Vector Knowledge Store Feature Class Metadata message allows access to feature class metadata stored within the vector knowledge store. It is transferred in response to the Code 2A21h: Query Vector Knowledge Store Feature Class Metadata message. If the query message requests metadata for all feature classes, a separate message shall be sent for each feature class. These metadata are entered remotely using the Code 0A21h: Set Vector Knowledge Store Feature Class Metadata message.

Field #	Name	Type	Units	Interpretation
1	Feature Class	Unsigned Short Integer	N/A	Enumeration 0 … 65,535 See Feature Class Table
2	Number of String Characters	Unsigned Short Integer	N/A	0 … 65,535
3	Feature Class Metadata	String	N/A	Variable length string

Code 4A22h: Report Vector Knowledge Store Bounds

The Code 4A23h: Report Vector Knowledge Store Bounds message format is shown below. This message reports the bounds as a response to the Query Vector Knowledge Store Bounds message. In this message, the knowledge store returns the two geographic points that represent the extents of the data within a feature class layer or all feature class layers.

Field #	Name	Type	Units	Interpretation
1	Local Request ID	Byte	N/A	Request identifier sent in query message
2	Feature Class	Unsigned Short	N/A	Enumeration 0 … 65,534 - See Feature

Field #	Name	Type	Units	Interpretation
		Integer		Class Table 65,535 – All
3	Southwest Bound Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
4	Southwest Bound Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
5	Northeast Bound Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
6	Northeast Bound Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180

Code 4A23h: Report Vector Knowledge Store Objects

The Code 4A22h: Report Vector Knowledge Store Objects message is sent in direct response to a Code 2A20h: Query Vector Knowledge Store Objects message. Field 1 is a presence vector used to specify whether data are included with this message. This functionality would be useful when there may be a large number of data points in response to a query. Rather than receiving the large volume of data, the component may instead only need to know how many objects match the criteria specified in the query. Note: Because an object is defined by a type, buffer, associated feature classes, and boundary points, all of these fields are required for each object.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	Bit Field Bit 0: Data are included after field 3. This is based on the presence vector received in the Code 2A20h: Query Vector Knowledge Store Objects

Field #	Name	Type	Units	Interpretation
				Message. If data are present after field 3, this bit should be set. Bits 1-7: Reserved
2	Local Request ID	Byte	N/A	Request identifier sent in query message
3	Number of Objects	Unsigned Short Integer	N/A	0 … 65,535 Number of Objects in Response to Query Message
4	Object 1 ID	Unsigned Integer	N/A	0x00000000 – Reserved
5	Object 1 Type	Byte	N/A	Enumeration 0: Point 1: Line 2: Polygon 3 – 255: Reserved
6	Object 1 Buffer	Float	Meters	
7	Object 1 Number of Feature Classes	Unsigned Short Integer	N/A	
8	Object 1 Feature Class 1	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – Reserved
9	Object 1 Feature Class 1 Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
10	Object 1 Feature Class Attribute 1	Varies (see field 4)	Varies with Feature Class	
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

Field #	Name	Type	Units	Interpretation
	Object 1 Feature Class m	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – Reserved
	Object 1 Feature Class m Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
	Object 1 Feature Class Attribute m	Varies (see previous field)	Varies with Feature Class	
	Number of Points for Object 1	Unsigned Short Integer	N/A	
	Object 1 Point 1 Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
	Object 1 Point 1 Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
	…	…	…	…
	…	…	…	…
	Object 1 Point n Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
	Object 1 Point n Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
	Object p ID	Unsigned Integer	N/A	0x00000000 – Reserved
	Object p Type	Byte	N/A	Enumeration 0: Point 1: Line 2: Polygon 3 – 255: Reserved

Field #	Name	Type	Units	Interpretation
	Object p Buffer	Float	Meters	
	Object p Number of Feature Classes	Byte	N/A	
	Object p Feature Class 1	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – Reserved
	Object p Feature Class 1 Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved
	Object p Feature Class Attribute 1	Varies (see previous field)	Varies with Feature Class	
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
	Object p Feature Class m	Unsigned Short Integer	N/A	Enumeration 0 … 65,534 - See Feature Class Table 65,535 – Reserved
	Object p Feature Class m Attribute Data Type	Byte	N/A	Enumeration 0: Byte 1: Short Integer 2: Integer 3: Long Integer 4: Unsigned Short Integer 5: Unsigned Integer 6: Unsigned Long Integer 7: Float 8: Long Float 9: RGB (3 Bytes) 10 – 255: Reserved

