



AEROSPACE STANDARD

AS8024™

Issued 2019-06
Reaffirmed 2025-04

J AUS Autonomous Capabilities Service Set

RATIONALE

This document, the JAUS Autonomous Capabilities Service Set (AS8024), defines a message-passing interface for services commonly found in unmanned vehicle systems. These services represent the platform-independent capabilities common across multiple vehicle types, including air, sea, and ground. Inherited capabilities are specified in the JAUS Core Service Set (AS5710) and are frequently referenced herein.

AS8024 has been reaffirmed to comply with the SAE Five-Year Review policy.

INTRODUCTION

The primary goal of the JAUS Autonomous Capabilities Service Set is logical interoperability between communicating elements in an unmanned system. To this end, each service defines the messages (vocabulary) and protocol (rules) for data exchange. This logical interoperability is independent of the physical transport, and it is expected that a Transport Standard, such as the JAUS Transport Specification [AS5669], is used in conjunction with this specification.

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

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

The tables and diagrams in this document are hand-transcribed from the JSIDL XML specification in the Appendixes. In case of transcription errors, the XML specification should be considered correct.

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

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

Copyright © 2025 SAE International

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

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

SAE WEB ADDRESS:

Full document details: SAE AS8024

For more information on this standard, visit
<https://www.sae.org/standards/content/AS8024/>

Documents

Standard, visit Sharing Hub

TABLE OF CONTENTS

1.	SCOPE.....	4
1.1	Purpose.....	4
1.2	J AUS Core Service Set.....	4
1.3	Compliance.....	4
1.4	Document Organization.....	4
2.	REFERENCES.....	4
2.1	SAE Publications.....	5
3.	COMMON CONVENTIONS.....	5
3.1	Safe Motion on Exit from Ready State.....	5
4.	SERVICE DEFINITIONS.....	5
4.1	Comms Lost Policy Manager.....	5
4.1.1	Description.....	5
4.1.2	Assumptions.....	5
4.1.3	References.....	5
4.1.4	Vocabulary.....	6
4.1.5	Encoding.....	6
4.1.6	Protocol Behavior.....	19
4.2	Cost Map 2D.....	20
4.2.1	Description.....	20
4.2.2	Assumptions.....	21
4.2.3	References.....	21
4.2.4	Vocabulary.....	21
4.2.5	Encoding.....	21
4.2.6	Protocol Behavior.....	25
4.3	Path Reporter.....	26
4.3.1	Description.....	26
4.3.2	Assumptions.....	26
4.3.3	References.....	26
4.3.4	Vocabulary.....	26
4.3.5	Encoding.....	27
4.3.6	Protocol Behavior.....	31
4.4	Retrotraverse.....	32
4.4.1	Description.....	32
4.4.2	Assumptions.....	32
4.4.3	References.....	32
4.4.4	Vocabulary.....	32
4.4.5	Encoding.....	32
4.4.6	Protocol Behavior.....	40
4.5	Self Righting.....	42
4.5.1	Description.....	42
4.5.2	Assumptions.....	42
4.5.3	References.....	42
4.5.4	Vocabulary.....	42
4.5.5	Encoding.....	42
4.5.6	Protocol Behavior.....	44
5.	NOTES.....	45
5.1	Revision Indicator.....	45
APPENDIX A	XML FOR SERVICE DEFINITIONS.....	46

Figure 1	CommsLostPolicyManager service protocol behavior	19
Figure 2	CostMap2D service protocol behavior	25
Figure 3	PathReporter service protocol behavior	31
Figure 4	Retrotraverse service protocol behavior	40
Figure 5	SelfRighting service protocol behavior	44
Table 1	CommsLostPolicyManager service message vocabulary	6
Table 2	CommsLostPolicyManager service internal events set	6
Table 3	CommsLost event encoding	6
Table 4	CommsRestored event encoding	7
Table 5	SetCommsLostPolicy message encoding	7
Table 6	QueryCommsLostCapabilities message encoding	11
Table 7	QueryCommsLostConfiguration message encoding	12
Table 8	QueryCommsLostStatus message encoding	12
Table 9	SetCommsLostPolicyResponse message encoding	12
Table 10	ReportCommsLostCapabilities message encoding	13
Table 11	ReportCommsLostConfiguration message encoding	14
Table 12	ReportCommsLostStatus message encoding	18
Table 13	CommsLostPolicyManager service state transitions	19
Table 14	CommsLostPolicyManager service conditions	20
Table 15	CommsLostPolicyManager service transition actions	20
Table 16	CostMap2D service message vocabulary	21
Table 17	QueryCostMap2D message encoding	22
Table 18	ReportCostMap2D message encoding	23
Table 19	CostMap2D service state transitions	25
Table 20	CostMap2D service transition actions	25
Table 21	PathReporter service message vocabulary	26
Table 22	QueryPathReporterCapabilities message encoding	27
Table 23	QueryPath message encoding	27
Table 24	ReportPathReporterCapabilities message encoding	28
Table 25	ReportPath message encoding	29
Table 26	PathReporter service state transitions	31
Table 27	PathReporter service conditions	31
Table 28	PathReporter service transition actions	31
Table 29	Retrotraverse service message vocabulary	32
Table 30	QueryRetrotraverseCapabilities message encoding	32
Table 31	QueryRetrotraverseStatus message encoding	33
Table 32	StartRetrotraverse message encoding	33
Table 33	CancelRetrotraverse message encoding	36
Table 34	ReportRetrotraverseCapabilities message encoding	36
Table 35	ReportRetrotraverseStatus message encoding	37
Table 36	StartRetrotraverseResponse message encoding	40
Table 37	Retrotraverse service state entry/exit actions	40
Table 38	Retrotraverse service state transitions	41
Table 39	Retrotraverse service conditions	41
Table 40	Retrotraverse service transition actions	41
Table 41	SelfRighting service message vocabulary	42
Table 42	SetSelfRightingState message encoding	42
Table 43	QuerySelfRightingState message encoding	43
Table 44	ReportSelfRightingState message encoding	43
Table 45	SelfRighting service state entry/exit actions	44
Table 46	SelfRighting service state transitions	44
Table 47	SelfRighting service conditions	44
Table 48	SelfRighting service transition actions	44

