Unmanned Maritime Autonomy Architecture (UMAA) Experimental Services (EXP) Interface Control Document (ICD) (UMAA-SPEC-EXPICD)

MDE Version 4.2.1 Commit 7c2b513

UMAA Spec Commit f2fa426

Version 3.0.1
25 February 2021

DISTRIBUTION STATEMENT A: Approved for public release. Distribution is unlimited.

UMAA Experimental Services ICD

(UMAA-SPEC-EXP-ICD)

Signature Page

Submitted by: Rothgeb.Mark.E.ORC30 11008580.ID Digitally signed by Rothgeb.Mark.E.ORC3011008580.ID Date: 2021.05.05 16:29:25 -04'00' **Date Signed:** ____________

Mark Rothgeb
Unmanned Maritime Autonomy Architecture
Standards Board Chair

PMS 406 Approvals: ANDERSON.JEREMI AH.P.1254266481 Digitally signed by ANDERSON.JEREMIAH.P.1254266481 Date: 2021.05.10 14:49:47 -04'00' **Date Signed:** ____________

CDR Jeremiah Anderson
PMS 406 Advanced Autonomous Capabilities,
Principal Assistant Program Manager

SMALL.PETER. D.1048778646 Digitally signed by SMALL.PETER.D.1048778646 Date: 2021.06.03 17:17:23 -04'00' **Date Signed:** ____________

CAPT Pete Small
Program Manager
PMS 406 Unmanned Maritime Systems,
PEO Unmanned and Small Combatants

Contents

List of Figures

List of Tables

1 Scope

1.1 Identification

This document defines a set of ***experimental*** services as part of the Unmanned Maritime Autonomy Architecture (UMAA)—experimental services are not required to satisfy UMAA compliance, but are provided to industry for feedback. As such, it provides services that are in an experimental state and are in the process of being developed. This document is generated automatically from data models that define its services and their interfaces as part of the Unmanned Systems (UxS) Control Segment (UCS) Architecture as extended by UMAA to provide autonomy services for UMVs.

To put each ICD in context of the UMAA Architecture Design Description (ADD), the UMAA functional decomposition mapping to UMAA ICDs is shown in Figure 1.

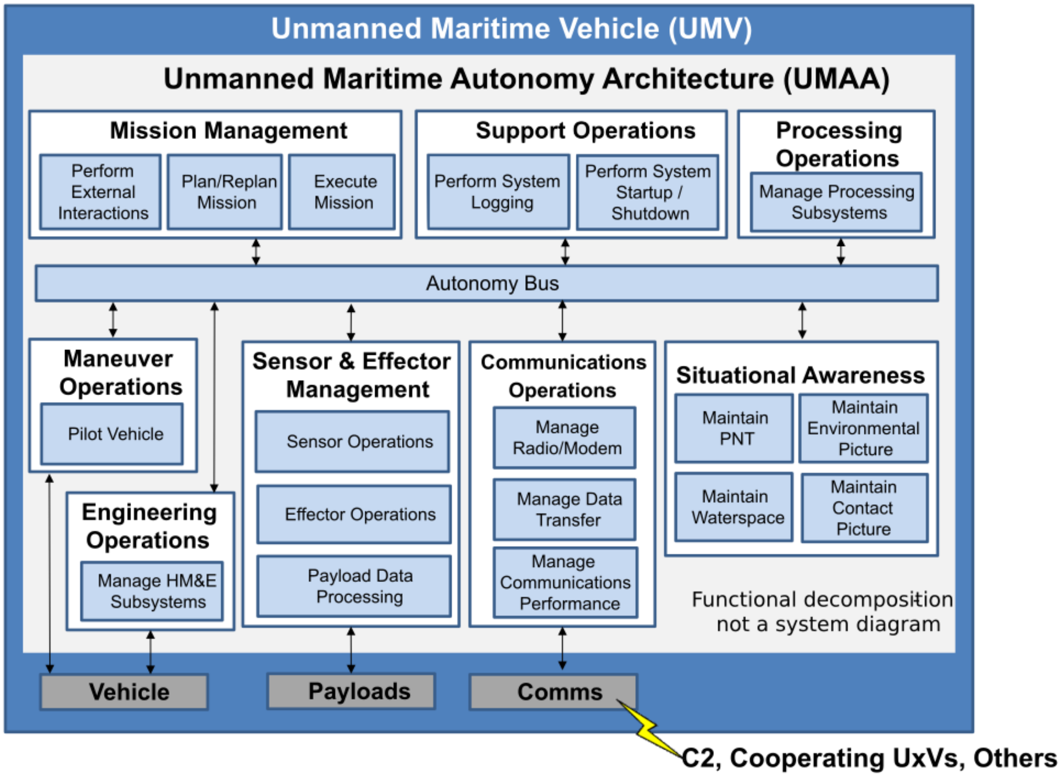


Figure 1: UMAA Functional Organization

1.2 Overview

The fundamental purpose of UMAA is to promote the development of common, modular, and scalable software for UMV's that is independent of a particular autonomy implementation. Unmanned Maritime Systems (UMSs) consist of Command and Control (C2), one or more UMVs, and support equipment and software (e.g. recovery system, Post Mission Analysis applications). The scope of UMAA is focused on the autonomy that resides on-board the UMV. This includes the autonomy for all classes of UMVs and must support varying levels of communication in mission (i.e., constant, intermittent, or none) with its C2 System. To enable modular development and upgrade of the functional capabilities of the on-board autonomy, UMAA defines eight high-level functions. These core functions include: Communications Operations, Engineering Operations, Maneuver Operations, Mission Management, Processing Operations, Sensor and Effector Operations, Situational Awareness, and Support Operations. In each of these areas, it is anticipated that new capabilities will be required to satisfy evolving Navy missions over time. UMAA seeks to define standard interfaces for these functions so that individual programs can leverage capabilities developed to these standard interfaces across programs that meet the standard interface specifications. Individual programs may group services and interfaces into components in different ways to serve their particular vehicle's needs. However, the entire interface defined by UMAA will be required as defined in the ICDs for all services that are included in a component. This requirement is what enables autonomy software to be ported between heterogeneous UMAA-compliant vehicles with their disparate vendor-defined vehicle control interfaces without recoding to a vehicle specific platform interface.

Experimental Services defines the services that are still under early state development. Figure 2 depicts an example of a possible component service grouping is shown with the dashed lines.

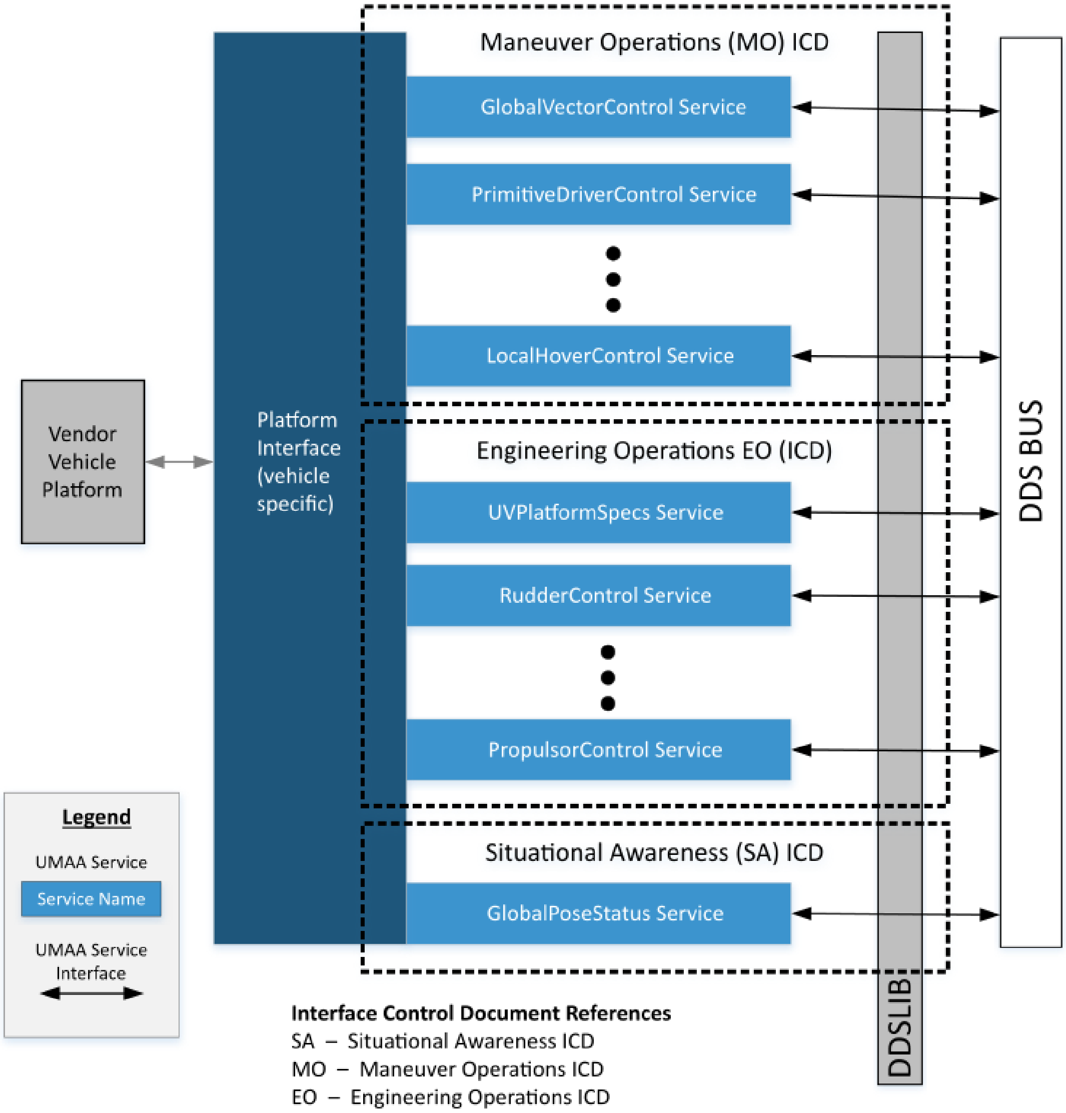

Maneuver Operations (MO) ICD
GlobalVectorControl Service
PrimitiveDriverControl Service
LocalHoverControl Service
Engineering Operations EO (ICD)
UVPlatformSpecs Service
RudderControl Service
PropulsorControl Service
Situational Awareness (SA) ICD
GlobalPoseStatus Service
Vendor Vehicle Platform
Platform Interface (vehicle specific)
DDS BUS
DDSLIB
Legend
UMAA Service
Service Name
UMAA Service Interface
Interface Control Document References
SA – Situational Awareness ICD
MO – Maneuver Operations ICD
EO – Engineering Operations ICD

Figure 2: UMAA Services and Interfaces Example

1.3 Document Organization

This interface control document is organized as follows:

Section 1 – Scope: A brief purview of this document

Section 2 – Referenced Documents: A listing of associated of government and non-government documents and standards

Section 3 – Introduction to Data Model, Services, and Interfaces: A description of the common data model across all services and interfaces

Section 4 – Introduction to Coordinate Reference Frames and Position Model: An overview of the reference frame model used by UMAA

Section 5 – Flow Control: A description of different flow control patterns used throughout UMAA.

Section 6 – Experimental Services (EXP) Services and Interfaces: A description of specific services and interfaces for this ICD

2 Referenced Documents

The documents in the following table were used in the creation of the UMAA interface design documents. Not all references may be applicable to this particular document.

Table 3: Standards Documents

Title	**Release Date**
A Universally Unique IDentifier (UUID) URN Namespace	July 2005
Data Distribution Service for Real-Time Systems Specification, Version 1.4	March 2015
Data Distribution Service Interoperability Wire Protocol (DDSI-RTPS), Version 2.3	April 2019
Object Management Group Interface Definition Language Specification (IDL)	March 2018
Extensible and Dynamic Topic Types for DDS, Version 1.3	February 2020
UAS Control Segment (UCS) Architecture, Architecture Description, Version 2.4	27 March 2015
UCS Architecture, Conformance Specification, Version 2.2	27 September 2014
UCS-SPEC-MODEL v3.4 Enterprise Architect Model	27 March 2015
UCS Architecture, Architecture Technical Governance, Version 2.5	27 March 2015
System Modeling Language Specification, Version 1.5	May 2017
Unified Modeling Language Specification, Version 2.5.1	December 2017
Interface Definition Language (IDL), Version 4.2	March 2018
U.S. Department Of Homeland Security, United States Coast Guard "Navigation Rules International-Inland" COMDTINST M16672.2D	March 1999
IEEE 1003.1-2017 - IEEE Standard for Information Technology–Portable Operating System Interface (POSIX(R)) Base Specifications, Issue 7	December 2017

Table 4: Government Documents

Title	**Release Date**
Unmanned Maritime Autonomy Architecture (UMAA) Architecture Design Description (ADD), Version 1.0	January 2019
MANUAL FOR THE SUBMISSION OF OCEANOGRAPHIC DATA COLLECTED BY UNMANNED UNDERSEA VEHICLES (UUVs)	October 2018

3 Introduction to Data Model, Services, and Interfaces

3.1 Data Model

A common data model is at the heart of UMAA. The common data model describes the entities that represent system state data, the attributes of those entities and relationships between those entities. This is a "data at rest" view of system level information. It also contains data classes that define types of messages that will be produced by components, a "data in motion" view of system level information.

The common data model and coordinated service interfaces are described in a Unified Modeling Language (UML^{TM}) modeling tool and are represented as UML^{TM} class diagrams. Interface definition source code for messages/topics and other interface definition products and documentation will be automatically generated from the common data model to assure they are consistent with the data model and to ensure delivered software matches its interface specification.

The data model is maintained as a maritime extension to the UCS Architecture and will be maintained under configuration control by the UMAA Board. Section 6 content is automatically generated from this data model as are other automated products such as IDL that are used for automated code generation.

3.2 Definitions

UMAA ICDs follow the UCS terminology definitions found in the UCS Architecture Description v2.4. The normative (required) implementation to satisfy compliance with a UMAA ICD is to provide service and interface specification compliance. Components may group services and their required interfaces in any manner so long as every service meets its interface specifications. Figure 3 shows a particular grouping of services into components. The interfaces are represented by the blue and green lines and may represent 1 or more independent input and output interfaces for each service. The implementation of the service into software components is left up to the individual system development. Compliance is satisfied at the individual service level. Given this context, section 6 correspondingly defines services with their interfaces and not components.