Field #	Name	Type	Units	Interpretation
	Object p Feature Class Attribute m	Varies (see previous field)	Varies with Feature Class	
	Number of Points for Object p	Unsigned Short Integer	N/A	
	Object p Point 1 Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
	Object p Point 1 Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
	…	…	…	…
	…	…	…	…
	Object p Point r Latitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
	Object p Point r Longitude (WGS84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180

Code 4A24h: Report Vector Knowledge Store Data Transfer Termination

The Code 4A24h: Report Vector Knowledge Store Data Transfer Termination message notifies other JAUS components that data that were being transferred or were going to be transferred to them has been stopped. This message is sent in response to the Code 0A25h: Terminate Vector Knowledge Store Data Transfer message. It is also sent whenever data transfer is interrupted due to a change in the component state.

2.3.8 Dynamic Configuration Subgroup – Codes 4B00-4BFF

Code 4B00: Report Identification

This message shall provide the requesting component an identification summary of the Subsystem, Node, or Component.

Field #	Name	Type	Units	Interpretation
1	Query Type	Byte	N/A	Used to mark the type of identification being returned. Should correspond to its respective Query. 0: Reserved 1: System Identification 2: SS Identification 3: Node Identification 4: Component Identification 5 – 255: Reserved
2	Authority	Byte	N/A	Lowest level of authority required to gain control of Subsystem, Node, or Component Capabilities
3	Type	Unsigned Short	N/A	This field identifies the particular Unmanned Vehicle Type, Node Type or Component Type. The following values apply: 00000 = Reserved 10001 = Vehicle TBD 10002 = Vehicle TBD … 20000 = Reserved 20001 = OCU TBD 20002 = OCU TBD … 30000 = Reserved 30001 = Other Subsystem TBD 30002 = Other Subsystem TBD … 40000 = Reserved 40001 = Node TBD 40002 = Node TBD … 50000 = Reserved 50001 = Payload TBD 50002 = Payload TBD … 60000– 65535 Reserved
4	Identification	Byte	N/A	Human-recognizable name of the Subsystem, Node or Component. This shall be a null terminated ASCII string.

Code 4B01: Report Configuration

This message shall provide the receiving component a table of all existing components located on the source's subsystem or node depending on the value of field 1 of the Query Configuration message.

Field #	Name	Type	Units	Interpretation
1	Node Count	Byte	N/A	# of Nodes reported. For a single Node Report this field shall be 1.
2	NodeID1	Byte	N/A	Node ID of first node in table. For single Node or Component reporting this field shall contain the Node ID as specified in the Destination Address of the Query Configuration message
3	Comp Count 1	Byte	N/A	# of Comps reported for NodeID1
4	CompID1	Byte	N/A	Comp ID of first Component reported. For Single Component reporting this field shall contain the Component ID as specified in the Destination Address of the Query Configuration message and shall be the only Component reported
5	InstID1	Byte	N/A	Inst ID of first Component reported.
	…			
	CompIDi	Byte	N/A	Comp ID of ith comp on first node in table
	InstIDi	Byte	N/A	Inst ID of ith comp on first node in table
	NodeID2	Byte	N/A	Node ID of second node in table
	Comp Count 2	Byte	N/A	# of Comps on second node in table
	CompID1	Byte	N/A	Comp ID of first comp on second node in table
	InstID1	Byte	N/A	Inst ID of first comp on second node in table
	…			
	CompIDj	Byte	N/A	Comp ID of jth comp on second node in table
	InstIDj	Byte	N/A	Inst ID of jth comp on second node in table
	…			
	NodeIDn	Byte	N/A	Node ID of the nth node in table
	Comp Count n	Byte	N/A	# of Comp on nth node in table
	CompID1	Byte	N/A	Comp ID of first comp on nth node in table
	InstID1	Byte	N/A	Inst ID of first comp on nth node in table
	…			
	CompIDk	Byte	N/A	Comp ID of kth comp on nth node in table
	InstIDk	Byte	N/A	Inst ID of kth comp on nth node in table

Code 4B02: Report Subsystem List

This message shall provide the receiving component a table of all subsystems located in the source's system. It also provides the ID of the component to send a Query Configuration message within the subsystem.