1. SCOPE

This document defines a set of standard application layer interfaces called *JAUS Autonomous Capabilities Services*. JAUS Services provide the means for software entities in an unmanned system or system of unmanned systems to communicate and coordinate their activities. The Autonomous Behaviors Services represent the platform-independent capabilities commonly found in platforms across domains, including air, maritime, and ground. At present five (5) services are defined in this document. These services are:

- Comms Lost Policy Manager: Detect and recover from loss of communications with a control station
- Retrotraverse: Return along a path previously traveled
- Self-Righting: Attempt to recover from a tip over condition
- Cost Map 2D: Provides information about the current operating environment of the platform
- Path Reporter: Provides information about the previous or future planned path of the platform

1.1 Purpose

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

1.2 JAUS Core Service Set

The JAUS Service Definitions defined herein make use of the inheritance functionality provided by JSIDL to incorporate capabilities as specified by the JAUS Core Service Set (AS5710). These documents must be used together to define a complete service. It is important to note that details related to 'Message Serialization' and 'Understanding Protocol Descriptions' can be found in the JAUS Core Service Set document, and are not repeated here.

1.3 Compliance

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

1.4 Document Organization

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

2. REFERENCES

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

2.1 SAE Publications

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

AS5669	J AUS / SDP Transport Specification
AS5684	J AUS Service Interface Definition Language
AS5710	J AUS Core Service Set
AS6009	J AUS Mobility Service Set
AS6062	J AUS Mission Spooling Service Set

3. COMMON CONVENTIONS

3.1 Safe Motion on Exit from Ready State

All of the driver services contained herein inherit from the Management Service published in AS5710 and define an exit action when leaving the Controlled::Ready state. As commanded motion of the platform is only allowed in the Ready state, this transition represents an exception from a normal operating state and requires the platform to transition to safe motion. However, the actual behavior of the safe motion condition is not defined within the document, as it may be highly dependent on the domain, the vehicle capabilities, and the mission. For example, the behavior for a small ground vehicle may simply be to stop motion, while a larger ground vehicle may require both stopping and applying the parking brake. On the other hand, a surface or air vehicle capable of hovering might actively apply thruster or engines to actively maintain a fixed global position while combating effects of current and wind. Fixed wing aircraft might loiter in a circle around a global position where the transition occurred. These examples are notional only and intended for clarity; the exact behavior of any particular implementation is not defined by the Standard.

4. SERVICE DEFINITIONS

4.1 Comms Lost Policy Manager

name=CommsLostPolicyManager

version=1.1

id=urn:jaus:jss:behaviors:CommsLostPolicyManager

4.1.1 Description

The Comms Lost Policy Manager Service provides a mechanism for setting the behavior of a platform when communication is lost with the current controller. Note that no default policy is specified by the Standard and is left to the implementation.

4.1.2 Assumptions

Messages may be delayed, lost, or reordered.

4.1.3 References

Inherits-from=AccessControl

version=1.1

id=urn:jaus:jss:core:AccessControl

4.1.4 Vocabulary

Table 1 - CommsLostPolicyManager service message vocabulary

Message ID (hex)	Name	Command
Input Set		
0C01	SetCommsLostPolicy	true
2C02	QueryCommsLostCapabilities	false
2C03	QueryCommsLostConfiguration	false
2C04	QueryCommsLostStatus	false
Output Set		
4C01	SetCommsLostPolicyResponse	false
4C02	ReportCommsLostCapabilities	false
4C03	ReportCommsLostConfiguration	false
4C04	ReportCommsLostStatus	false

Table 2 - CommsLostPolicyManager service internal events set

Name	Interpretation
CommsLost	An internally generated event when communication is lost with the controlling client. The exact criteria for determining a comms-lost event is implementation dependent and is therefore left unspecified.
CommsRestored	An internally generated event when communication is restored with the previous controlling client. The exact criteria for determining a communications restored event is implementation dependent and is therefore left unspecified.

4.1.5 Encoding

4.1.5.1 Internal Event Set

4.1.5.1.1 CommsLost

An internally generated event when communication is lost with the controlling client. The exact criteria for determining a comms-lost event is implementation dependent and is therefore left unspecified.

Table 3 - CommsLost event encoding

body └ (empty)
Empty message body

4.1.5.1.2 CommsRestored

An internally generated event when communication is restored with the previous controlling client. The exact criteria for determining a communications restored event is implementation dependent and is therefore left unspecified.

Table 4 - CommsRestored event encoding

body └ (empty)
Empty message body

4.1.5.2 Input Set

4.1.5.2.1 ID 0C01: SetCommsLostPolicy

This message is used to set the active behavior for a comms-lost event for the platform. A comms-lost event is defined as the loss of communications with the controlling client for more than CommsLostTimeout seconds. Behaviors supported are: stop, continue on mission, move to a given position at a specified speed, perform retrotraverse, or run a specified mission spool. In the case of move to a given position, the overall maximum distance to move in an effort to restore communications can be further restricted.

Table 5 - SetCommsLostPolicy message encoding

<pre> body └ sequence name = PolicySeq └ record name = PolicyRec └ sequence name = CommsLostSeq └ record name = CommsLostDetectionRec └ variant name = CommsLostPolicyVar (vtag_field = unsigned byte) └ record name = StopMissionVar └ record name = ContinueMissionVar └ sequence name = MoveToGlobalPosSeq └ record name = GlobalWaypointExtRec └ record name = TravelSpeedExtRec └ record name = MaxDistanceRec (optional=true) └ sequence name = MoveToLocalPosSeq └ record name = LocalWaypointExtRec └ record name = TravelSpeedExtRec └ record name = MaxDistanceRec (optional=true) └ sequence name = RetrotraverseParameterSeq └ record name = RetrotraverseRec └ variant name = RetrotraverseVar (vtag_field = unsigned byte) └ record name = DistanceRec └ record name = GlobalWaypointExtRec └ record name = LocalWaypointExtRec └ record name = RunSpoolRec └ sequence name = CommsRegainedSeq └ record name = CommsRegainedRec </pre>					
Record Name = PolicyRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestID	unsigned byte	one	false	A request identifier specified by the client that will be returned in the response message. This field allows the client to coordinate requests and responses.