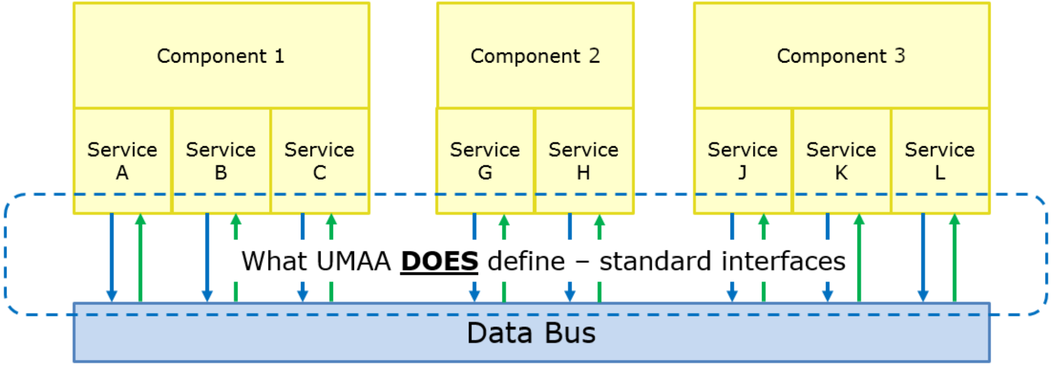


Figure 3: Services and Interfaces Exposed on the UMAA Data Bus

Services may use other services within this ICD or in other UMAA defined ICDs in order to provide their capability. Additionally, components for acquisition and development may span ICDs. An example of this would be a vehicle control system on a UMV. The control of the vehicle would be found in the Maneuver Operations ICD. However, an Inertial Navigation Unit (INU) that gives dynamic vehicle status is found in the Situational Awareness ICD. These are often organic to a vehicle and in that case are provided together with the vehicle as a component.

3.3 Data Distribution Service (DDS^{TM})

The data bus supporting autonomy messaging as depicted in figure 3 is implemented via DDS^{TM}. DDS is a middleware protocol and API standard for data-centric connectivity from the Object Management Group (OMG). It integrates the components of a system together, providing low-latency data connectivity, extreme reliability, and a scalable architecture. In a distributed system, middleware is the software layer that lies between the operating system and applications. It enables the various components of a system to more easily communicate and share data. It simplifies the development of distributed systems by letting software developers focus on the specific purpose of their applications rather than the mechanics of passing information between applications and systems. The DDS specification is fully described in free reference material on the OMG website and there are both open source and commercially available implementations.

3.4 Naming Conventions

UMAA services are modeled within the UCS Architecture under the Multi-Domain Extension (MDE). The UCS Architecture uses SoaML concepts of participant, serviceInterface, service port and request port to describe the interfaces that make up a service and show how the service is used. Each service defines the capability it provides as well as required interfaces. Each interface consists of an operation that accepts a single message (A SoaML MessageType). In SoaML, a MessageType is a defined as a unit of information exchanged between participant Request and Service ports via ServiceInterfaces. Instances of a MessageType are passed as parameters in ServiceInterface operations. (UCSArchitecture,ArchitectureTechnicalGovernance)

In order to promote commonality across service definitions, a common way of naming services and their set of operations and messages has been adopted for defining services within UCS-MDE. The convention uses the Service Base Name (SBN) and an optional Function Name (FN) to derive all service names and their associated operations and messages. As this is meant to be a guide, services might not include all of the defined operations and messages and their names might not follow the convention where a more appropriate name adds clarity.

Furthermore services in UMAA will not be broken up as indicated below when all parts of the service capabilities are required for the service to be meaningful (such as ResourceAllocation).

Additionally, note that for UMAA not all operations defined in UCS-MDE result in a message being published to the DDS bus, e.g., since DDS uses publish/subscribe, most query operations result in a subscription to a topic and do not actually publish the associated request message. In the case of cancel commands, there is no associated implementation of the cancel<SBN><FN>CommandStatus as it is just the intrinsic response of the DDS dispose function so it is essentially a NOOP in implementation. The conventions used to define UCS-MDE services are as follows:

Service Name
<SBN>Config
<SBN>Control
<SBN>Specs
<SBN>Status

where the SBN should be descriptive of the task or information provided by the service.

Table 5: Service Requests and Associated Responses

	Service Requests (Inputs)	**Service Responses (Outputs)**
Config	query<SBN><FN>Config	report<SBN><FN>Config
Control	set<SBN><FN> query<SBN><FN>CommandAck cancel<SBN><FN>Command query<SBN><FN>ExecutionStatus	report<SBN><FN>CommandStatus report<SBN><FN>CommandAck report<SBN><FN>CancelCommandStatus report<SBN><FN>ExecutionStatus
Specs	query<SBN><FN>Specs	report<SBN><FN>Specs
Status	query<SBN><FN>	report<SBN><FN>

Service Requests (operation:message)
query<SBN><FN>Config:<SBN><FN>ConfigRequestType[1]
set<SBN><FN>:<SBN><FN>CommandType
query<SBN><FN>CommandAck:<SBN><FN>CommandAckRequestType[1]
cancel<SBN><FN>Command:<SBN><FN>CancelCommandType
query<SBN><FN>ExecutionStatus:<SBN><FN>ExecutionStatusRequestType[1]
query<SBN><FN>Specs:<SBN><FN>SpecsRequestType[1]
query<SBN><FN>:<SBN><FN>RequestType [1] [2]

[1] These message types are required for compatibility with the UCS model but are not used by the UMAA specification.
[2] At this time there are no Requests in the specification but when they have been added, this will be the message format.

Service Responses (operation:message)
report<SBN><FN>Config:<SBN><FN>ConfigReportType
report<SBN><FN>CommandStatus:<SBN><FN>CommandStatusType
report<SBN><FN>CommandAck:<SBN><FN>CommandAckReportType
report<SBN><FN>CancelCommandStatus:<SBN><FN>CancelCommandStatusType
report<SBN><FN>ExecutionStatus:<SBN><FN>ExecutionStatusReportType
report<SBN><FN>Specs:<SBN><FN>SpecsReportType
report<SBN><FN>:<SBN><FN>ReportType

where,

- Config (Configuration) Report – the setup of a resource for operation of a particular task. Attributes may be static or variable. Examples include: maximum RPM allowed, operational sonar frequency range allowed, maximum allowable radio transmit power.
- Command Status – the current state of a particular command (either control or configuration)
- Command – the ability to influence or direct the behavior of a resource during operation of a particular task. Attributes are variable. Examples include a vehicleâ€™s speed, engine RPM, antenna raising/lowering, controlling a light or gong.
- Command Ack (Acknowledgement) Report – the command currently being executed.
- Cancel – the ability to cancel a particular command that has been issued.
- Execution Status Report – the status related to executing a particular command. Examples associated with a waypoint command include cross track error, time to achieve, distance remaining.
- Specs (Specifications) Report – a detailed description of a resource and/or its capabilities and constraints. Attributes are static. Examples include: maximum RPM of a motor, minimum frequency of a passive sonar sensor, length of the UMV, cycle time of a radar.
- Report – the current information provided by a resource. Examples include a vehicle speed, rudder angle, current waypoint, contact bearing.

3.5 Namespace Conventions

Each UMAA service and the messages under the service can be accessed through their appropriate UMAA namespace. The namespace reflects the mapping of a specific service to its parent ICD, and the parent ICD's mapping to the overall UMAA Design Description. For example:

Access the Primitive Driver service under Maneuver Operations:
UMAA::MO::PrimitiveDriver
Access the Feature Service under Situational Awareness:
UMAA::SA::Feature

The UMAA model uses common data types that are re-used through the model to define service interface topics, interface topics, and other common data topics. These data types are not intended to be directly utilized but for reference they can be accessed in the same manner:

Access the common UMAA Report Message Fields:
UMAA::UMAARpt
Access the common UMAA Position2D (i.e., latitude and longitude) structure:
UMAA::Measurement::Position2D

3.6 Cybersecurity

The UMAA standard addressed in this ICD is independent from defining specific measures to achieve Cybersecurity compliance. This UMAA ICD does not preclude the incorporation of security measures, nor does it imply or guarantee any level of Cybersecurity within a system. Cybersecurity compliance will be performed on a program specific basis and compliance testing is outside the scope of UMAA.

3.7 GUID algorithm

The UMAA standard utilizes the Globally Unique IDentifier (GUID), conforming to the variant defined in RFC 4122 (variant value of 2). Generators of GUIDs may generate GUIDs of any valid, RFC 4122-defined version that is appropriate for their specific use case and requirements. (Reference: A Universally Unique IDentifier (UUID) URN Namespace)

3.8 Large Sets

Some reports under the UMAA standard utilize Large Sets, which are unordered sets of related data. The purpose of a Large Set is to provide the ability to update one or more elements of the set without having to republish the entire set on the DDS bus and consuming more resources as a set is appended or updated. In a given DDS topic, each element of the set is tracked to its identifier through the use of the <service>SetID identifier (a key). Additionally, users will be able to trace an element in a set by its source attribute (a NumericGUID) to the Service Provider that is generating the report with this set.

When elements of the set are updated, the timestamp of the metadata must be updated as well to signal a change in the set. The element timestamp for the update must be later than the current metadata timestamp. Once the element is updated, the timestamp of the metadata must be updated to a time equal to or later than the timestamp of the individual element update. The set can be updated as a batch (multiple elements in a single "update cycle," as determined by the provider) provided the metadata timestamp is updated to a time that is no earlier than the the most recent timestamp of all element updates in the batch. This allows for a coarse synchronization: data elements with timestamps later than the current metadata timestamp can be assumed to be part of an in-progress update cycle. Consumers can choose to immediately act on those data individually or wait until the metadata timestamp is advanced beyond the element's timestamp to signal the complete update cycle has finished and consider the set as a whole.

4 Introduction to Coordinate Reference Frames and Position Model

4.1 Platform Reference Frame

In the following Service Definitions we use the parameters yaw, pitch, and roll to define the orientation of the vehicle with respect to the specified reference frame. Each parameter is described as a rotation around a given axis: Yaw about the Z axis. Pitch about the Y axis. Roll about the X axis.

The axes are defined as:

- X - positive in the forward direction, negative in the aft
- Y - positive in the starboard direction, negative in the port.
- Z - positive in the down direction, negative in the up.

Additionally, rotations about all axes follow the right hand rule.

4.2 Platform Orientation

Determining the orientation of the vehicle (Figure 4) with respect to any reference frame is carried out via the following procedure (Figure 5).

1. Align the vehicle's Longitudinal or X axis with the reference frame X axis. In the global reference, this is the North direction.
2. Align the vehicle's down or Down, or Z axis with the reference frame's Z axis. In the global reference frame, this is the Gravity direction.
3. Ensure that the vehicle's Transverse or Y axis is aligned with the reference frame's Y axis. In the global reference frame this is the East direction.
4. Rotate the vehicle about the vehicle's Z axis by the Yaw angle (Figure 6).
5. Rotate the vehicle about the vehicle's newly oriented Y axis by the pitch angle (Figure 7).
6. Rotate the vehicle about the vehicle's newly oriented X axis by the roll angle (Figure 8).