Field #	Name	Type	Units	Interpretation
1	Subsystem Count(n)	Byte	N/A	# of subsystems with the system.
4(n-1)+2	SubsysID(i)	Byte	N/A	Subsystem ID.
4(n-1)+3	NodeID(i)	Byte	N/A	Node ID used for Query Configuration.
4(n-1)+4	CompID(i)	Byte	N/A	Component ID used for Query Configuration
4(n-1)+5	InstID(i)	Byte	N/A	Instance ID used for Query Configuration.

Code 4B03: Report Services

This message allows a component to publish its capabilities, according to the Service Dictionary presented below. If a component ID is specified in the RA, it may report only one service in beyond the core message support, and this service must be equal to the component ID. If a component ID is not listed in the RA, it may report any number of services. For example, a component with ID 33 must provide only service 33. The exception to this rule is component ID 1 (the Node Manager) which may provide any number of services in addition to core message support.

Field #	Name	Type	Units	Interpretation
1	Service Count	Byte	N/A	# of services supported by sending component
2	Service Type (1)	Unsigned Short	N/A	Service type (See Service Type Table)
3	Input Message Count	Byte	N/A	# of input messages used to support service 1
4	Input Message 1	Unsigned Short	N/A	Message code for supported input message 1 in service 1

Field #	Name	Type	Units	Interpretation
5	Input Message 1 Presence Vector	Unsigned Integer	N/A	Presence vector for message code in previous field. This field, and all subsequent Presence Vector fields, shall always be 32 bits. For presence vectors smaller than 32 bits, the representative data shall be inserted with matching bit significance.
…				
	Input Message i	Unsigned Short	N/A	Message code for supported input message j in service 1
	Input Message i Presence Vector	Unsigned Integer	N/A	See interpretation for field 5
…				
	Output Message Count	Byte	N/A	# of output messages used to support service 1
	Output Message 1	Unsigned Short	N/A	Message code for supported input message in service 1
	Output Message 1 Presence Vector	Unsigned Integer	N/A	See interpretation for field 5
…				
	Output Message j	Unsigned Short	N/A	Message code for supported input message in service 1
	Output Message j Presence Vector	Unsigned Integer	N/A	See interpretation for field 5
…				
	Service n	Unsigned Short	N/A	Service Type (see Service Type Table
	Input Message Count	Byte	N/A	# of input messages used to support service 1
	Input Message 1	Unsigned Short	N/A	Message code for supported input message 1 in service n

Field #	Name	Type	Units	Interpretation
	Input Message 1 Presence Vector	Unsigned Integer	N/A	Presence vector for message code in previous field. This field, and all subsequent Presence Vector fields, shall always be 32 bits. For presence vectors smaller than 32 bits, the representative data shall be inserted with matching bit significance.
…				
	Input Message k	Unsigned Short	N/A	Message code for supported input message in service n
	Input Message k Presence Vector	Unsigned Integer	N/A	See interpretation for field 5
…				
	Output Message Count	Byte	N/A	# of output messages used to support service n
	Output Message 1	Unsigned Short	N/A	Message code for supported input message in service n
	Output Message 1 Presence Vector	Unsigned Integer	N/A	See interpretation for field 5
…				
	Output Message p	Unsigned Short	N/A	Message code for supported input message in service n
	Output Message p Presence Vector	Unsigned Integer	N/A	See interpretation for field 5

Table 2-1 Service Type Dictionary

Field #	Name
0	Core Message Support
32	Subsystem Commander
33	Primitive Driver
34	Global Vector Driver
35	Communicator
37	Visual Sensor
38	Global Pose Sensor