Record Name = CommsLostDetectionRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> CommsLostTimeout	float	second	false	Number of seconds of no contact after which communication is considered lost and the policy is initiated. Note that implementations that monitor for comms lost at discrete intervals may round this value up to the nearest supported interval. For example, an implementation that monitors connectivity at 10 Hz would treat any timeout below 0.1 seconds as 0.1 seconds.
Record Name = StopMissionVar					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> StopMission	unsigned byte	one	false	Stop mission execution. The exact behavior may depend on the capabilities of the platform. For example, ground vehicles may stop completely, while fixed wing aircraft might loiter around the comms lost position.
Record Name = ContinueMissionVar					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> ContinueMission	unsigned byte	one	false	Allows the platform to continue mission execution, even in the absence of communications.
Record Name = GlobalWaypointExtRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> Latitude	unsigned integer	degree	false	Decimal degrees (scaled range = [-90,90], round)
3	<fixed_field> Longitude	unsigned integer	degree	false	Decimal degrees (scaled range = [-180,180], round)
4	<variable_field> ZPosition	variable	N/A	true	Index = 0 AltitudeAGL Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 1 AltitudeMSL Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 2 AltitudeASF Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 3 Depth Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round)
5	<fixed_field> Roll	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
6	<fixed_field> Pitch	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
7	<fixed_field> Heading	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
8	<fixed_field> WaypointTolerance	unsigned integer	meter	true	As defined in AS6009 (scaled range = [0,10000], round)
9	<fixed_field> PathTolerance	unsigned integer	meter	true	As defined in AS6009 (scaled range = [0,100000], round)

Record Name = LocalWaypointExtRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> X	unsigned integer	meter	false	(scaled range = [-20000000,20000000], round)
3	<fixed_field> Y	unsigned integer	meter	false	(scaled range = [-20000000,20000000], round)
4	<variable_field> ZPosition	variable	N/A	true	Index = 0 AltitudeAGL Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 1 AltitudeMSL Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 2 AltitudeASF Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 3 Depth Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 4 ZLocal Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round)
5	<fixed_field> Roll	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
6	<fixed_field> Pitch	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
7	<fixed_field> Heading	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
8	<fixed_field> WaypointTolerance	unsigned integer	meter	true	As defined in AS6009 (scaled range = [0,10000], round)
9	<fixed_field> PathTolerance	unsigned integer	meter	true	As defined in AS6009 (scaled range = [0,100000], round)
Record Name = TravelSpeedExtRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<variable_field> Speed	variable	N/A	false	Index = 0 SpeedOverGround Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round) Index = 1 SpeedRelativeToMedium Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round) Index = 2 RotationsPerMinute Field Type = unsigned short integer Field Units = rpm (scaled range = [-1000,1000], round) Index = 3 IndicatedSpeed Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round)

Record Name = MaxDistanceRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> MaxDistance	unsigned integer	meter	false	The maximum distance to travel while trying to regain comms. Once this distance has been traversed the recovery will stop. The exact behavior may depend on the capabilities of the platform. For example, ground vehicles may stop completely, while fixed wing aircraft might loiter around this position. (scaled range = [0,1000000], round)
Record Name = RetrotraverseRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<variable_field> MaxSpeed	variable	N/A	false	The maximum traversal speed. A value of zero means the vehicle should not exceed the speed of original travel. Index = 0 SpeedOverGround Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round) Index = 1 SpeedRelativeToMedium Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round) Index = 2 RotationsPerMinute Field Type = unsigned short integer Field Units = rpm (scaled range = [-1000,1000], round) Index = 3 IndicatedSpeed Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round)
2	<bit_field> Parameters	unsigned byte	bit_field	false	Bits 0..0, Direction: This bit defines the direction of motion used in retrotraverse. If the bit is set, the vehicle will retrotraverse with positive velocity (conventional direction of travel). If the bit is not set, the vehicle will travel with negative velocity (in reverse). Value set, offset=false, ranges/enums: 1= REVERSE_ORIENTATION 0= TRAVEL_BACKWARDS Bits 1..1, AltitudeChange: This bit dictates how the platform should achieve the desired altitude for systems capable of three dimensional movement. If the bit is set, the vehicle will immediately ascend/descend to the final altitude. If the bit is not set, the vehicle will follow the altitude of the previous path. Value set, offset=false, ranges/enums: 1= IMMEDIATE 0= FOLLOW_TRACK Bits 2..2, LoopRemoval: This bit dictates if loops should be removed from the retrotraverse action. Value set, offset=false, ranges/enums: 1= REMOVE_LOOPS 0= AS_TRAVELLED
3	<fixed_field> StandoffDistance	unsigned integer	meter	false	Exclusion radius around the destination point. The vehicle may not approach closer than this distance. The exact behavior may depend on the capabilities of the platform. For example, ground vehicles may stop completely at the edge of this exclusion zone, while fixed wing aircraft might loiter treating it as a no-go zone. (scaled range = [0,100000], round)