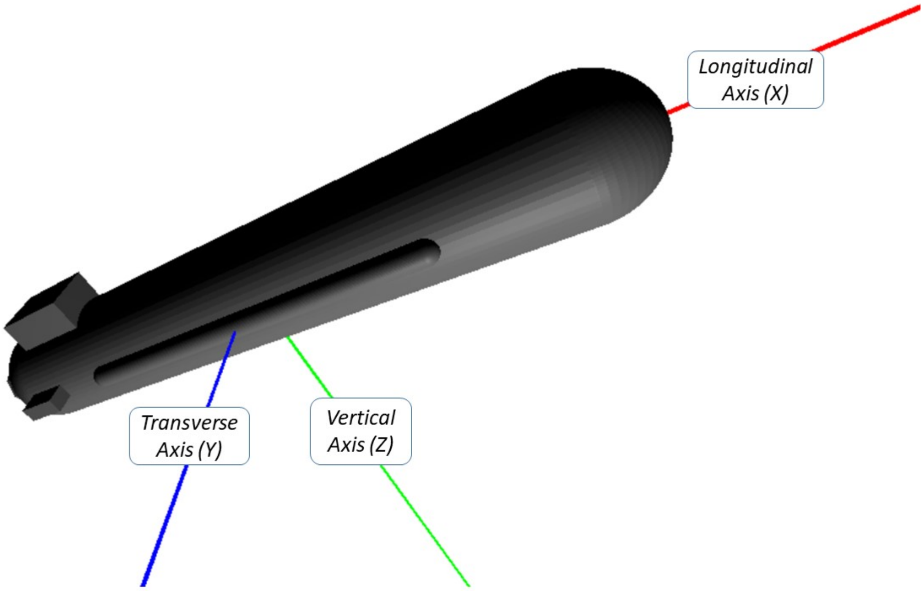


Figure 4: Given a vehicle in arbitrary orientation

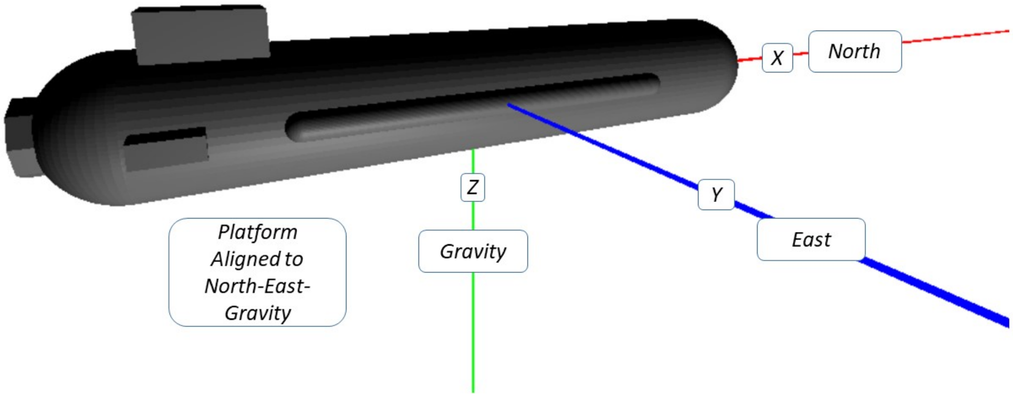


Figure 5: Align the vehicle with the reference frame axes

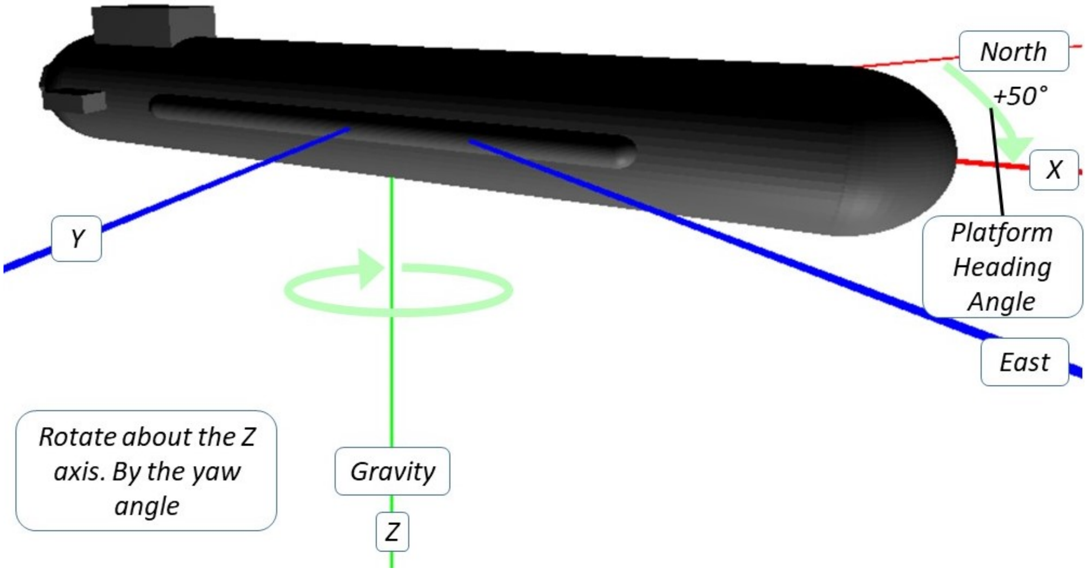


Figure 6: Rotate the vehicle by the Yaw angle

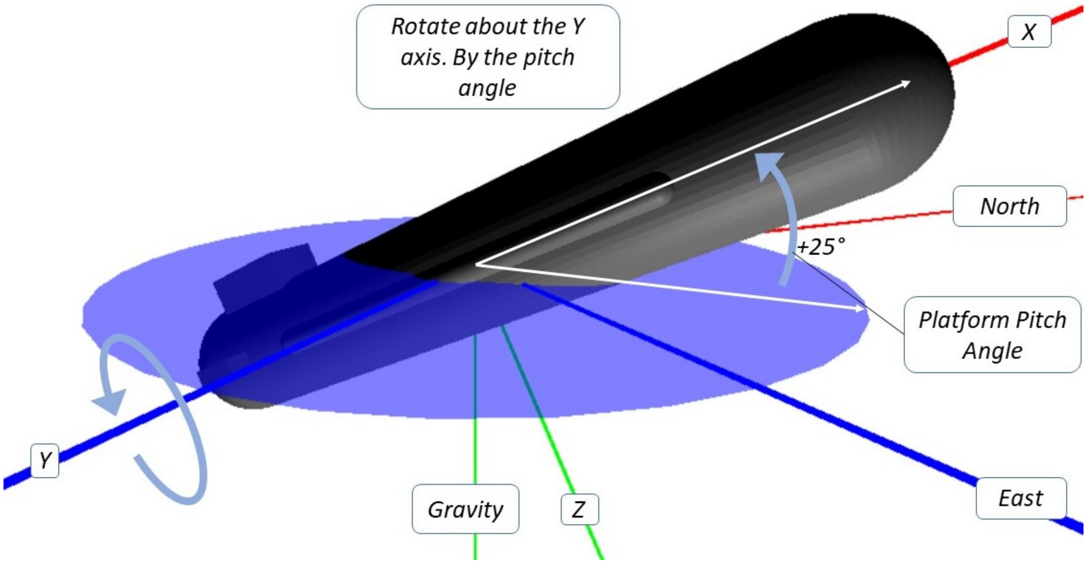


Figure 7: Rotate the vehicle by the Pitch angle

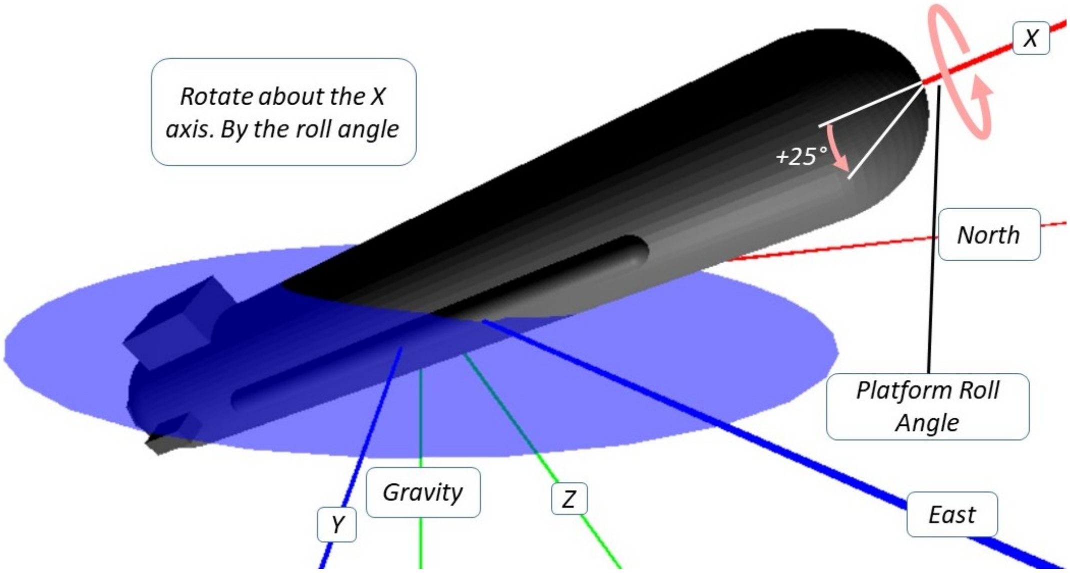


Figure 8: Rotate the vehicle by the Roll angle

4.3 Vehicle Coordinate Reference Frame Origin

UMAA does not specify a required origin for the vehicle coordinate reference frame. However, certain applications may benefit from defining a specific origin such as the registration of multiple sensors with associated offsets for data fusion.

Definitions

- Design Waterline (DWL) - The line representing the waterline on the vehicle at designed load in summer temperature seawater.
- Centerline - The vertical plane passing fore and aft down the center of the ship.
- Aft Perpendicular (AP) - The vertical line passing through the rudder stock.
- Forward Perpendicular (FP) - The vertical line through the intersection of the forward side of the stern with the Design Waterline.
- Amidships - The midpoint between the Forward and Aft Perpendiculars.

Common practice puts the origin at the intersection of the Design Waterline, Centerline, and Amidships (Figure 9).

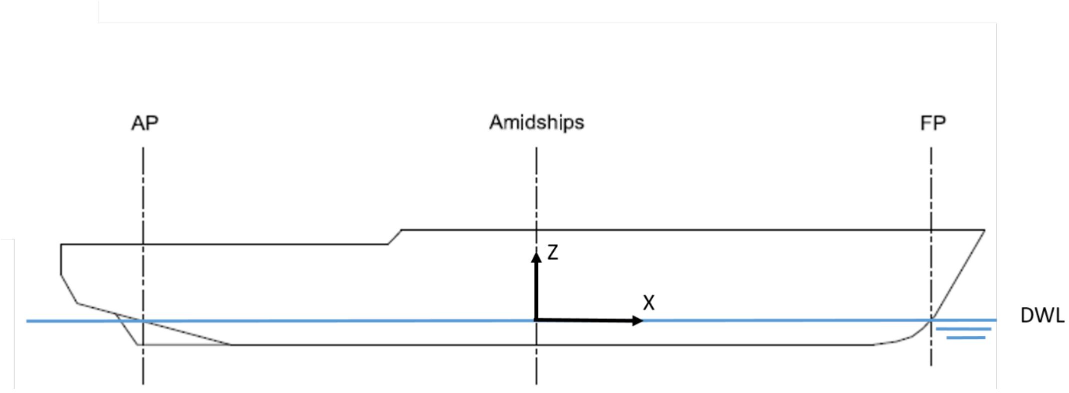


Figure 9: Origin location on a USV as example

For UUVs, common practice puts the origin as in Figure 10:

- X - at the Longitudinal Center of Buoyancy (LCB) when fully submerged
- Y - at the symmetrical centerline
- Z - at the Vertical Center of Buoyancy (VCB) when fully submerged

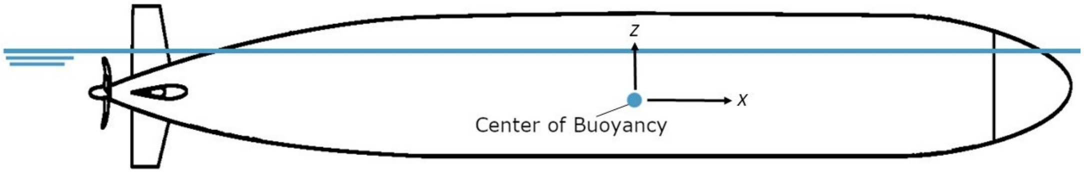


Figure 10: Origin location on a UUV as example

5 Flow Control

5.1 Command / Response

This section defines the flow of control for command/response over the DDS bus. A command/response is used to control a specific service. While the exact names and processes will depend on the specific service and command being executed, all command/responses in UMAA follow a similar pattern. A notional "Function" command `FunctionCommand` is used in the following examples. As will be described in subsequent paragraphs, DDS publish/subscribe methods are used in implementations to issue commands and responses.

To direct a `FunctionCommand` at a specific Service Provider, UMAA includes a `destination` GUID in all commands. A Service Provider is required to respond to all `FunctionCommand`s where the `destination` is the same as the Service Provider's ID. The Service Consumer will also create a unique `sessionID` for the command when commanded. The `sessionID` is used to track the command execution as a key into other command-related messages. Service Provider and Service Consumer terminology in the following sections is adopted from the OMG Service-oriented architecture Modeling Language (SoaML).

To initialize, a Service Provider (controllable resource) subscribes to the `FunctionCommand` DDS topic. At startup or right before issuing a command, the Service Consumer (controlling resource) subscribes to the `FunctionCommandStatus` DDS topic. Optionally, the Service Consumer may also subscribe to the `FunctionCommandAckReport` to monitor which command is currently being executed, and the `FunctionExecutionStatusReport`, if defined for the Function service, that provides reporting on function-specific data status.

Both Service Providers and Service Consumers are required to recover or clean up any previous persisted commands on the bus during initialization.

To execute a command the Service Consumer publishes a `FunctionCommandType` to the DDS bus. The Service Provider will be notified and will begin processing the request. During each phase of processing, the Service Provider will provide updates to the Service Consumer via published updates to a related `FunctionCommandStatus` topic. Command responses are correlated to their originating command via the `sessionID`. Command status updates are provided in the command responses via the `commandStatus` field with additional details included in the `commandStatusReason` field. The Service Provider will also publish the current executing command to the `FunctionCommandAckReport` topic. When defined for the Function service, the Service Provider must also publish the `FunctionExecutionStatusReport` topic and update it as appropriate throughout the execution of the command.

The required state transitions for the `commandStatus` field are shown in Figure 11. Every command must transition through the states as defined. For example, it is a violation to transition from `ISSUED` to `EXECUTING` without transitioning through `COMMANDED`. Even in the case where there is no logic executing between the `ISSUED` and `EXECUTING` states the Service Provider is required to transition through `COMMANDED`. This ensures consistent behavior across different Service Providers, including those that do require the `COMMANDED` state.

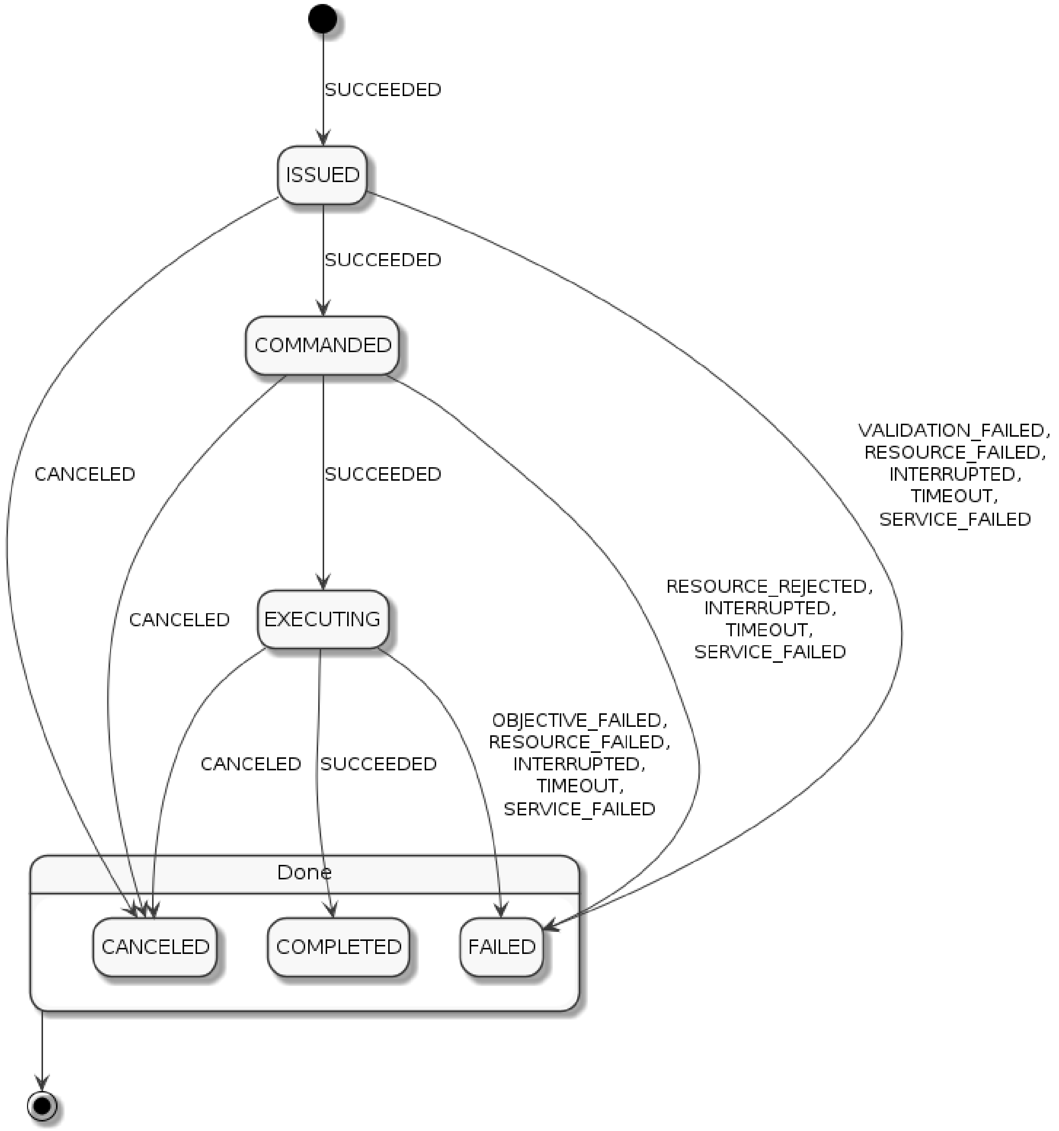


Figure 11: The state transitions of the `commandStatus` as commands are processed. Labels on the arrows represent valid `commandStatusReason` values for each transition.