40	System Commander
41	Local Pose Sensor
42	Velocity State Sensor
43	Reflexive Driver
44	Local Vector Driver
45	Global Waypoint Driver
46	Local Waypoint Driver
47	Global Path Segment Driver
48	Local Path Segment Driver
49	Primitive Manipulator
50	Range Sensor
51	Manipulator Joint Position Sensor
52	Manipulator Joint Velocity Sensor
53	Manipulator Joint Force/Torque Sensor
54	Manipulator Joint Positions Driver
55	Manipulator End-Effector Pose Driver
56	Manipulator Joint Velocities Driver
57	Manipulator End-Effector Velocity State Driver
58	Manipulator Joint Move Driver
59	Manipulator End-Effector Discrete Pose Driver

2.3.9 Payload Subgroup – Codes 4D00-4DFF

Code 4D00h: Report Payload Interface Message

The Report Payload Interface Message is used to publish the command and informational data elements, their respective types, units and ranges to the using component(s). The total number of interface elements is determined by adding the first two fields. All payloads must have at least one Informational interface.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Byte	N/A	See mapping table below
2	Number Command Interfaces (N)	Byte	Integer	The number (N) of unique command and control data elements

Field #	Name	Type	Units	Interpretation
				supported by the Payload.
3	Number Information Interfaces (M)	Byte	Integer	The number (M) of unique information data elements supported by the Payload.
4	Identifier Command 1	String	N/A	NULL-Terminated Identification of the Command Interface
5	Type Code Command 1	Byte	N/A	See Type Code Table in Appendix A
6	Units Command 1	Byte	N/A	See Units Table in Appendix A
7	Blocking Flag	Byte	N/A	Bit Field 0: Blocking (Active High) 1-7: Reserved
8	Minimum Command 1	See Note *	Defined by Units	Minimum Acceptable Value
9	Default Command 1	See Note *	Defined by Units	Default Value
10	Maximum Command 1	See Note *	Defined by Units	Maximum Acceptable Value
11	Enumerations Length	Unsigned Short	Bytes	Number of Bytes in the Enumerations Command field. This number includes the terminating Null character, the comma delimiters and the spaces.
12	Enumerations Command 1	String	N/A	See "Enumeration Discussion" Below.
13	HMI Recommendation	Byte	N/A	See HMI Enumeration Table in Appendix A
14	HMI Recommended Position X	Unsigned Short	Pixels	
15	HMI Recommended Position Y	Unsigned Short	Pixels	
16	HMI Recommended Width	Unsigned Short	Pixels	
17	HMI Recommended Height	Unsigned Short	Pixels	

Field #	Name	Type	Units	Interpretation
. . .	Repeat fields 4 through 17			
14N+4	Identifier Info 1	String	N/A	NULL-Terminated Identification of the Information Interface
14N+5	Command Interface Association	Byte	N/A	Number of the Command Interface that this Information Interface is associated with. If there is no association, this value shall be set to zero.
14N + 6	Type Code Info 1	Byte	N/A	See Type Code Table in Appendix A
14N + 7	Units Info 1	Byte	N/A	See Units Table in Appendix A
14N + 8	Minimum Info 1	See Note *	Defined by Units	Minimum Acceptable Value
14N + 9	Default Info 1	See Note *	Defined by Units	Default Value
14N + 10	Maximum Info 1	See Note *	Defined by Units	Maximum Acceptable Value
14N + 11	Enumerations Length	Unsigned Short	Bytes	Number of Bytes in the Enumerations Command field. This number includes the terminating Null character, the comma delimiters and the spaces.
14N + 12	Enumerations Info 1	String	N/A	See "Enumeration Discussion" Below.
14N + 13	HMI Recommendation	Byte	N/A	See HMI Enumeration Table in Appendix A
14N+14	HMI Recommended Position X	Unsigned Short	Pixels	
14N+15	HMI Recommended Position Y	Unsigned Short	Pixels	
14N+16	HMI Recommended Width	Unsigned Short	Pixels	

Field #	Name	Type	Units	Interpretation
14N+17	HMI Recommended Height	Unsigned Short	Pixels	
. . .	Repeat fields 14N+4 through 14N+17			

Vector to Data Field Mapping for Above Command								
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	R	17	16	15	14	13

"R" indicates that the bit is reserved.