Record Name = DistanceRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> Distance	unsigned integer	meter	false	Distance along the path to retrotraverse. (scaled range = [0,1000000], round)
Record Name = RunSpoolRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<bit_field> SpoolerComponent	unsigned integer	bit_field	false	Identifier of the component hosting the Mission Spooler Service Bits 0..7, ComponentID: Value set, offset=false, ranges/enums: [1,254] Bits 8..15, NodeID: Value set, offset=false, ranges/enums: [1,254] Bits 16..31, SubsystemID: Value set, offset=false, ranges/enums: [1,65534]
2	<fixed_field> SpoolID	unsigned byte	one	false	ID of the spool to be run
3	<fixed_field> StartingTaskStep	unsigned byte	one	false	Specifies which task in the zero-indexed task list to start execution with, where 0 is the first in the list
Record Name = CommsRegainedRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> CommsRegainedTimeout	float	second	false	Number of seconds of communication with previous controlling client to consider communications regained. The exact criteria for such an event are implementation dependent. Note that implementations that monitor for comms regained at discrete intervals may round this value up to the nearest supported interval. For example, an implementation that monitors connectivity at 10 Hz would treat any timeout below 0.1 seconds as 0.1 seconds.
2	<fixed_field> CommsRegainedBehavior	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= DoNothing (Continue comms-lost behavior) 1= StopRecovery (Stop any recovery action. The exact behavior may depend on the capabilities of the platform. For example, ground vehicles may stop completely, while fixed wing aircraft might loiter around the comms regained position.)

4.1.5.2.2 ID 2C02: QueryCommsLostCapabilities

This message is used to query for a list of available behaviors on a comms-lost event.

Table 6 - QueryCommsLostCapabilities message encoding

body └ (empty)
Empty message body

4.1.5.2.3 ID 2C03: QueryCommsLostConfiguration

This message is used to query for the currently configured comms-lost behavior.

Table 7 - QueryCommsLostConfiguration message encoding

body └ (empty)
Empty message body

4.1.5.2.4 ID 2C04: QueryCommsLostStatus

This message is used to query for the current comms-lost behavior status.

Table 8 - QueryCommsLostStatus message encoding

body └ (empty)
Empty message body

4.1.5.3 Output Set

4.1.5.3.1 ID 4C01: SetCommsLostPolicyResponse

This message is sent as a response to a SetCommsLostPolicy message.

Table 9 - SetCommsLostPolicyResponse message encoding

body └ record name = CommsLostResponseRec					
Record Name = CommsLostResponseRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> RequestID	unsigned byte	one	false	A request identifier sent in the original set message. This field allows the client to coordinate requests and responses.
2	<fixed_field> CommsLostResult	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= Success 1= NotSupported (A requested policy or argument values are not supported by the implementation) 2= PointNotOnPath (A waypoint was specified for retrotraverse, but that point does not lay on the travelled path) 3= SpoolerNotFound (Mission Spooler not found at specified JAUS ID.)

4.1.5.3.2 ID 4C02: ReportCommsLostCapabilities

This message is used to report which behaviors are supported on a comms-lost event.

Table 10 - ReportCommsLostCapabilities message encoding

body └ record name = CommsLostCapabilitiesRec					
Record Name = CommsLostCapabilitiesRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<bit_field> SupportedBehaviors	unsigned byte	bit_field	false	Bits 0..0, Stop: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 1..1, ContinueMission: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 2..2, MoveToGlobalPos: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 3..3, MoveToLocalPos: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 4..4, Retrotraverse: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED Bits 5..5, RunSpool: Value set, offset=false, ranges/enums: 0= NOT_SUPPORTED 1= SUPPORTED

4.1.5.3.3 ID 4C03: ReportCommsLostConfiguration

This message is used to report the currently configured behavior for comms-lost.

Table 11 - ReportCommsLostConfiguration message encoding

<pre> body ├── sequence name = ReportPolicySeq │ ├── sequence name = CommsLostSeq │ │ ├── record name = CommsLostDetectionRec │ │ │ └── variant name = CommsLostPolicyVar │ │ │ (vtag_field = unsigned byte) │ │ │ ├── record name = StopMissionVar │ │ │ ├── record name = ContinueMissionVar │ │ │ ├── sequence name = MoveToGlobalPosSeq │ │ │ │ ├── record name = GlobalWaypointExtRec │ │ │ │ ├── record name = TravelSpeedExtRec │ │ │ │ └── record name = MaxDistanceRec (optional=true) │ │ │ ├── sequence name = MoveToLocalPosSeq │ │ │ │ ├── record name = LocalWaypointExtRec │ │ │ │ ├── record name = TravelSpeedExtRec │ │ │ │ └── record name = MaxDistanceRec (optional=true) │ │ │ ├── sequence name = RetrotraverseParameterSeq │ │ │ │ ├── record name = RetrotraverseRec │ │ │ │ └── variant name = RetrotraverseVar │ │ │ │ (vtag_field = unsigned byte) │ │ │ │ ├── record name = DistanceRec │ │ │ │ ├── record name = GlobalWaypointExtRec │ │ │ │ └── record name = LocalWaypointExtRec │ │ │ └── record name = RunSpoolRec │ └── sequence name = CommsRegainedSeq │ └── record name = CommsRegainedRec </pre>					
Record Name = CommsLostDetectionRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> CommsLostTimeout	unsigned short integer	second	false	Number of seconds of no contact after which communication is considered lost and the policy is initiated. Note that implementations that monitor for comms lost at discrete intervals may round this value up to the nearest supported interval. For example, an implementation that monitors connectivity at 10 Hz would treat any timeout below 0.1 seconds as 0.1 seconds.
Record Name = StopMissionVar					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> StopMission	unsigned byte	one	false	Stop mission execution. The exact behavior may depend on the capabilities of the platform. For example, ground vehicles may stop completely, while fixed wing aircraft might loiter around the comms lost position.
Record Name = ContinueMissionVar					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> ContinueMission	unsigned byte	one	false	Allows the platform to continue mission execution, even in the absence of communications.