In the following sections, the sequence diagrams demonstrate different exchanges between a Service Consumer and Service Provider. Within the diagrams, the dashed arrows represent implementation-specific communications that are outside of UMAA's scope. These sequence diagrams are just an example of one possible implementation. Other implementations may have different communication patterns between the Service Provider and the Resource or be implemented completely within the Service Provider process itself (no dependency on an external Resource). Likewise, the interactions between the User and Service Consumer may follow similiar or different patterns. However, the UMAA-defined exchanges with the DDS bus between the Service Consumer and Service Provider must happen in the order shown within the sequence diagrams.

5.1.1 High-Level Flow

The high-level flow of a command sequence is shown in Figure 12 and can be described as follows:

1. The Command Startup Sequence is performed
2. For each command to be executed
   (a) The Command Start Sequence is performed
   (b) The command is executed (sequence depends on the execution path, i.e., success, failure, or cancel)
   (c) The Command Cleanup Sequence is performed

3. The Command Shutdown Sequence is performed

The `ref` blocks will be defined in later sequence diagrams. Note that the duration of the system execution for any particular `FunctionCommandType` is defined by the combination of the Service Provider(s) and Service Consumer(s) in the system and my not be identical to the overall system execution duration. For example, providers may only be available to execute certain commands during specific phases of a misison or when certain hardware is in specific configurations. This Command Startup Sequence is not required to happend during a system startup phase. The only requirement is it must be completed by at least one Service Provider and one Service Consumer before any `FunctionCommandType` commands can be fully executed. Likewise, the Command Shutdown sequence may occur at anytime the `FunctionCommandType` will no longer be supported. There is no requirement the Command Shutdown Sequence only be performed during a system shutdown phase.

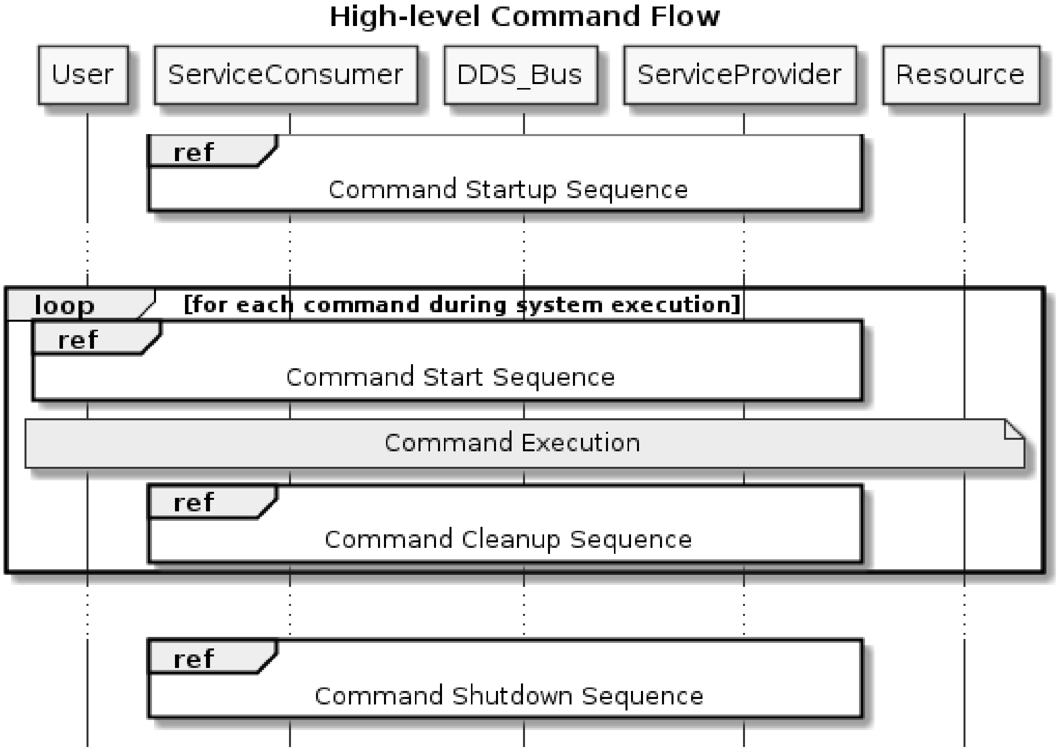


Figure 12: The sequence diagram for the high-level description of a command exeuction.

5.1.2 Command Startup Sequence

As part of initialization both the Service Provider and Service Consumer are required to perform a startup sequence. This startup prepares the Service Provider to execute commands and the Service Consumer to request commands and monitor the progress of those requested commands.

The Service Provider and Service Consumer can initialize in any order. Commands will not be completely executed until both have completed their initialization. The sequence diagram is shown in Figure 13.

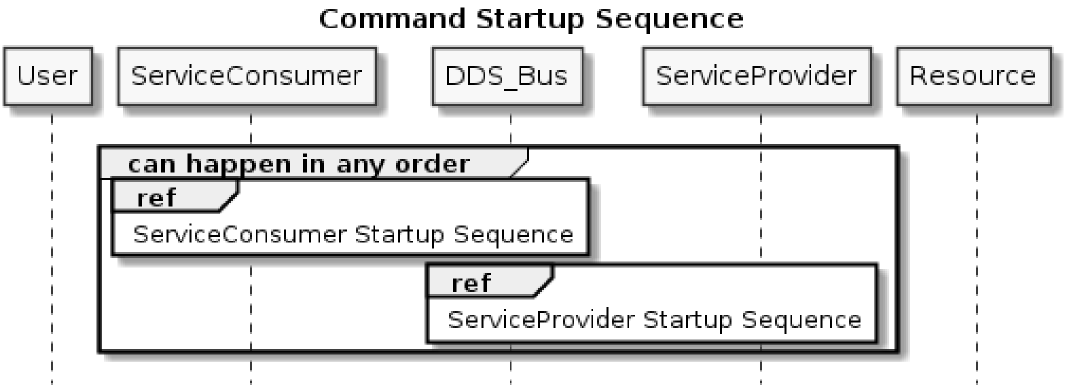


Figure 13: The sequence diagram for command startup.

5.1.2.1 Service Provider Startup Sequence During startup the Service Provider is required to register as a publisher to the `FunctionCommandStatus`, `FunctionCommandAckReport`, and, if defined for the Function service, the `FunctionExecutionStatus` topics.

The Service Provider is also required to subscribe to the `FunctionCommand` topic to be notified when new commands are published.

Finally, the Service Provider is required to handle any existing `FunctionCommandType` commands persisted on the DDS bus with the Service Provider's ID. For each command, if the Service Provider can and wishes to recover, it can continue to execute the command. To obtain the last published state of the command, the Service Provider must subscribe to the `FunctionCommandStatusType`. The Service Provider will continue following the normal status update sequence, picking up from the last status on the bus. If the Service Provider cannot or choses not to continue processing the command, it must fail the command by publishing a `FunctionCommandStatus` with a `commandStatus` of `FAILED` and a `reason` of `SERVICE_FAILED`.

The Service Provider Startup sequence is shown in Figure 14.

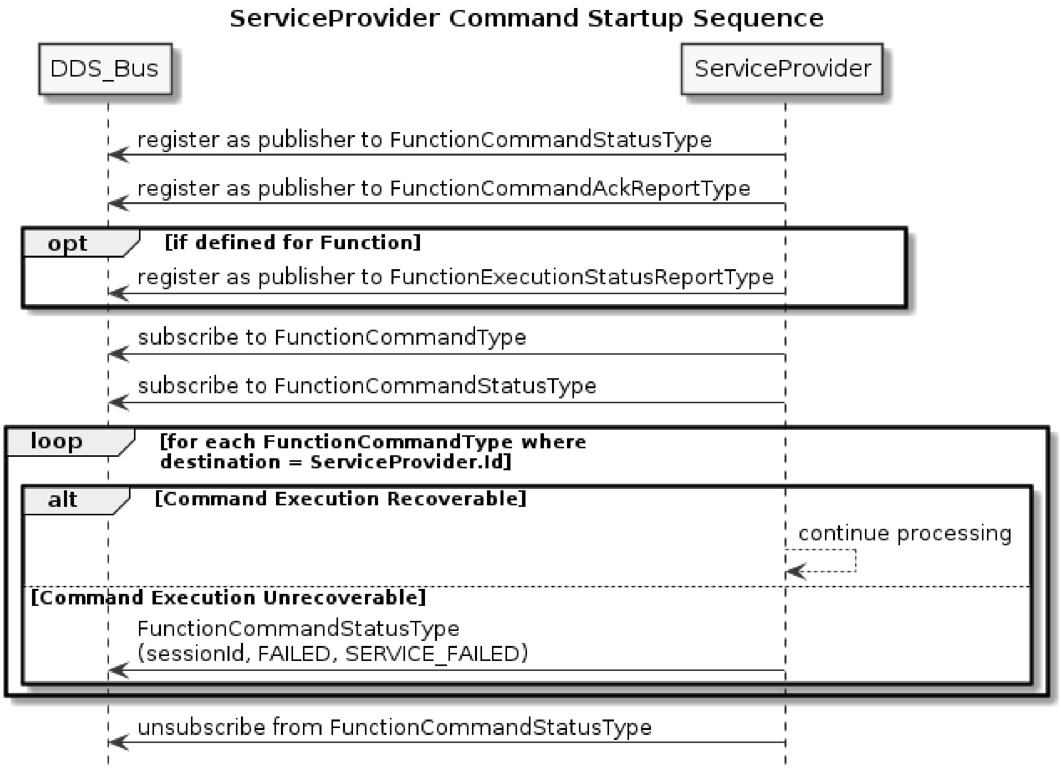

ServiceProvider Command Startup Sequence
DDS_Bus
ServiceProvider
register as publisher to FunctionCommandStatusType
register as publisher to FunctionCommandAckReportType
opt
[if defined for Function]
register as publisher to FunctionExecutionStatusReportType
subscribe to FunctionCommandType
subscribe to FunctionCommandStatusType
loop
[for each FunctionCommandType where destination = ServiceProvider.Id]
alt
[Command Execution Recoverable]
continue processing
[Command Execution Unrecoverable]
FunctionCommandStatusType (sessionId, FAILED, SERVICE_FAILED)
unsubscribe from FunctionCommandStatusType

Figure 14: The sequence diagram for command startup for Service Providers.

5.1.2.2 Service Consumer Startup Sequence

During startup the Service Consumer is required to register as a publisher of the `FunctionCommandType`.

The Service Consumer is also required to subscribe to the `FunctionCommandStatusType` to monitor the execution of any published commands. The Service Consumer can optionally register for the `FunctionCommandAckReportType` and, if defined for the Function service, the `FunctionExecutionStatusReportType` if it desires to track additional status of the execution of commands.

Finally, the Service Consumer is required to handle any existing `FunctionCommandType` commands persisted on the DDS bus with this Service Consumer's ID. To find existing `FunctionCommandType`s on the bus, it must first subscribe to the topic. If the Service Consumer can and wishes to recover, it can continue to monitor the execution of the command. If the Service Consumer cannot or choses not to continue the execution of the command, it must cancel the command via the normal command cancel method.

The Service Consumer Startup sequence is shown in Figure 15.

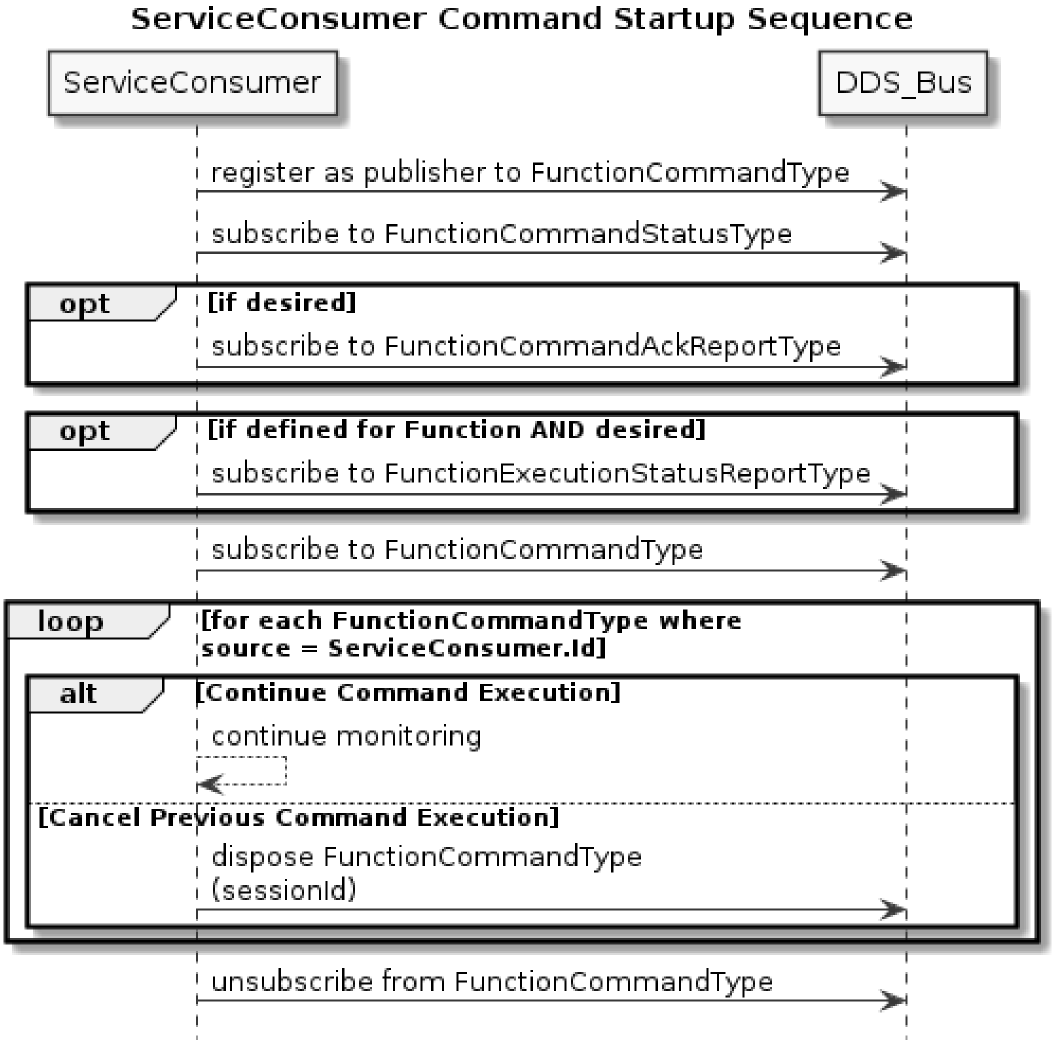

ServiceConsumer Command Startup Sequence
ServiceConsumer
DDS_Bus
register as publisher to FunctionCommandType
subscribe to FunctionCommandStatusType
opt
[if desired]
subscribe to FunctionCommandAckReportType
opt
[if defined for Function AND desired]
subscribe to FunctionExecutionStatusReportType
subscribe to FunctionCommandType
loop
[for each FunctionCommandType where source = ServiceConsumer.Id]
alt
[Continue Command Execution]
continue monitoring
[Cancel Previous Command Execution]
dispose FunctionCommandType (sessionId)
unsubscribe from FunctionCommandType

Figure 15: The sequence diagram for command startup for Service Consumers.