Note *: The preceding Type Code field defines the Type with the exception of Scaled types when the type of this field is *float*. The values of the Minimum, and Maximum fields specify the range of the Scaled type. The type of the Default value is also a *float in this message*. ***Additionally, when the preceding Type Code field is 19, a length and string type, the Minimum, Default, and Maximum fields shall be assigned type byte.***

Enumerations allow sensor data formats that categorize their input, or detect states, or have some other discrete nature. The implementation is an Unsigned Short Integer, which is used as an index into the Text Array defined in the Enumerations field.

Booleans are a specialization of Enumerations that only have two values – True and False. The implementation is a Byte with 0 = False, 1 = True, 2-255 = Reserved.

Enumeration Discussion: The Enumerations value is a comma delimited, NULL terminated string of enumeration descriptions. The Enumerations Length field shall be 0 if there are no enumerations for that particular interface. When the Enumerations Length is 0, the Enumerations field does not exist. The enumerations are string representations of sequential numeric values from 1 up to 255. The intent of this type of data is to provide an obvious mapping between the numeric values and the human readable value.

When a payload receives a query for a data element (Query Payload Data Element) that maps to an enumeration, the response (Report Payload Data Element) value field will

contain a numeric value. The user (component that initially performed the query) will use the value as in index into the list of string representations.

Code 4D01h: Report Payload Data Element

The Report Payload Data Element message is used as a response to the Query Payload Data Element message. **NOTE**: This message cannot presently be supported via Service Connection as the Create Service Connection message will not allow the required parameter specified within the Query Payload Data Element message. The Information Interface Number is the unique identifier derived from the placement of the interface in the Report Payload Interface message. The Information Interface Number starts at 1 for the first published informational interface.

Field #	Name	Type	Units	Interpretation
1	Number Information Interfaces	Byte	Integer	The number (M) of information data elements being reported by the Payload.
2	Information Interface Number	Byte	Integer	The number of the information data element as reported in the Report Payload Interface.
3	Value	Defined by Type Code	Defined by Units	Current/Reported Value
…				
2M	Information Interface Number	Byte	Integer	The number of the information data element as reported in the Report Payload Interface.
2M+1	Value	Defined by Type Code	Defined by Units	Current/Reported Value

2.3.10 Planning Subgroup – Codes 4E00 – 4EFF

Code 4E00h: Report Spooling Preference

A Report Spooling Preference message is sent in response to a Query Spooling Preference message. This message indicates how components want commands to be

spooled to them. A discrete number of commands, a desired distance worth of commands, or a specified time worth of commands can be spooled. If the Spool Type is set to Count, then the data field indicates how many commands to send in each Spool Mission message. If the Spool Type is set to Distance, then the data field indicates how many meters worth of commands to send in each Spool Mission message. If the commands in the mission are Set Path Segment messages, the distance of each path segment is summed up until the distance is obtained and the corresponding Set Path Segment commands are spooled. If the Spool Type is set to Time, then the data field indicates how many seconds worth of commands to send in each Spool Mission message.

Field #	Name	Type	Units	Interpretation
1	Spool Type	Byte	N/A	0 = Count 1 = Distance 2 = Time 3 – 255 Reserved
2	Data	Integer	N/A if Spool Type = 0 Meters if Spool Type = 1 Seconds if Spool Type = 2	If spool type is 0, this value shall indicate how many messages a component prefers. If spool type is 1, this value shall indicate the distance in meters a component prefers. Scaled Integer Lower Limit = 0 Upper Limit = 35,000 If spool type is 2, this value shall indicate the time in seconds a component prefers. Scaled Integer Lower Limit = 0 Upper Limit = 10,000

Code 4E01h: Report Mission Status

The Report Mission Status message is sent in response to the Query Mission Status message to inform the recipient of the status of a Mission, Task, or Message within a Mission.

The Type field indicates whether the status refers to a Mission, Task, or Message. The following fields indicate the specific Mission, Task, or Message the status refers to.