Record Name = GlobalWaypointExtRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> Latitude	unsigned integer	degree	false	Decimal degrees (scaled range = [-90,90], round)
3	<fixed_field> Longitude	unsigned integer	degree	false	Decimal degrees (scaled range = [-180,180], round)
4	<variable_field> ZPosition	variable	N/A	true	Index = 0 AltitudeAGL Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 1 AltitudeMSL Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 2 AltitudeASF Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 3 Depth Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round)
5	<fixed_field> Roll	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
6	<fixed_field> Pitch	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
7	<fixed_field> Heading	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
8	<fixed_field> WaypointTolerance	unsigned integer	meter	true	As defined in AS6009 (scaled range = [0,10000], round)
9	<fixed_field> PathTolerance	unsigned integer	meter	true	As defined in AS6009 (scaled range = [0,100000], round)
Record Name = LocalWaypointExtRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<presence_vector>	unsigned byte			
2	<fixed_field> X	unsigned integer	meter	false	(scaled range = [-20000000,20000000], round)
3	<fixed_field> Y	unsigned integer	meter	false	(scaled range = [-20000000,20000000], round)

4	<variable_field> ZPosition	variable	N/A	true	Index = 0 AltitudeAGL Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 1 AltitudeMSL Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 2 AltitudeASF Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 3 Depth Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round) Index = 4 ZLocal Field Type = unsigned integer Field Units = meter (scaled range = [-50000,50000], round)
5	<fixed_field> Roll	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
6	<fixed_field> Pitch	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
7	<fixed_field> Heading	unsigned short integer	radian	true	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
8	<fixed_field> WaypointTolerance	unsigned integer	meter	true	As defined in AS6009 (scaled range = [0,10000], round)
9	<fixed_field> PathTolerance	unsigned integer	meter	true	As defined in AS6009 (scaled range = [0,100000], round)

Record Name = TravelSpeedExtRec

Field #	Name	Type	Units	Optional	Interpretation
1	<variable_field> Speed	variable	N/A	false	Index = 0 SpeedOverGround Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round) Index = 1 SpeedRelativeToMedium Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round) Index = 2 RotationsPerMinute Field Type = unsigned short integer Field Units = rpm (scaled range = [-1000,1000], round) Index = 3 IndicatedSpeed Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round)

Record Name = MaxDistanceRec

Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> MaxDistance	unsigned integer	meter	false	The maximum distance to travel while trying to regain comms. Once this distance has been traversed the recovery will stop. The exact behavior may depend on the capabilities of the platform. For example, ground vehicles may stop completely, while fixed wing aircraft might loiter around this position. (scaled range = [0,1000000], round)

Record Name = RetrotraverseRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<variable_field> MaxSpeed	variable	N/A	false	The maximum traversal speed. A value of zero means the vehicle should not exceed the speed of original travel. Index = 0 SpeedOverGround Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round) Index = 1 SpeedRelativeToMedium Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round) Index = 2 RotationsPerMinute Field Type = unsigned short integer Field Units = rpm (scaled range = [-1000,1000], round) Index = 3 IndicatedSpeed Field Type = unsigned short integer Field Units = meter per second (scaled range = [-327.68,327.67], round)
2	<bit_field> Parameters	unsigned byte	bit_field	false	Bits 0..0, Direction: This bit defines the direction of motion used in retrotraverse. If the bit is set, the vehicle will retrotraverse with positive velocity (conventional direction of travel). If the bit is not set, the vehicle will travel with negative velocity (in reverse). Value set, offset=false, ranges/enums: 1= REVERSE_ORIENTATION 0= TRAVEL_BACKWARDS Bits 1..1, AltitudeChange: This bit dictates how the platform should achieve the desired altitude for systems capable of three dimensional movement. If the bit is set, the vehicle will immediately ascend/descend to the final altitude. If the bit is not set, the vehicle will follow the altitude of the previous path. Value set, offset=false, ranges/enums: 1= IMMEDIATE 0= FOLLOW_TRACK Bits 2..2, LoopRemoval: This bit dictates if loops should be removed from the retrotraverse action. Value set, offset=false, ranges/enums: 1= REMOVE_LOOPS 0= AS_TRAVELLED
3	<fixed_field> StandoffDistance	unsigned integer	meter	false	Exclusion radius around the destination point. The vehicle may not approach closer than this distance. The exact behavior may depend on the capabilities of the platform. For example, ground vehicles may stop completely at the edge of this exclusion zone, while fixed wing aircraft might loiter treating it as a no-go zone. (scaled range = [0,100000], round)
Record Name = DistanceRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> Distance	unsigned integer	meter	false	Distance along the path to retrotraverse. (scaled range = [0,1000000], round)

Record Name = RunSpoolRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<bit_field> SpoolerComponent	unsigned integer	bit_field	false	Identifier of the component hosting the Mission Spooler Service Bits 0..7, ComponentID: Value set, offset=false, ranges/enums: [1,254] Bits 8..15, NodeID: Value set, offset=false, ranges/enums: [1,254] Bits 16..31, SubsystemID: Value set, offset=false, ranges/enums: [1,65534]
2	<fixed_field> SpoolID	unsigned byte	one	false	ID of the spool to be run
3	<fixed_field> StartingTaskStep	unsigned byte	one	false	Specifies which task in the zero-indexed task list to start execution with, where 0 is the first in the list
Record Name = CommsRegainedRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> CommsRegainedTimeout	unsigned short integer	second	false	Number of seconds of communication with previous controlling client to consider communications regained. The exact criteria for such an event are implementation dependent. Note that implementations that monitor for comms regained at discrete intervals may round this value up to the nearest supported interval. For example, an implementation that monitors connectivity at 10 Hz would treat any timeout below 0.1 seconds as 0.1 seconds.
2	<fixed_field> CommsRegainedBehavior	unsigned byte	one	false	Value set, offset=false, ranges/enums: 0= DoNothing (Continue comms-lost behavior) 1= StopRecovery (Stop any recovery action. The exact behavior may depend on the capabilities of the platform. For example, ground vehicles may stop completely, while fixed wing aircraft might loiter around the comms regained position.)