5.1.3 Command Execution Sequences

Once both the Service Provider and Service Consumer have performed the startup sequence, the system is ready be begin issuing and executing commands.

5.1.4 Command Start Sequence

The initial start sequence to execute a single command follows this pattern:

1. The User of the Service Consumer issues a request for a command to be executed.
2. The Service Consumer publishes the `FunctionCommandType` with a unique session ID, the source ID of the Service Consumer and the destination ID of the desired Service Provider.
3. The Service Provider, upon notification of the new `FunctionCommandType`, publishes a new `FunctionCommandStatusType` with the same session ID as the new `FunctionCommandType` and the status of `ISSUED` and reason of `SUCCEEDED` to notify the Service Consumer it has received the new command.

The Command Start Sequence is shown in Figure 16. This pattern will be repeated each time a new command is requested. After the Command Start Sequence, the sequence can take different paths depending on the actual execution of the command. Some possible paths are detailed in the following sections, but they do not enumerate all of the possible execution paths. Other paths (e.g., an objective failing) will follow a similiar pattern to other failures; all are required to follow the state diagram shown in Figure 11 and eventually end with the Command Cleanup Sequence (as shown in Figure 22).

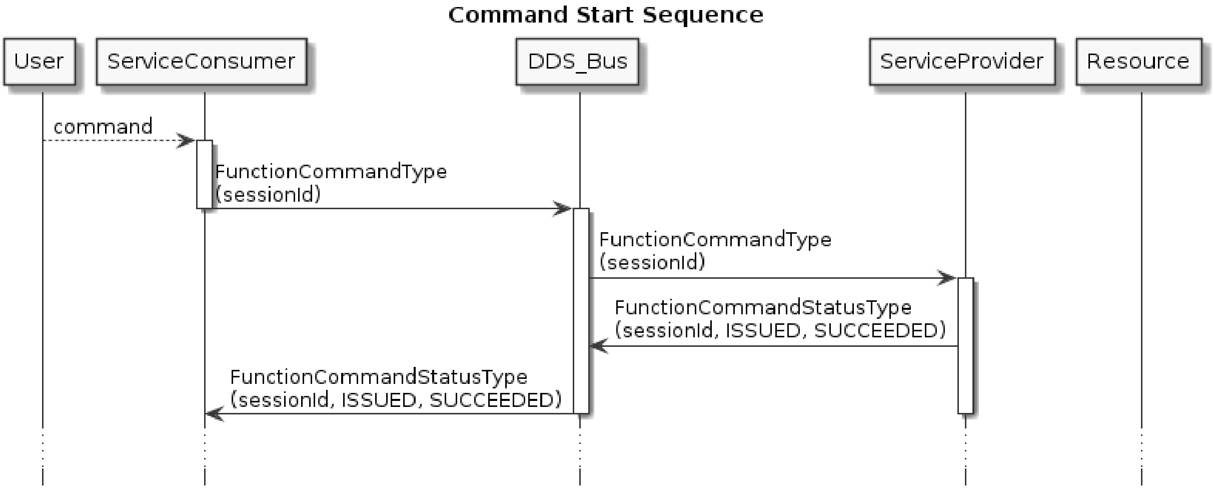


Figure 16: The sequence diagram for the start of a command execution.

5.1.4.1 Command Execution Once a Service Provider starts to process a command, the Command Execution sequence is:

1. The Service Provider publishes a `FunctionCommandAckReportType` with matching session ID and parameters as the `FunctionCommandType` it is starting to process.
2. The Service Provider performs any validation and negotiation with backing resources as necessary. Once the command is ready to be executed the Service Provider publishes a `FunctionCommandStatusType` with a status `COMMANDED` and reason `SUCCEEDED` to notify the Service Consumer that the command has been validated and commanded to start execution.
3. Once the command has begun executing the Service Provider publishes a `FunctionCommandStatusType` with a status `EXECUTED` and reason `SUCCEEDED` to notify the Service Consumer that the command has been validated and commanded to start.
4. If the Function has a defined `FunctionExecutionStatusReportType`, the Service Provider must publish a new instance with matching session ID as the associated `FunctionCommandType`. The `FunctionExecutionStatusReportType` must be updated by the Service Provider throughout the execution as dictated by the defintions of the command-specific attributes in the execution status report.

The command execution sequence is shown in Figure 17. This sequence holds until the command completes execution.

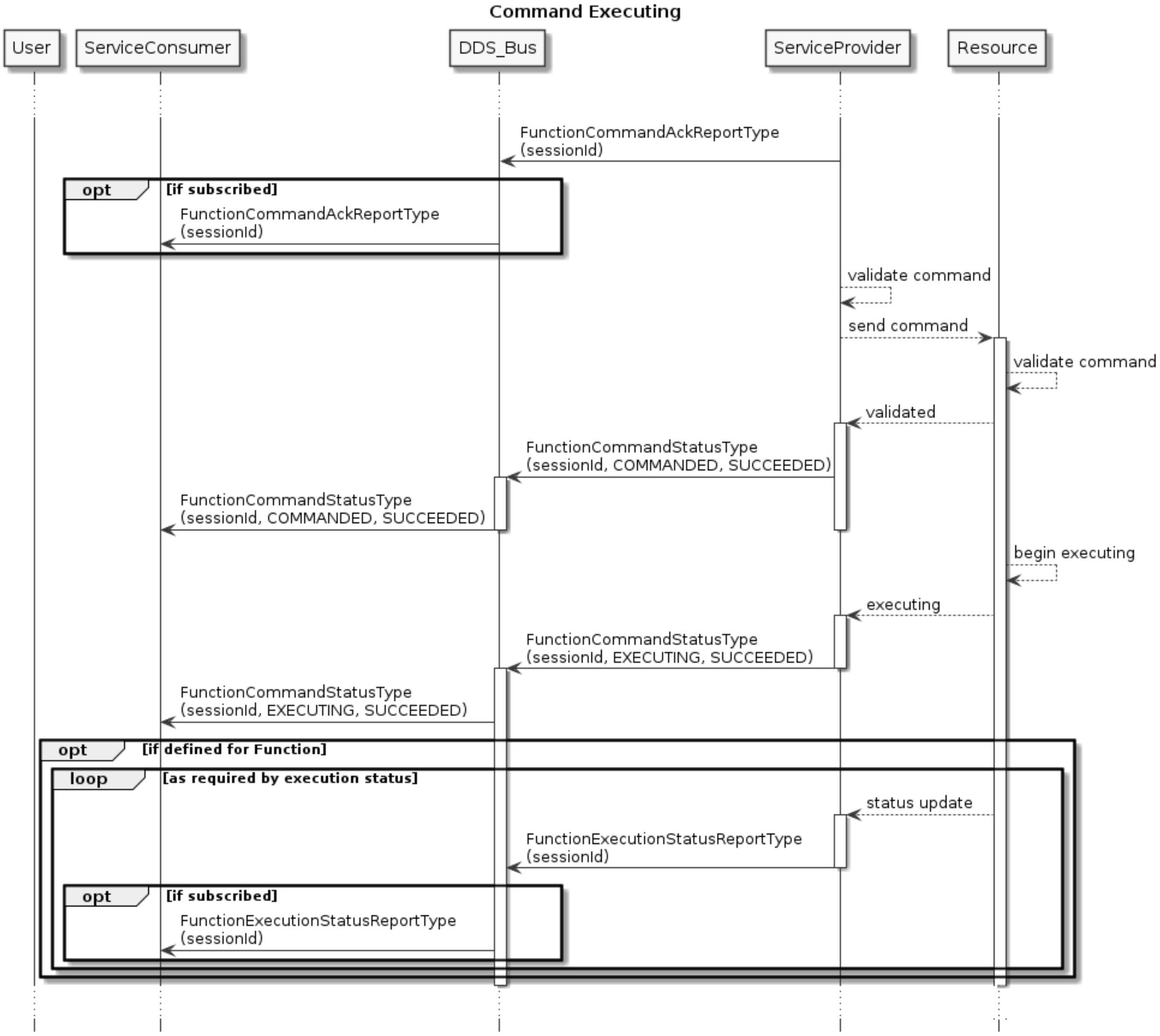

Command Executing
User
ServiceConsumer
DDS_Bus
ServiceProvider
Resource
FunctionCommandAckReportType
(sessionId)
opt
[if subscribed]
FunctionCommandAckReportType
(sessionId)
validate command
send command
validate command
validated
FunctionCommandStatusType
(sessionId, COMMANDED, SUCCEEDED)
FunctionCommandStatusType
(sessionId, COMMANDED, SUCCEEDED)
begin executing
executing
FunctionCommandStatusType
(sessionId, EXECUTING, SUCCEEDED)
FunctionCommandStatusType
(sessionId, EXECUTING, SUCCEEDED)
opt
[if defined for Function]
loop
[as required by execution status]
status update
FunctionExecutionStatusReportType
(sessionId)
opt
[if subscribed]
FunctionExecutionStatusReportType
(sessionId)

Figure 17: The beginning sequence diagram for a command execution.

The normal successful conclusion of a command being executed in some cases is initiated by the Service Consumer (an endless GlobalVector command concluded by canceling it) and in other cases is initiated by the Service Provider (a GlobalWaypoint commanded concluded by reaching the last waypoint). Unless otherwise explicitly stated, it is assumed the Service Provider will be able to identify the successful conclusion of a command. In the cases where commands are defined to be indeterminate the Service Consumer must cancel the command when the Service Consumer no longer desires the command to be executed.

5.1.4.2 Command Execution Success When the Service Provider determines a command has succesfully completed, it must update the associated `FunctionCommandStatusType` with as status of `COMPLETED` and reason of `SUCCEEDED`. This signals to the Service Consumer the command has completed successfully.

The Command Execution Success sequence is shown in Figure 18.

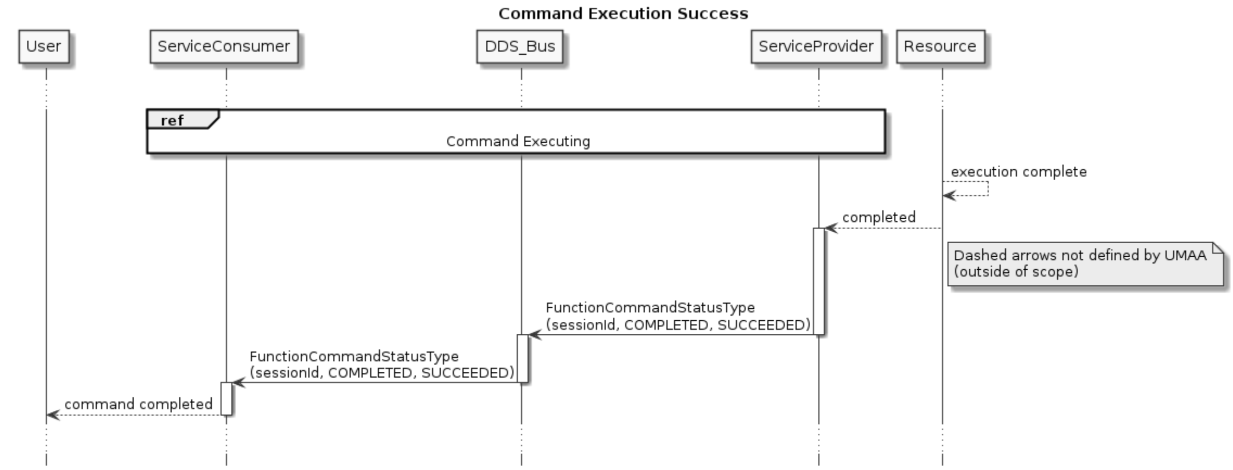

Command Execution Success
User
ServiceConsumer
DDS_Bus
ServiceProvider
Resource
ref
Command Executing
execution complete
completed
Dashed arrows not defined by UMAA
(outside of scope)
FunctionCommandStatusType
(sessionId, COMPLETED, SUCCEEDED)
FunctionCommandStatusType
(sessionId, COMPLETED, SUCCEEDED)
command completed

Figure 18: The sequence diagram for a command that completes successfully.

5.1.4.3 Command Execution Failure The command may fail to complete for any number of reasons including software errors, hardware failures, or unfavorable environmental conditions. The Service Provider may also reject a command for a number of reasons including inability to perform the task, malformed or out of range requests, or a command being interrupted by a higher priority process. In all cases the Service Provider must publish a `FunctionCommandStatusType` with an identical `sessionID` as the originating `FunctionCommandType` with a status of `FAILED` and the reason that reflects the cause of the failure (`VALIDATION_FAILED, SERVICE_FAILED, OBJECTIVE_FAILED,` etc).

The following figures provide examples of cases where a command has failed.

In the first example, the backing Resource has failed and the Service Provider in unable to communicate with it. In this case the Service Provider will report a `FunctionCommandStatusType` with a status of `FAILED` and a reason of `RESOURCE_FAILED`. This is shown in Figure 19.

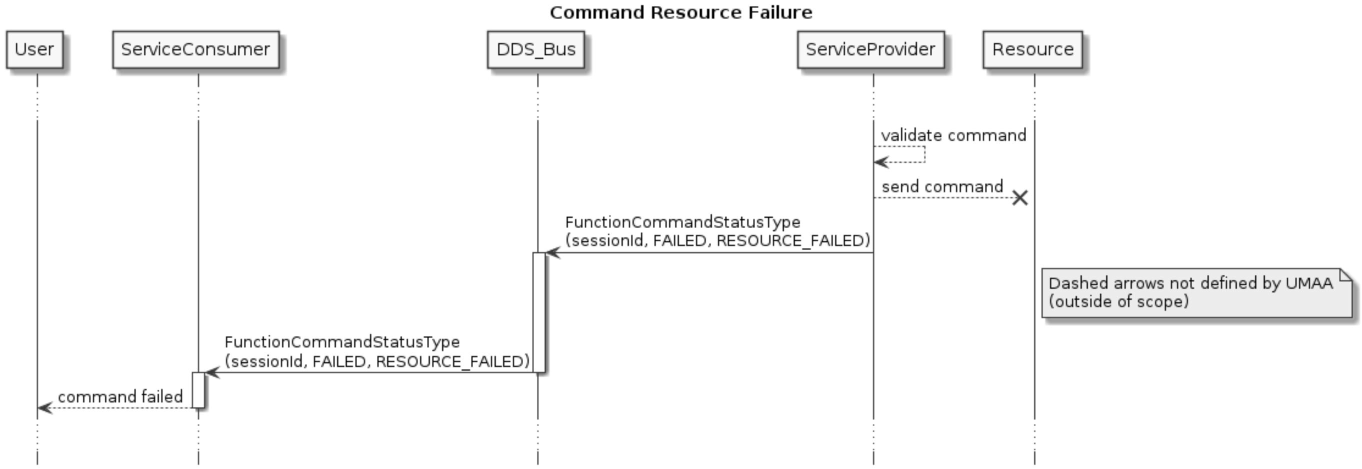

Command Resource Failure
User
ServiceConsumer
DDS_Bus
ServiceProvider
Resource
validate command
send command
FunctionCommandStatusType
(sessionId, FAILED, RESOURCE_FAILED)
Dashed arrows not defined by UMAA
(outside of scope)
FunctionCommandStatusType
(sessionId, FAILED, RESOURCE_FAILED)
command failed

Figure 19: The sequence diagram for a command that fails due to Resource failure.