Field #	Name	Type	Units	Interpretation
1	Type	Byte	N/A	0 = Mission 1 = Task 2 = Message 3 – 255 Reserved
2	Status	Byte	N/A	0 = Spooling (in progress) 1 = Pending 2 = Paused 3 = Aborted 4 = Finished 5 – 255 Reserved
3	Secondary Status	Byte	N/A	0 = Non-Error condition 1 = Lost component control 2 = Tolerance not met 3 – 255 Reserved
4	Mission ID	Unsigned Short Integer	N/A	Unique Mission ID
5	Task ID	Unsigned Short Integer	N/A	Unique Task ID (If Type = 0, Task ID = 0)
6	UID	Unsigned Short Integer	N/A	Unique Message ID (If Type = 0 or 1, Message ID = 0)

2.4 Event Setup Class (DEPRICATED IN V4.0)

Event setup class messages are used to arm asynchronous events. Currently no Event Setup Class messages are defined.

2.5 Event Notification Class (DEPRICATED IN V4.0)

Event notification class messages are used to report occurrence of an armed event. Currently no Event Notification Class messages are defined.

2.6 Node Management Class

Currently no Node Management Class messages are defined.

Appendix A - Payload Message Enumerations

Table A-1 – Payload Type Definitions

Enumeration	Type Definition
0	Reserved
1	Short Integer (2 bytes)
2	Integer (4 bytes)
3	Long Integer (8 bytes)
4	Byte (1 byte)
5	Unsigned Short (2 bytes)
6	Unsigned Integer (4 bytes)
7	Unsigned Long (8 bytes)
8	Float (4 bytes)
9	Long Float (8 bytes)
10	Scaled Unsigned Byte (1 byte)
11	Scaled Signed Short Integer (2 bytes)
12	Scaled Unsigned Short Integer (2 bytes)
13	Scaled Signed Integer (4 bytes)
14	Scaled Unsigned Integer (4 bytes)
15	Scaled Signed Long Integer (8 bytes) (not used in experiment #2)
16	Scaled Unsigned Long Integer (8 bytes) (not used in experiment #2)
17	Enumeration (2 bytes)
18	Boolean (1 byte)
19	Length (unsigned short) followed by Null Terminated ASCII String
20	Unsigned Byte 2-Tuple (2 Bytes)
21	Unsigned Short 2-Tuple (4 Bytes)
22	Unsigned Integer 2-Tuple (8 Bytes)
23-255	Reserved

The HMI Recommendation Field in the Report Payload Interface message serves only as a recommendation for the display output associated with that particular data or control element.

Table A-2 HMI Enumerations

Enum	HMI Widget Type	Description
0	Reserved	Reserved
1	Alpha-list	Allows a user to select from a list of words, with the ability to narrow the search list by typing in a few characters of the desired word.
2	Button	This creates a button widget
3	Button-box	This creates a multiple button widget
4	Calendar	Creates a little simple calendar widget
5	Canvas	Canvas widgets implement structured graphics
6	Dialog	Prompts the user with a message, and the user can pick an answer from the buttons provided
7	Gauge	
8	Graph	A graphing widget
9	Histogram	Creates a histogram. (vertical or horizontal)
10	Item List	Creates a pop up field that allows the user to select one of several choices in a small field. Very useful for things like days of the week or month names.
11	Label	Displays messages in a pop up box, or the label can be considered part of the screen.
12	Marquee	Displays a message in a scrolling marquee
13	Matrix	Creates a complex matrix with lots of options.
14	Menu	Creates a pull-down menu interface
15	Multiple Line Entry	A multiple line entry field. Very useful for long fields. (like a description field)
16	Point 2-D	Two-dimensional point coordinates are displayed with a text entry for each of their component values.
17	Point 3-D	Three-dimensional point coordinates are displayed with a text entry for each of their component values.
18	Push Button	
19	Radio List	Creates a radio button list.
20	Scale	Creates a numeric scale. Used for allowing a user to pick a numeric value and restrict them to a range of values
21	Scrolling List	Creates a scrolling list/menu list
22	Scrolling Window	Creates a scrolling log file viewer. Can add information into the window while it's running. A good widget for displaying the progress of something. (akin to a console window)
23	Selection List	Creates a multiple option selection list