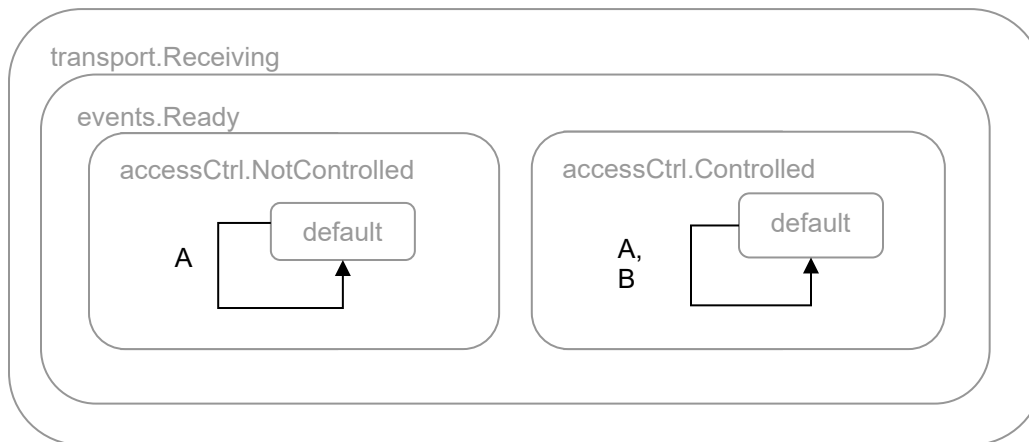
4.1.5.3.4 ID 4C04: ReportCommsLostStatus

This message is used to report the current status of the comms-lost policy. Values are default (policy not specified by external client), set (policy is set but not engaged), active (policy has triggered and vehicle is executing said policy) and restored (the comms restored policy is active) .

Table 12 - ReportCommsLostStatus message encoding

body └ record name = CommsLostStatusRec					
Record Name = CommsLostStatusRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> CommsLostStatus	byte	one	false	Value set, offset=false, ranges/enums: 0= Default 1= Set 2= Active 3= Restored

4.1.6 Protocol Behavior

**Figure 1 - CommsLostPolicyManager service protocol behavior****Table 13 - CommsLostPolicyManager service state transitions**

Start State	Trigger	Conditions	Actions
A	QueryCommsLostCapabilities		sendReportCommsLostCapabilities (msg, transportData)
	QueryCommsLostConfiguration		sendReportCommsLostConfiguration (msg, transportData)
	QueryCommsLostStatus		sendReportCommsLostStatus (msg, transportData)
	CommsLost		executeCommsLostPolicy ()
	CommsRestored		executeCommsRegainedPolicy ()
B	SetCommsLostPolicy	isControllingClient(transportData) && !isSupportedCommand(msg)	sendSetCommsLostPolicyResponse ('NOT_SUPPORTED' ,transportData)
	SetCommsLostPolicy	isControllingClient(transportData) && isWaypointSpecified(msg) && !isOnPath(msg)	sendSetCommsLostPolicyResponse ('POINT_NOT_ON_PATH' ,transportData)
	SetCommsLostPolicy	isControllingClient(transportData) && isSupportedCommand(msg)	setCommsLostPolicy (msg) , sendSetCommsLostPolicyResponse ('SUCCESS' ,transportData)

Table 14 - CommsLostPolicyManager service conditions

Condition	Interpretation
isControllingClient(transportData) && !isSupportedCommand(msg)	True if the command message was received from the client currently controlling this component AND if the requested comms-lost behavior is NOT supported by the service.
isControllingClient(transportData) && isWaypointSpecified(msg) && !isOnPath(msg)	True if the command message was received from the client currently controlling this component AND the command message specified a Retrotraverse behavior with the destination given as a waypoint AND that waypoint is not on the path travelled by the vehicle. Note that the tolerance of what is considered on the path is not specified by this document, and may vary by implementation.
isControllingClient(transportData) && isSupportedCommand(msg)	True if the command message was received from the client currently controlling this component AND if the requested comms-lost behavior is supported by the service.

Table 15 - CommsLostPolicyManager service transition actions

Action	Interpretation
sendReportCommsLostCapabilities	Send a ReportCommsLostCapabilities message
sendReportCommsLostConfiguration	Send a ReportCommsLostConfiguration message
sendReportCommsLostStatus	Send a ReportCommsLostStatus message
executeCommsLostPolicy	Execute the configured comms-lost behavior.
executeCommsRegainedPolicy	Execute the configured comms-regained behavior.
sendSetCommsLostPolicyResponse	Send a SetCommsLostPolicyResponse message
setCommsLostPolicy	Set the specified policy as enabled.

4.2 Cost Map 2D

name=CostMap2D

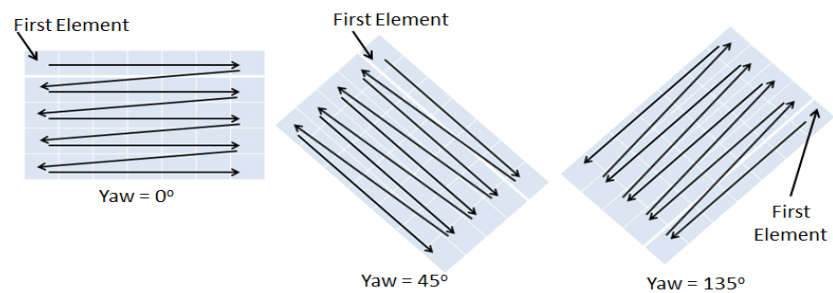
version=1.1

id=urn:jaus:jss:behaviors:CostMap2D

4.2.1 Description

The Cost Map 2D Service provides a mechanism to report obstacles, hazardous terrain, and no-go zones using an overhead 2D cost map. The cost map is represented by a list of cells, broken into a specified number of rows and columns, which in turn corresponds to the width and height (respectively) of the map in meters. Each cell within the map specifies a cost value, and in some cases, a confidence value. The cost is a relative measure of occupancy or effort to traverse the cell, such that a value of zero (0) means no occupancy or no effort, while a maximum cost value means a cell is non-traversable. The center point of the cost map may be specified in global or local coordinates, depending on the particular implementation, as defined by [AS6009]. The map may also be rotated around the coordinate frame's Z-axis by the specified rotation value.