In the second example, the Resource takes too long to response, so the Service Provider cancels the request and reports a `FunctionCommandStatusType` with a status of `FAILED` and a reason of `TIMEOUT`. This is shown in Figure 20.

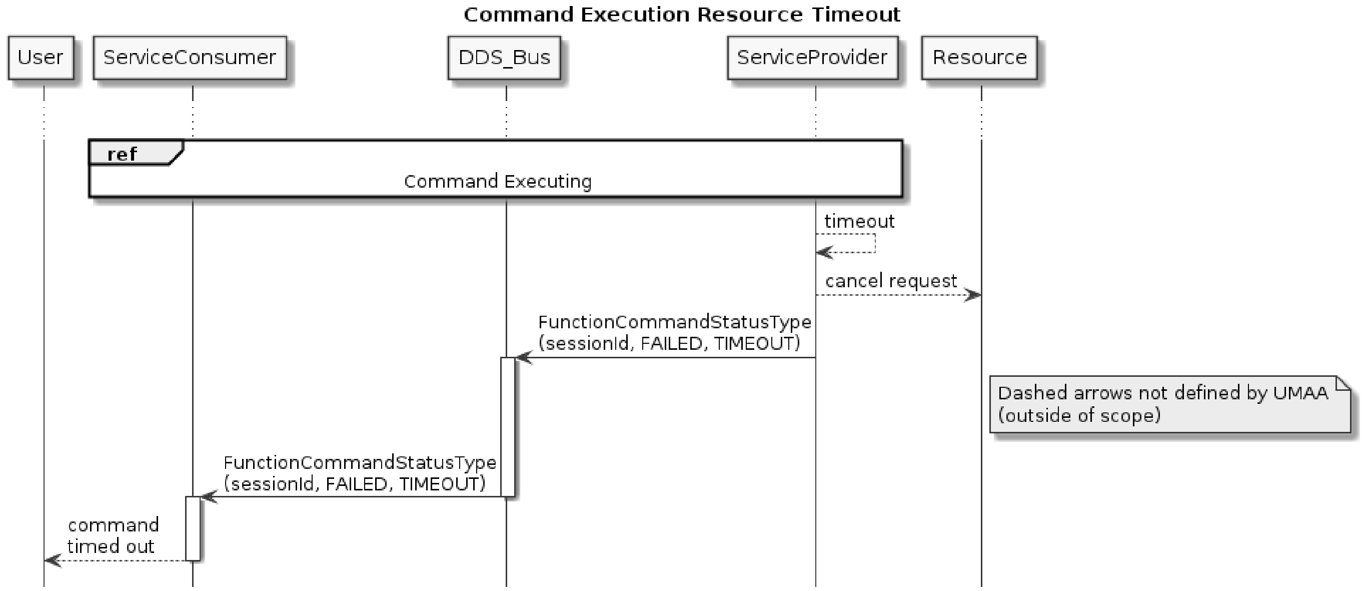
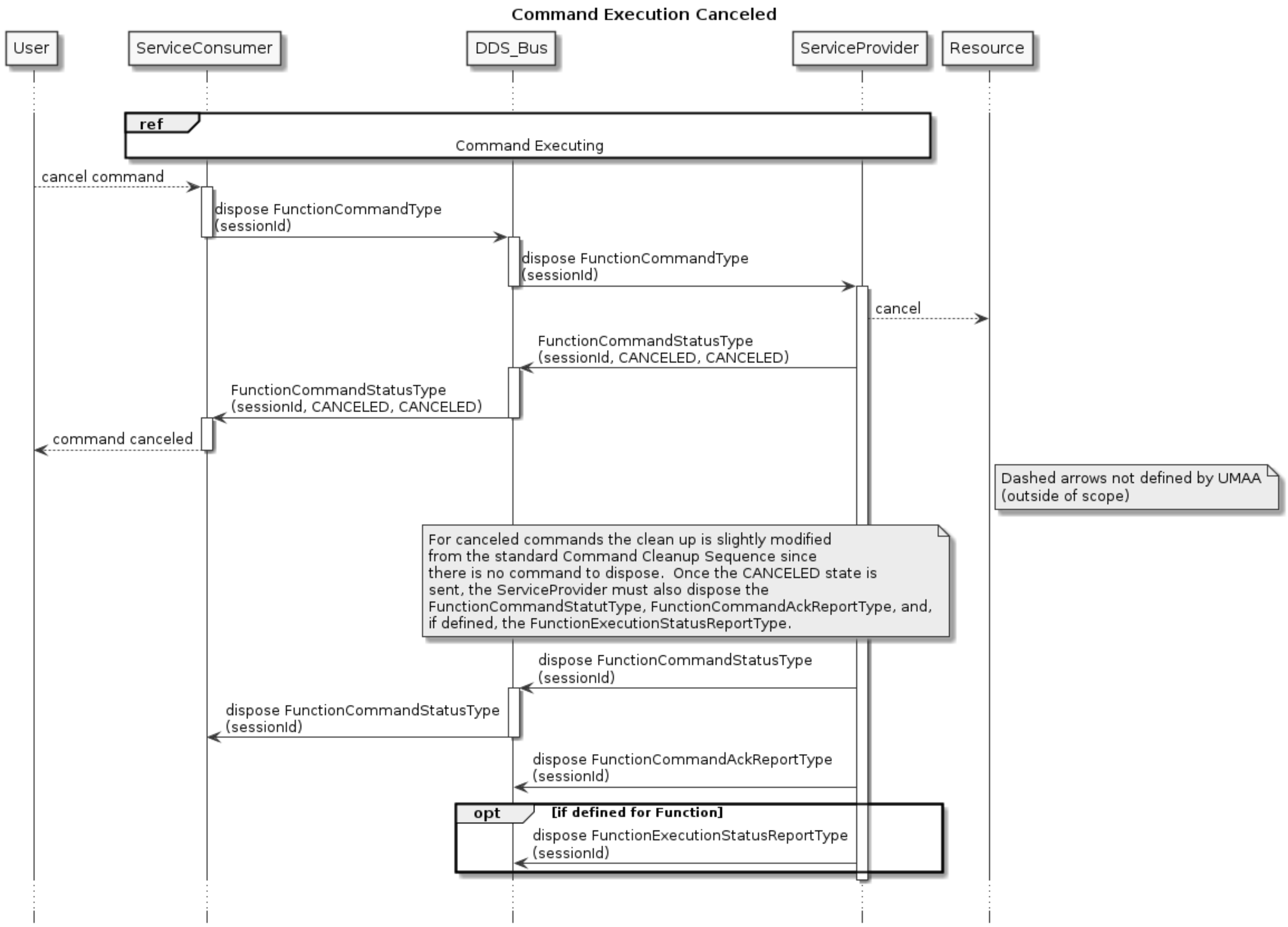

Command Execution Resource Timeout
User
ServiceConsumer
DDS_Bus
ServiceProvider
Resource
ref
Command Executing
timeout
cancel request
FunctionCommandStatusType
(sessionId, FAILED, TIMEOUT)
Dashed arrows not defined by UMAA
(outside of scope)
FunctionCommandStatusType
(sessionId, FAILED, TIMEOUT)
command
timed out

Figure 20: The sequence diagram for a command that times out before completing.

Other failure conditions will follow a similar pattern: when the failure is recognized, the Service Provider will publish a `FunctionCommandStatusType` with a status of `FAILED` and a reason that reflect the cause of the failure.

5.1.4.4 Command Canceled The Service Consumer may decide to cancel the command before processing is finished. To signal a desire to cancel a command, the Service Consumer disposes the existing `FunctionCommandType` from the DDS bus before the execution is complete. When notified of the command disposal, if the Service Provider is able to cancel the command it should respond to the Service Consumer with a `FunctionCommandStatusType` with both the status and reason as `CANCELED` and then dispose the `FunctionCommandStatusType` and `FunctionCommandAckReportType` and, if defined for the Function service, the `FunctionExecutionStatusReportType` from the bus. This is shown in Figure 21. If the command cannot be canceled the Service Provider can continue to update the command status until the execution is completed, reporting `FunctionCommandStatusType` with a status of `COMPLETED` and a reason of `SUCCEEDED`, and then dispose the `FunctionCommandStatusType` and `FunctionCommandAckReportType` and, if defined for the Function service, the `FunctionExecutionStatusReportType` from the DDS bus.

There is no new unique specific status message response to a cancel command from the Service Provider. The cancel command status can be inferred through the corresponding `FunctionCommandStatusType` status and reason updates.


Command Execution Canceled
User
ServiceConsumer
DDS_Bus
ServiceProvider
Resource
ref
Command Executing
cancel command
dispose FunctionCommandType (sessionId)
dispose FunctionCommandType (sessionId)
cancel
FunctionCommandStatusType (sessionId, CANCELED, CANCELED)
FunctionCommandStatusType (sessionId, CANCELED, CANCELED)
command canceled
dispose FunctionCommandStatusType (sessionId)
dispose FunctionCommandStatusType (sessionId)
dispose FunctionCommandAckReportType (sessionId)
opt
[if defined for Function]
dispose FunctionExecutionStatusReportType (sessionId)

Dashed arrows not defined by UMAA (outside of scope)

For canceled commands the clean up is slightly modified from the standard Command Cleanup Sequence since there is no command to dispose. Once the CANCELED state is sent, the ServiceProvider must also dispose the FunctionCommandStatutType, FunctionCommandAckReportType, and, if defined, the FunctionExecutionStatusReportType.

Figure 21: The sequence diagram for a command that is canceled by the Service Consumer before the Service Provider is able to complete it.

5.1.5 Command Cleanup

The Service Consumer and Service Provider are responsible for disposing corresponding data published to the DDS bus when the command is no longer active. With the exception of a canceled command, the signal that a `FunctionCommandType` can be disposed is when the `FunctionCommandStatusType` reports a terminal state (`COMPLETED` or `FAILED`)[3]. In turn, the signal that a `FunctionCommandStatusType`, `FunctionCommandAckReportType`, and if defined for the Function service, the `FunctionExecutionStatusReportType` can be disposed is when the corresponding `FunctionCommandType` has been disposed. This is shown in Figure 22.

[3] While `CANCELED` is also a terminal state, `CANCELED` command cleanup is handled specially as part of the cancelling sequence and, as such, does not need to be handled here.

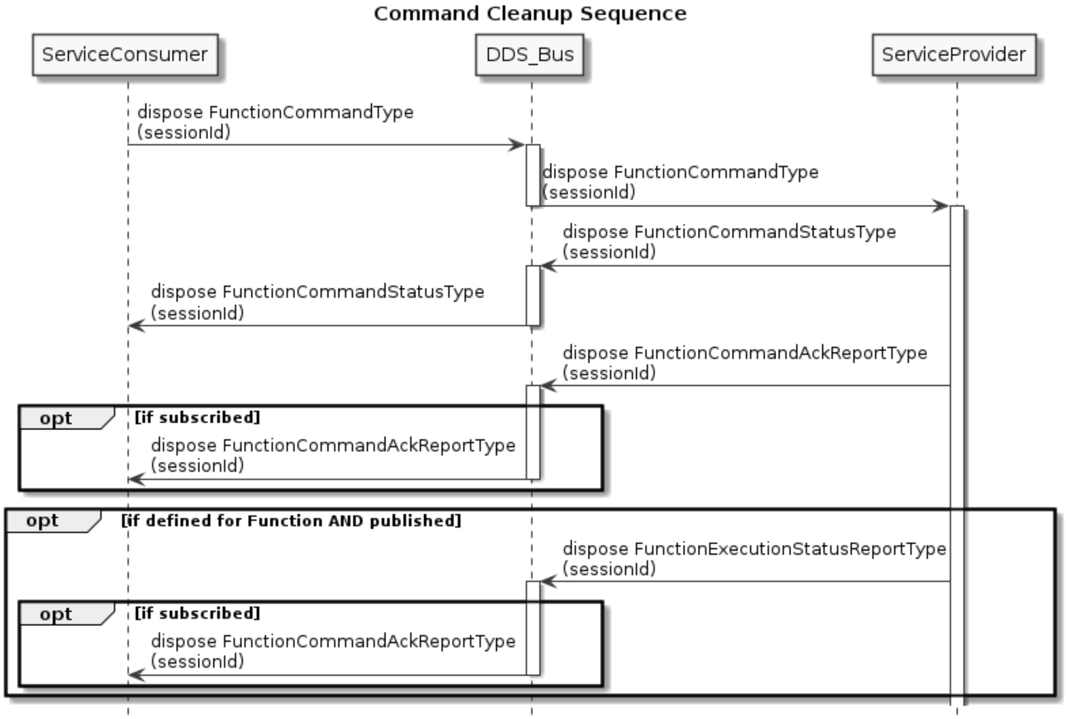

Command Cleanup Sequence
ServiceConsumer
DDS_Bus
ServiceProvider
dispose FunctionCommandType (sessionId)
dispose FunctionCommandType (sessionId)
dispose FunctionCommandStatusType (sessionId)
dispose FunctionCommandStatusType (sessionId)
dispose FunctionCommandAckReportType (sessionId)
opt [if subscribed]
dispose FunctionCommandAckReportType (sessionId)
opt [if defined for Function AND published]
dispose FunctionExecutionStatusReportType (sessionId)
opt [if subscribed]
dispose FunctionCommandAckReportType (sessionId)

Figure 22: The sequence diagram showing cleanup of the bus when a command has been completed and the Service Consumer no longer wishes to maintain the commanded state.

5.1.6 Command Shutdown Sequence

As part of shutdown both the Service Provider and Service Consumer are required to perform a shutdown sequence. This shutdown cleans up resources on the DDS bus and informs the system that the Service Provider and Service Consumer are no longer available.

The Service Provider and Service Consumer can shutdown in any order. The sequence diagram is shown in Figure 23.