Enum	HMI Widget Type	Description
24	Slider	Akin to the scale widget, this widget provides a visual slide bar to represent the numeric value
25	Spin Box	Numeric variables are usually displayed with a simple text entry widget.
26	Toggle Button	Toggle buttons are used to represent Boolean variables, which can have the value of either "true" or "false"
27	Viewer	This is a file/information viewer. Very useful when you need to display loads of information.
28	Joystick	
29	Image	This HMI interface allows an entire GUI or portions of a GUI to be sent to the OCU as an image. Clicks on the image are reported to the payload as points with coordinates of X and Y. These coordinates shall be reported using one of the 2-tuple data types defined in the TYPE CODE TABLE.
30	TBD	
31 – 255	Reserved	

Table A-3 – Payload Unit Types

Enum	Quantity	Name	Symbol
0	Unit less/dimensionless	Place holder	N/A
1	Length	Meter	m
2	Mass	Kilogram	kg
3	Time	Second	s
4	Electric current	Ampere	A
5	Thermodynamic temperature	Kelvin	K
6	Amount of substance	Mole	mol
7	Luminous Intensity	Candela	cd
8	Area	Square meter	m2
9	Volume	Cubic meter	m3
10	Speed, Velocity	Meter per second	m/s
11	Acceleration	Meter per second squared	m/s2
12	Wave Number	Reciprocal meter	m-1
13	Mass Density	Kilogram per cubic meter	kg/m3
14	Specific Volume	Cubic meter per kilogram	m3/kg
15	Current density	Ampere per square meter	A/m2
16	Magnetic field strength	Ampere per meter	A/m
17	Amount-of-substance Concentration	Mole per cubic meter	mol/m3
18	Luminance	Candela per square meter	cd/m2

Enum	Quantity	Name	Symbol
19	Plane Angle	Radian	rad
20	Solid Angle	Steradian	sr
21	Frequency	Hertz	Hz
22	Force	Newton	N
23	Pressure, stress	Pascal	Pa
24	Energy, work, quantity of heat	Joule	J
25	Power, radiant flux	Watt	W
26	Electric charge, quantity of electricity	Coulomb	C
27	Electric potential difference, electromotive force	Volt	V
28	Capacitance	Farad	F
29	Electric resistance	Ohm	W
30	Electric conductance	Siemens	S
31	Magnetic Flux	Weber	Wb
32	Magnetic Flux Density	Tesla	T
33	Inductance	Henry	H
34	Celsius Temperature	Degree Celsius	C
35	Luminous Flux	Lumen	lm
36	Illuminance	Lux	lx
37	Activity (of a radionuclide)	Becquerel	Bq
38	Absorbed dose, Specific Energy (imparted), Kerma	Gray	Gy
39	Dose equivalent (d)	Sievert	Sv
40	Catalytic activity	Katal	kat
41	Dynamic viscosity	Pascal second	Pa.s
42	Moment of force	Newton meter	N.m
43	Surface tension	Newton per meter	N/m
44	Angular velocity	Radian per second	rad/s
45	Angular acceleration	Radian per second squared	rad/s2
46	Heat flux density, irradiance	Watt per square meter	W/m2
47	Heat capacity, entropy	Joule per Kelvin	J/K
48	Specific heat capacity, specific entropy	Joule per kilogram Kelvin	J/(kg.K)
49	Specific energy	Joule per kilogram	J/kg
50	Thermal conductivity	Watt per meter Kelvin	W/(m.K)
51	Energy density	Joule per cubic meter	J/m3
52	Electric field strength	Volt per meter	V/m
53	Electric charge density	Coulomb per cubic meter	C/m3
54	Electric flux density	Coulomb per square meter	C/m2
55	Permittivity	Farad per meter	F/m
56	Permeability	Henry per meter	H/m
57	Molar energy	Joule per mole	J/mol

Enum	Quantity	Name	Symbol
58	Molar entropy, molar heat capacity	Joule per mole Kelvin	J/(mol.K)
59	Exposure (x-rays)	Coulomb per kilogram	C/kg
60	Absorbed Dose Rate	Gray per second	Gy/s
61	Radiant Intensity	Watt per Steradian	W/sr
62	Radiance	Watt per square meter Steradian	W/(m2.sr)
63	Catalytic (activity) Concentration	Katal per cubic meter	kat/m3
127	Percent		
128	Decimal Degrees		
129			
130	Pixels		