Cells within the list must be sequenced such that the most northwest point (i.e. most forward left), relative to the non-rotated coordinate frame, is the first element. Subsequent elements represent cells in an easterly direction, until the end of the row is reached. At that point, the next row starts again on the west side of the map.



To add support for client-specified no-go (exclusion) zones, this service should be co-located (hosted on the same component) as the Global or Local Operating Zone Service.

Note that the cost of each cell is described across a range providing a low value (no cost, free cell) and a high cost (no-go, lethal cell). The interpretation of these values, as well as values in between, is not defined by the Standard, as it is expected to be vehicle- and implementation-specific. As a result, users are cautioned against using these values across multiple systems. For example, what may be considered high cost for a small, low traction ground vehicle may be trivial to a larger, more capable system.

4.2.2 Assumptions

Messages may be delayed, lost, or reordered.

4.2.3 References

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

4.2.4 Vocabulary

Table 16 - CostMap2D service message vocabulary

Message ID (hex)	Name	Command
Input Set		
2C11	QueryCostMap2D	false
Output Set		
4C11	ReportCostMap2D	false

4.2.5 Encoding

4.2.5.1 Input Set

4.2.5.1.1 ID 2C11: QueryCostMap2D

This message is used to query for cost map data. An optional maximum map size may be specified to limit the amount of data returned. If no values are specified, the service may still return a subset of the cost map stored internally, with emphasis on the immediate vicinity of the platform. An optional map center and orientation may be specified. If the map center lies outside the bounds of the known cost map, the data returned should reflect the unknown (zero confidence) elements. If the map center is specified in an unsupported coordinate frame, or the map center is not specified, the returned map should be centered the current vehicle location. Note that the implementation should seek to satisfy these requests, but it is not strictly required to do so.

Table 17 - QueryCostMap2D message encoding

<pre> body └─ sequence name = QueryCostMap2DSeq └─ record name = QueryCostMap2DResolutionRec (optional=true) └─ record name = QueryCostMap2DSizeRec (optional=true) └─ variant name = QueryCostMap2DCenterVar (optional=true) (vtag_field = unsigned byte) └─ record name = CostMap2DGlobalPoseRec └─ record name = CostMap2DLocalPoseRec </pre>					
Record Name = QueryCostMap2DResolutionRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> MaxNumberOfRows	unsigned short integer	one	false	Maximum number of rows in the returned data.
2	<fixed_field> MaxNumberOfColumns	unsigned short integer	one	false	Maximum number of columns in the returned data.
Record Name = QueryCostMap2DSizeRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> MaxWidth	unsigned integer	meter	false	(scaled range = [0,500000], round)
2	<fixed_field> MaxHeight	unsigned integer	meter	false	(scaled range = [0,500000], round)
Record Name = CostMap2DGlobalPoseRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> MapCenterLatitude	unsigned integer	degree	false	(scaled range = [-90,90], round)
2	<fixed_field> MapCenterLongitude	unsigned integer	degree	false	(scaled range = [-180,180], round)
3	<fixed_field> MapRotation	unsigned short integer	radian	false	Rotation about the global Z axis. (scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
Record Name = CostMap2DLocalPoseRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> MapCenterX	unsigned integer	meter	false	(scaled range = [-100000,100000], round)
2	<fixed_field> MapCenterY	unsigned integer	meter	false	(scaled range = [-100000,100000], round)
3	<fixed_field> MapRotation	unsigned short integer	radian	false	Rotation about the local Z axis. (scaled range = [-3.14159265358979323846,3.14159265358979323846], round)

4.2.5.2 Output Set

4.2.5.2.1 ID 4C11: ReportCostMap2D

This message is used to report the cost map data in accordance to the received QueryCostMap2D message. While the implementation should seek to satisfy the conditions given in the query message, it is not strictly required to do so. For example, for systems that do not interpolate cost information, the reported resolution may be limited to the resolution of the stored map data rather than the requested resolution. Similarly, the size of the returned map may be limited by internal storage rather than the requested values.

Table 18 - ReportCostMap2D message encoding

<pre> body ├── sequence name = CostMap2DSeq │ ├── record name = CostMap2DRec │ ├── variant name = CostMap2DPoseVar │ │ ├── (vtag_field = unsigned byte) │ │ ├── record name = CostMap2DGlobalPoseRec │ │ └── record name = CostMap2DLocalPoseRec │ └── variant name = CostMap2DDataVar │ ├── (vtag_field = unsigned byte) │ ├── list name = CostDataList │ │ ├── (count_field = unsigned short integer) │ │ └── record name = CostDataRec │ ├── list name = CostAndConfidenceDataList │ │ ├── (count_field = unsigned short integer) │ │ └── record name = CostAndConfidenceDataRec │ ├── list name = RunLengthEncodedDataList │ │ ├── (count_field = unsigned short integer) │ │ └── record name = RunLengthEncodedDataRec │ └── list name = CostOrAttractivenessList │ ├── (count_field = unsigned short integer) │ └── record name = CostOrAttractivenessRec </pre>					
Record Name = CostMap2DRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> NumberOfRows	unsigned short integer	one	false	Number of rows in the map data.
2	<fixed_field> NumberOfColumns	unsigned short integer	one	false	Number of columns in the map data.
3	<fixed_field> MapWidth	unsigned integer	meter	false	Total width of the cost map. (scaled range = [0,500000], round)
4	<fixed_field> MapHeight	unsigned integer	meter	false	Total height of the cost map. (scaled range = [0,500000], round)
Record Name = CostMap2DGlobalPoseRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> MapCenterLatitude	unsigned integer	degree	false	(scaled range = [-90,90], round)
2	<fixed_field> MapCenterLongitude	unsigned integer	degree	false	(scaled range = [-180,180], round)
3	<fixed_field> MapRotation	unsigned short integer	radian	false	(scaled range = [-3.14159265358979323846, 3.14159265358979323846], round)