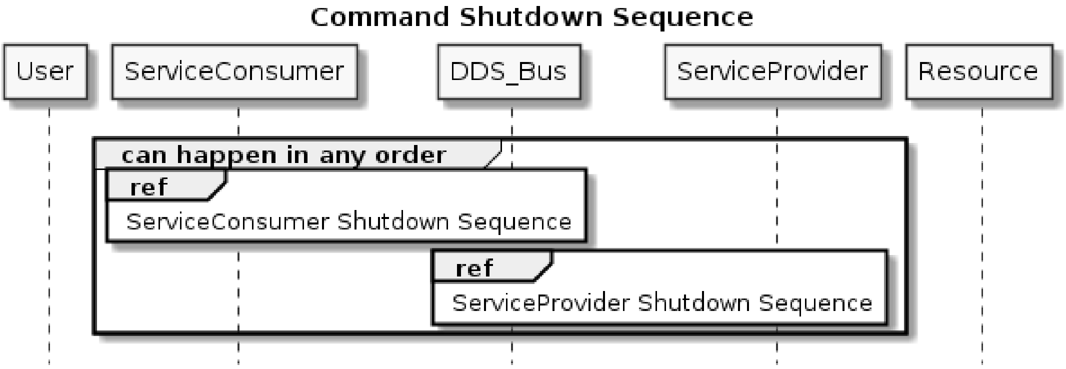

Command Shutdown Sequence
User
ServiceConsumer
DDS_Bus
ServiceProvider
Resource
can happen in any order
ref
ServiceConsumer Shutdown Sequence
ref
ServiceProvider Shutdown Sequence

Figure 23: The sequence diagram for command shutdown.

5.1.6.1 Service Provider Shutdown Sequence During shutdown the Service Provider is required to fail any incomplete requests and then unregisters as a publisher of the `FunctionCommandStatusType`, `FunctionCommandAckReportType`, and, if defined for the Function service, the `FunctionExecutionStatusReportType`.

The Service Provider is also required to unsubscribe from the `FunctionCommandType`.

The Service Provider Shutdown sequence is shown in Figure 24.

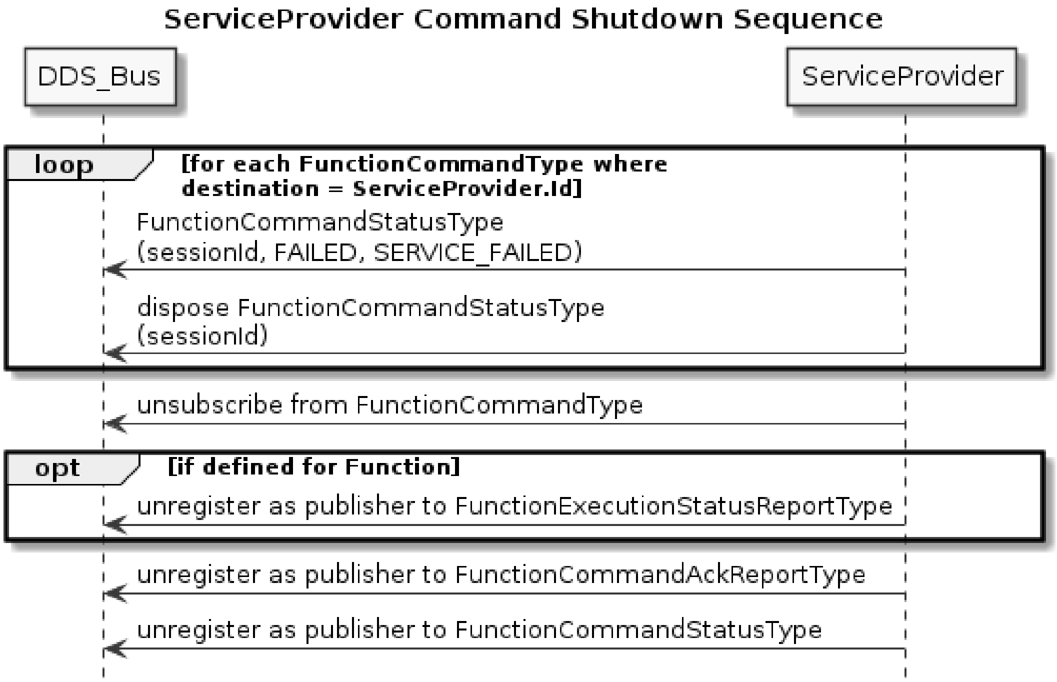

ServiceProvider Command Shutdown Sequence
DDS_Bus
ServiceProvider
loop
[for each FunctionCommandType where destination = ServiceProvider.Id]
FunctionCommandStatusType (sessionId, FAILED, SERVICE_FAILED)
dispose FunctionCommandStatusType (sessionId)
unsubscribe from FunctionCommandType
opt
[if defined for Function]
unregister as publisher to FunctionExecutionStatusReportType
unregister as publisher to FunctionCommandAckReportType
unregister as publisher to FunctionCommandStatusType

Figure 24: The sequence diagram for command shutdown for Service Providers.

5.1.6.2 Service Consumer Shutdown Sequence

During shutdown the Service Consumer is required to cancel any incomplete requests and then unregister as a publisher of the `FunctionCommandType`.

The Service Consumer is also required to unsubscribe from the `FunctionCommandStatusType`, the `FunctionCommandAckReportType` if subscribed, and the `FunctionExecutionStatusReportType` if defined for the Function service and subscribed.

The Service Consumer Shutdown sequence is shown in Figure 25.

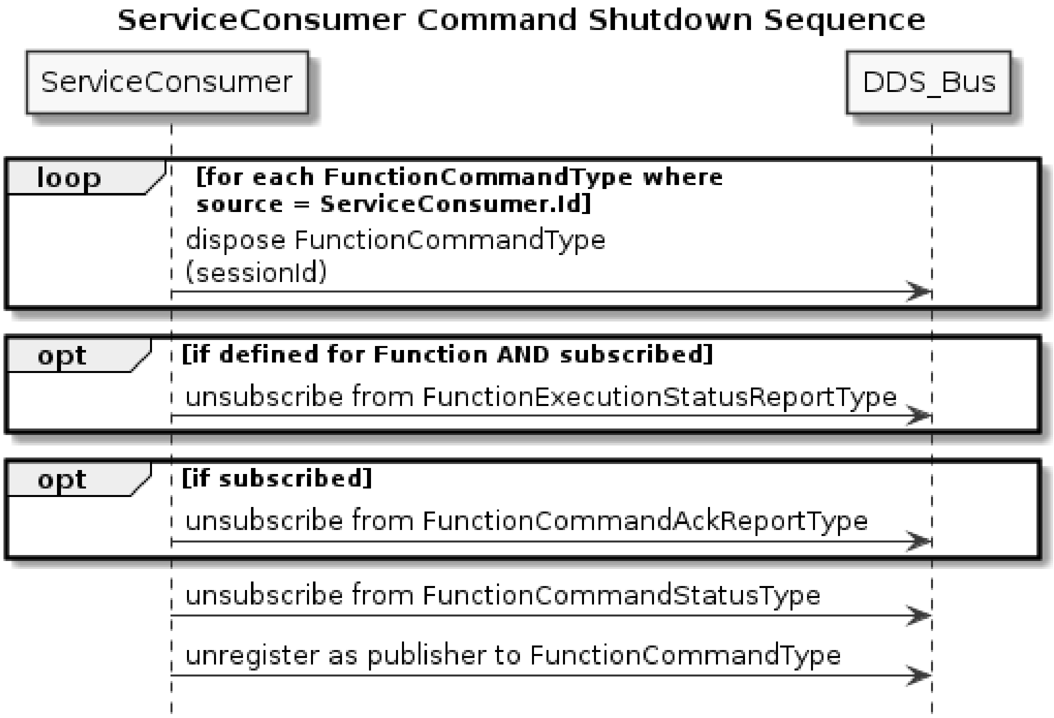

ServiceConsumer Command Shutdown Sequence
ServiceConsumer
DDS_Bus
loop
[for each FunctionCommandType where source = ServiceConsumer.Id]
dispose FunctionCommandType (sessionId)
opt
[if defined for Function AND subscribed]
unsubscribe from FunctionExecutionStatusReportType
opt
[if subscribed]
unsubscribe from FunctionCommandAckReportType
unsubscribe from FunctionCommandStatusType
unregister as publisher to FunctionCommandType

Figure 25: The sequence diagram for command shutdown for Service Consumers.

5.2 Request / Reply

This section defines the flow of control for request/reply over the DDS bus. A request/reply is used to obtain data or status from a specific Service Provider.

A Service Provider is required to reply to all requests it receives. In the case of requests with no query data, this is accomplished via a DDS subscribe. In the case of a request with associated query data, a message with the query data must be published by the requester. To direct a request at a specific Service Provider or set of services UMAA defines a `destination` GUID as part of requests.

In the following sections, the sequence diagrams demonstrate different exchanges between a Service Consumer and Service Provider. Within the diagrams, the dashed arrows represent implementation-specific communications that are outside of UMAA's scope. Additionally, these sequence diagrams are just an example of one possible implementation. Other implementations may have different communication patterns between the Service Provider and the Resource or be implemented completely within the Service Provider process itself (no external Resource). In all implementations, however, UMAA-defined exchanges with the DDS bus between the Service Consumer and Service Provider must happen in the order shown within the sequence diagrams.

5.2.1 Request/Reply without Query Data

In the case where there is no specific query data (i.e., the service is always just providing the current data to the bus) the sequence of exchanges is show in Figure 26.

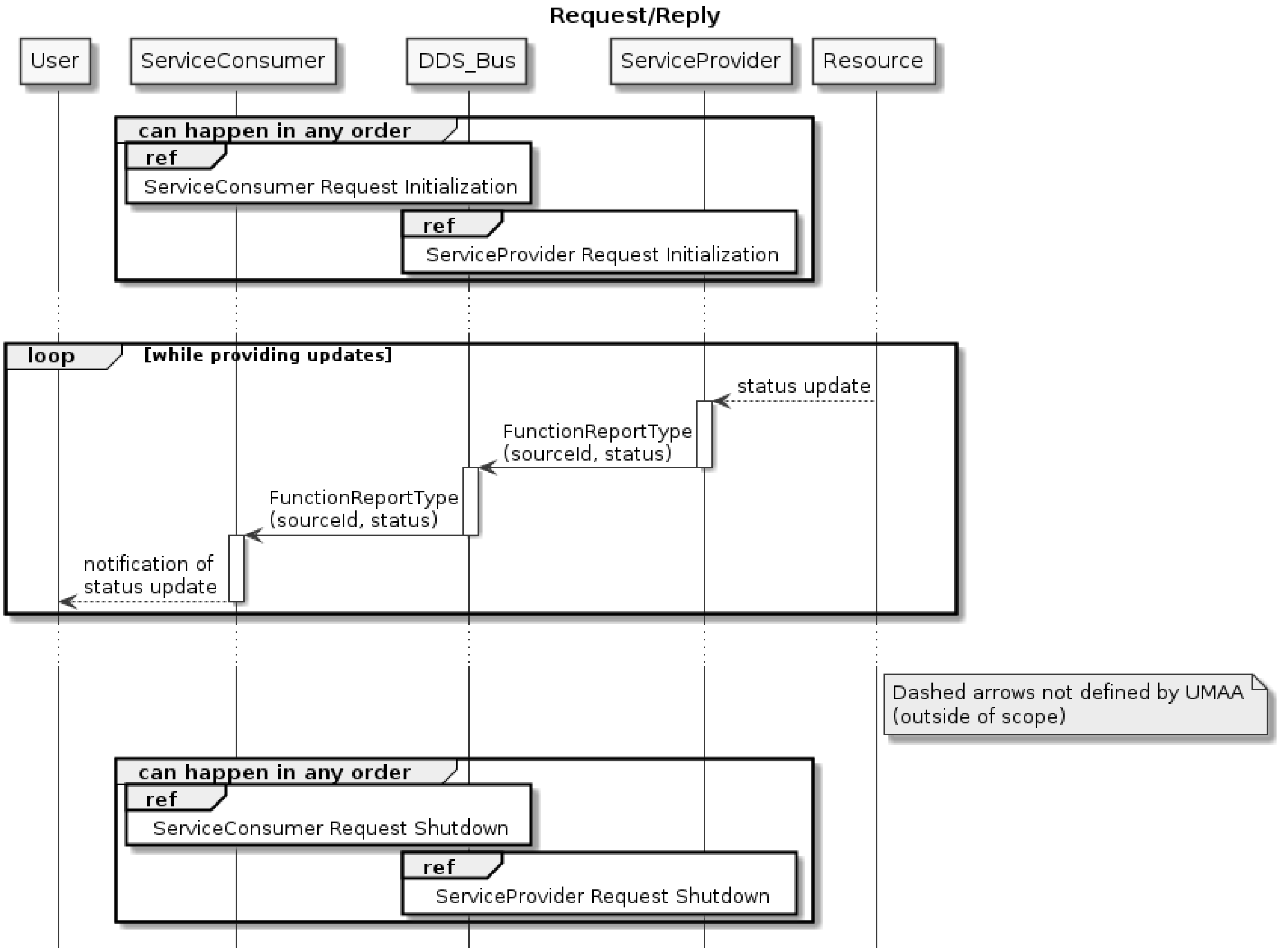


Figure 26: The sequence diagram for a request/reply for report data that does not require any specific query data.

5.2.1.1 Service Provider Startup Sequence

The Service Provider registers as a publisher of `FunctionReportType`s to be able to respond to requests. The Service Provider must also handle reports that exist on the bus from a previous instantiation, either by providing an immediate update or, if the status is unrecoverable, disposing of the old `FunctionReportType`. This is shown in Figure 27.

As `FunctionReportType` updates are required (either through event-driven changes or periodic updates), the Service Provider publishes the updated data. The DDS bus will deliver the updates to the Service Consumer.

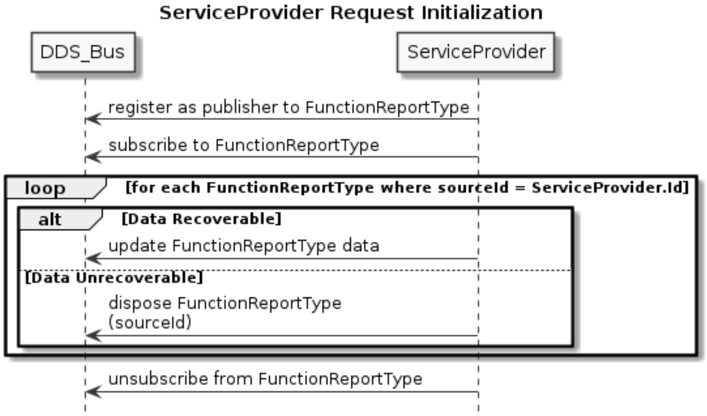

ServiceProvider Request Initialization
DDS_Bus
ServiceProvider
register as publisher to FunctionReportType
subscribe to FunctionReportType
loop
[for each FunctionReportType where sourceId = ServiceProvider.Id]
alt
[Data Recoverable]
update FunctionReportType data
[Data Unrecoverable]
dispose FunctionReportType (sourceId)
unsubscribe from FunctionReportType

Figure 27: The sequence diagram for initialization of a Service Provider to provide FunctionReportTypes.