Record Name = CostMap2DLocalPoseRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> MapCenterX	unsigned integer	meter	false	(scaled range = [-100000,100000], round)
2	<fixed_field> MapCenterY	unsigned integer	meter	false	(scaled range = [-100000,100000], round)
3	<fixed_field> MapRotation	unsigned short integer	radian	false	(scaled range = [-3.14159265358979323846,3.14159265358979323846], round)
Record Name = CostDataRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> Cost	unsigned byte	one	false	Value 0-254: A value of 0 indicates no cost (no restriction). A value of 254 means maximum cost (non-traversable or no-go zone). A value of 255 means the cost of the cell is unknown. Value set, offset=false, ranges/enums: [0,254] [255,255]
Record Name = CostAndConfidenceDataRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<fixed_field> Cost	unsigned byte	one	false	Value 0-254: Cost value associated with the cell. A value of 0 indicates no cost (no restriction). A value of 254 means maximum cost (non-traversable or no-go zone). A value of 255 means the cost of the cell is unknown. Value set, offset=false, ranges/enums: [0,254] [255,255]
2	<fixed_field> Confidence	unsigned byte	one	false	Confidence value associated with the cost of the cell. A value of 0 means there is no confidence in the specified cost (unknown cost). A value of 100 means the cost of the cell is known perfectly. (scaled range = [0,100], round)
Record Name = RunLengthEncodedDataRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<bit_field> CostAndRunLength	unsigned short integer	bit_field	false	Bits 0..3, CostSubField: Cost value associated with the cell. A value of 0 indicates no cost (no restriction). A value of 14 means maximum cost (non-traversable or no-go zone). A value of 15 means the cost of the cell is unknown Value set, offset=false, ranges/enums: [0,14] [15,15] Bits 4..15, NumberCellsSubField: The number of contiguous cells, including this one, at this specified cost and certainty. Value set, offset=false, ranges/enums: [0,4095]

Record Name = CostOrAttractivenessRec					
Field #	Name	Type	Units	Optional	Interpretation
1	<bit_field> CostOrAttractiveness	unsigned byte	bit_field	false	Bits 0..4, CostOrAttractivenessSubField: Cost value associated with the cell. A value of 16 indicates no cost or attractiveness. A value of 31 means maximum cost (non-traversable or no-go zone). A value of 0 means maximum attractiveness (negative cost). Value set, offset=false, ranges/enums: [0,31] Bits 5..5, NoGoCell: If this bit is set, the cell has been marked as a no-go zone by an external client. Value set, offset=false, ranges/enums: 0= CellsNotNoGo 1= CellsNoGo Bits 6..6, TraversedCell: If this bit is set, the cell has been traversed by the vehicle previously. Value set, offset=false, ranges/enums: 0= CellHasNotBeenTraversed 1= CellHasBeenTraversed Bits 7..7, UnknownCell: If this bit is set, the cell has not been evaluated. All other information is invalid. Value set, offset=false, ranges/enums: 0= CellsKnown 1= CellsNotKnown

4.2.6 Protocol Behavior

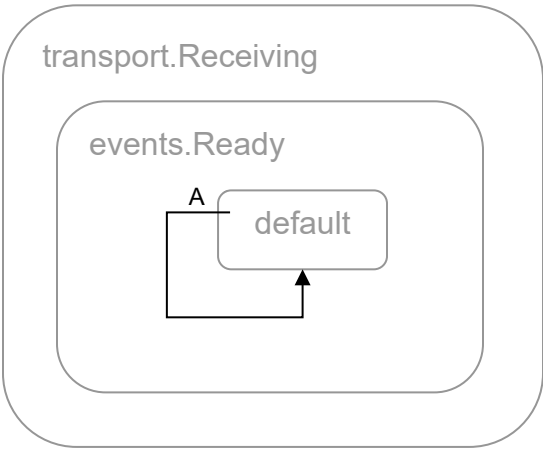


Figure 2 - CostMap2D service protocol behavior

Table 19 - CostMap2D service state transitions

Start State	Trigger	Conditions	Actions
A	QueryCostMap2D		sendReportCostMap2D (msg,transportData)

Table 20 - CostMap2D service transition actions

Action	Interpretation
sendReportCostMap2D	Send a Report Cost Map 2D message

4.3 Path Reporter

name=PathReporter

version=1.1

id=urn:jaus:jss:behaviors:PathReporter

4.3.1 Description

The Path Reporter Service provides a mechanism for reporting the past and/or future expected path of the vehicle. The service is used in cooperation with Global Waypoint Driver, Local Waypoint Driver, Global Waypoint List Driver, Local Waypoint List Driver, Global Pose Sensor and/or Local Pose Sensor. Clients may limit the amount of data reported by the service by specifying a maximum number of data points, maximum time, maximum distance along the path, and/or path resolution, within the limits of the implementation's reported capabilities. Note that the historical path is assumed to be represented by a FIFO queue; as a result, the Report Path message may be limited by the storage capabilities of the underlying implementation to reporting only the most recent data, e.g. the points nearest the current position. Older data may be discarded as needed by the implementation. Such limits should be specified in the ReportPathReporterCapabilities message. Also note that the future path may be valid only at that instance in time. It represents the current planned (future) path, at the given resolution. However, some waypoint drivers may frequently update the planned path, based on new information about the environment. As a result, planned path information may quickly become stale. Furthermore, the planned path represents the actual detailed path the vehicle expects to follow. It does not replace the Waypoint Driver's Query Waypoint messages. Rather, it provides additional information about how the vehicle plans to achieve the desired waypoints.

4.3.2 Assumptions

Messages may be delayed, lost or reordered.

4.3.3 References

Inherits-from=Events

version=1.1

id=urn:jaus:jss:core:Events

4.3.4 Vocabulary

Table 21 - PathReporter service message vocabulary

Message ID (hex)	Name	Command
Input Set		
2C40	QueryPathReporterCapabilities	false
2C41	QueryPath	false
Output Set		
4C40	ReportPathReporterCapabilities	false
4C41	ReportPath	false