5.2.1.2 Service Consumer Startup Sequence

The Service Consumer subscribes to the `FunctionReportType` to signal an outstanding request for updates. This is shown in Figure 28.

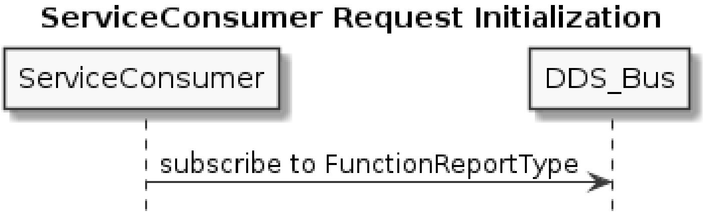

ServiceConsumer Request Initialization
ServiceConsumer
DDS_Bus
subscribe to FunctionReportType

Figure 28: The sequence diagram for initialization of a Service Consumer to request FunctionReportTypes.

5.2.1.3 Service Provider Shutdown

To no longer provide `FunctionReportType`s, the Service Provider disposes the `FunctionReportType` and unregisters as a publisher of the data as shown in Figure 29.

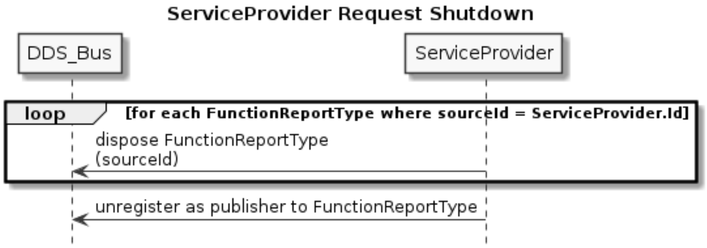

ServiceProvider Request Shutdown
DDS_Bus
ServiceProvider
loop
[for each FunctionReportType where sourceId = ServiceProvider.Id]
dispose FunctionReportType (sourceId)
unregister as publisher to FunctionReportType

Figure 29: The sequence diagram for shutdown of a Service Provider.

5.2.1.4 Service Consumer Shutdown

To no longer request `FunctionReportType`s, the Service Consumer unsubscribes from `FunctionReportType` as shown in Figure 30.

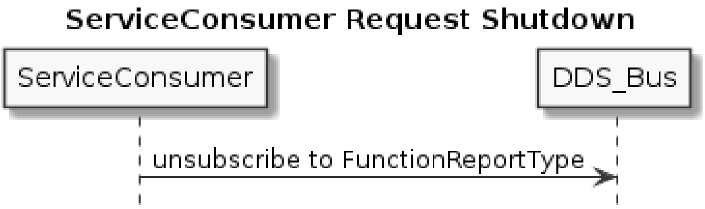

ServiceConsumer Request Shutdown
ServiceConsumer
DDS_Bus
unsubscribe to FunctionReportType

Figure 30: The sequence diagram for shutdown of a Service Consumer.

5.2.2 Request/Reply with Query Data

Currently UMAA does not define any request/reply interactions with query data, but it is expected some will be defined. When defined, this section will be expanded to describe how they must be used.

6 Experimental Services (EXP) Services and Interfaces

6.1 Services and Interfaces

The interfaces in the following subsections describe how each UCS-UMAA topic is defined by listing the name, namespace, and member attributes. The "name" corresponds with the message name of a given service interface. The "namespace" defines the scope of the "name" where similar commands are grouped together. The "member attributes" are fields that can be populated with differing data types, e.g. a generic "depth" attribute could be populated with a double data value. Note that using a UCS-UMAA "Topic Name" requires using the fully-qualified namespace plus the topic name.

Each interface topic is referenced by a UMAA service and is defined as either an input or output interface.
Attributes ending in one or more asterisk(s) denote the following:
* = Key (annotated with @key in IDL file, vendors may use different notation to indicate a key field)
†= Optional (annotated with @optional in IDL file, vendors may use different notation to indicate an optional field)

Optional fields should be handled as described in the UMAA Compliance Specification.

Commands issued on the DDS bus must be treated as if they are immutable in UMAA and therefore if updated (treated incorrectly as mutable), the resulting service actions are indeterminate and flow control protocols are no longer guaranteed.

Operations without DDS Topics

The following operations are all handled directly by DDS. They are marked in the operations tables with a ⊕.

query<...> - all query operations are used to retrieve the correlated report message. For UMAA, this operation is accomplished through subscribing to the appropriate DDS topic.

cancel<...> - all cancel operations are used to nullify the current command. For UMAA, this operation is accomplished through the DDS dispose action on the publisher.

report<...>CancelCommandStatus - all cancel reports are included here to show completeness of the MDE model mapping to UMAA. For UMAA, this operation is not used.

Instead, the cancel status is inferred from the associated command status. If the cancel command is successful, the corresponding command will fail with a command status and reason of CANCELED. If the corresponding command status reports COMPLETED, then this cancel command has failed.

6.1.1 AnalogVideoConfig

The purpose of this service is to provide access to the configuration of the analog camera, allowing the controlling component to set the camera to a particular operational profile.

Table 6: AnalogVideoConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryAnalogVideoConfig⊕	reportAnalogVideoConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.1.1 reportAnalogVideoConfig

Description: This operation is used to report the configuration of the analog video sensor.

Namespace: UMAA::SEM::AnalogVideoConfig

Topic: AnalogVideoConfigReport

Data Type: AnalogVideoConfigReportType

Table 7: AnalogVideoConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
analogVideoError	AnalogSensorErrorType	The error code associated with the analog video sensor
videoFormat	VideoFormatEnumType	Current video format configuration

6.1.2 AnalogVideoSpecs

The purpose of this service is to provide access to the capabilities of the analog camera. The actual transmission of the video stream is outside the scope of this service.

Table 8: AnalogVideoSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryAnalogVideoSpecs⊕	reportAnalogVideoSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.2.1 reportAnalogVideoSpecs

Description: This operation is used to report the capabilities of a analog video sensor.

Namespace: UMAA::SEM::AnalogVideoSpecs

Topic: AnalogVideoSpecsReport

Data Type: AnalogVideoSpecsReportType

Table 9: AnalogVideoSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
videoFormat	VideoFormatEnumType	Video format that is supporting by the analog video sensor

6.1.3 BeaconParametersControl

The purpose of this service is to provide a mechanism to control the beacon sensor on-board of the unmanned platform.

Table 10: BeaconParametersControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setBeaconParameters	reportBeaconParametersCommandStatus
queryBeaconParametersCommandAck⊕	reportBeaconParametersCommandAck
cancelBeaconParametersCommand⊕	reportBeaconParametersCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.3.1 reportBeaconParametersCommandAck

Description: This operation is used to report the current beacon system command on-board of the unmanned platform.

Namespace: UMAA::SEM::BeaconParametersControl

Topic: BeaconParametersCommandAckReport

Data Type: BeaconParametersCommandAckReportType

Table 11: BeaconParametersCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
frequencyBand	Count	A desired frequency band
IRLightsArmed	BooleanEnumType	A desired armed/disarmed infrared light
IRLightsOn	BooleanEnumType	A desired on/off infrared light
locatorArmed	BooleanEnumType	A desired armed/disarmed of the locator
locatorOn	BooleanEnumType	A desired on/off locator
multibandLocator	BooleanEnumType	A desired on/off of the multiband locator
RGLightsArmed	BooleanEnumType	A desired armed/disarmed green light
RGLightsOn	BooleanEnumType	A desired on/off green light
transmitChannel	Count	A desired transmit channel

6.1.3.2 reportBeaconParametersCommandStatus

Description: This operation is used to report the current status of the beacon system command on-board of the unmanned platform.

Namespace: UMAA::SEM::BeaconParametersControl

Topic: BeaconParametersCommandStatus

Data Type: BeaconParametersCommandStatusType

Table 12: BeaconParametersCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.3.3 setBeaconParameters

Description: This operation is used to control beacon system on-board of the unmanned platform.

Namespace: UMAA::SEM::BeaconParametersControl

Topic: BeaconParametersCommand

Data Type: BeaconParametersCommandType

Table 13: BeaconParametersCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
frequencyBand	Count	A desired frequency band
IRLightsArmed	BooleanEnumType	A desired armed/disarmed infrared light
IRLightsOn	BooleanEnumType	A desired on/off infrared light
locatorArmed	BooleanEnumType	A desired armed/disarmed of the locator
locatorOn	BooleanEnumType	A desired on/off locator
multibandLocator	BooleanEnumType	A desired on/off of the multiband locator
RGLightsArmed	BooleanEnumType	A desired armed/disarmed green light
RGLightsOn	BooleanEnumType	A desired on/off green light
transmitChannel	Count	A desired transmit channel

6.1.4 BeaconParametersStatus

The purpose of this service is to provide a mechanism to report status from the beacon sensor on-board of the unmanned platform.

Table 14: BeaconParametersStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBeaconParameters⊕	reportBeaconParameters

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.4.1 reportBeaconParameters

Description: This operation is used to report the current status of beacon system on-board of the unmanned platform.

Namespace: UMAA::SEM::BeaconParametersStatus

Topic: BeaconParametersReport

Data Type: BeaconParametersReportType

Table 15: BeaconParametersReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
frequencyBand	Count	A current frequency band
IRLightsArmed	BooleanEnumType	A current armed/disarmed status of the infrared light
IRLightsOn	BooleanEnumType	A current on/off status of the infrared light
locatorArmed	BooleanEnumType	A current armed/disarmed status of the locator
locatorOn	BooleanEnumType	A current on/off status locator
multibandLocator	BooleanEnumType	A current on/off status of the multiband locator
RGLightsArmed	BooleanEnumType	The current armed/disarmed status of the green light
RGLightsOn	BooleanEnumType	A current on/off status the green light
transmitChannel	Count	A current transmit channel

6.1.5 BilgePumpConfig

The purpose of this service is to provide the operations and interfaces to report the configuration of the bilge pumps on the vehicle.

Table 16: BilgePumpConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBilgePumpConfig⊕	reportBilgePumpConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.5.1 reportBilgePumpConfig

Description: This operation is used to report the configuration of the bilge pumps on the unmanned platform.

Namespace: UMAA::EO::BilgePumpConfig

Topic: BilgePumpConfigReport

Data Type: BilgePumpConfigReportType

Table 17: BilgePumpConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	The name of the bilge pump

6.1.6 BilgePumpControl

The purpose of this service is to provide the operations and interfaces to control and monitor the bilge pumps on the vehicle. Three modes of operation, Off, On, and Auto are supported per bilge pump. The auto mode means the pump will automatically be turned on by the service when flood is detected in its responsible area. Note: Flood and leak reporting is provided by the Compartment Sensor Service.

Table 18: BilgePumpControl Operations

Service Requests (Inputs)	**Service Responses (Outputs)**
setBilgePump	reportBilgePumpCommandStatus
queryBilgePumpCommandAck⊕	reportBilgePumpCommandAck
cancelBilgePumpCommand⊕	reportBilgePumpCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.6.1 reportBilgePumpCommandAck

Description: This operation is used to report the commanded bilge pump state on the unmanned platform.

Namespace: UMAA::EO::BilgePumpControl

Topic: BilgePumpCommandAckReport

Data Type: BilgePumpCommandAckReportType

Table 19: BilgePumpCommandAckReportType Message Definition

Attribute Name	**Attribute Type**	**Attribute Description**
Additional fields included from UMAA::UMAACommandStatusBase		
state	BilgeControlEnumType	The desired state of the bilge pump. When in auto state, the pump will be automatically turned on when flood is detected.

6.1.6.2 reportBilgePumpCommandStatus

Description: This operation is used to report the status of the bilge pump command.

Namespace: UMAA::EO::BilgePumpControl

Topic: BilgePumpCommandStatus

Data Type: BilgePumpCommandStatusType

Table 20: BilgePumpCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.6.3 setBilgePump

Description: This operation is used to activate the state of the bilge pumps on the unmanned platform. When in auto state, the pump will be automatically turned on when flood is detected. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::EO::BilgePumpControl

Topic: BilgePumpCommand

Data Type: BilgePumpCommandType

Table 21: BilgePumpCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
state	BilgeControlEnumType	The desired state of the bilge pump. When in auto state, the pump will be automatically turned on when flood is detected.

6.1.7 BilgePumpStatus

The purpose of this service is to provide the operations and interfaces to monitor the bilge pumps on the vehicle. Three modes of operation, Off, On, and Auto are supported per bilge pump. The auto mode means the pump will automatically be turned on by the service when flood is detected in its responsible area. Note: Flood and leak reporting is provided by the Compartment Sensor Service.

Table 22: BilgePumpStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBilgePump⊕	reportBilgePump

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.7.1 reportBilgePump

Description: This operation is used to report the current state of the bilge pumps on the unmanned platform.

Namespace: UMAA::EO::BilgePumpStatus

Topic: BilgePumpReport

Data Type: BilgePumpReportType

Table 23: BilgePumpReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
state	BilgeStateEnumType	The state of the bilge pump

6.1.8 BucketConfig

The purpose of this service is to provide the operations and interfaces to report bucket configuration. A water jet driven boat typically has a bucket for controlling the forward/reverse thrust.

Table 24: BucketConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBucketConfig⊕	reportBucketConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.8.1 reportBucketConfig

Description: This operation is used to report the configuration of the buckets on the unmanned surface platform.

Namespace: UMAA::EO::BucketConfig

Topic: BucketConfigReport

Data Type: BucketConfigReportType

Table 25: BucketConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	The name of the bucket

6.1.9 BucketControl

The purpose of this service is to provide the operations and interfaces to set bucket positions. A water jet driven boat typically has a bucket for controlling the forward/reverse thrust.

Table 26: BucketControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setBucket	reportBucketCommandStatus
queryBucketCommandAck⊕	reportBucketCommandAck
cancelBucketCommand⊕	reportBucketCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.9.1 reportBucketCommandAck

Description: This operation is used to report the commanded bucket position of the unmanned surface platform.

Namespace: UMAA::EO::BucketControl

Topic: BucketCommandAckReport

Data Type: BucketCommandAckReportType

Table 27: BucketCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
bucketPosition	AnglePosition	The desired position of the bucket to control forward or reserve thrust

6.1.9.2 reportBucketCommandStatus

Description: This operation is used to report the status of the bucket position command.

Namespace: UMAA::EO::BucketControl

Topic: BucketCommandStatus

Data Type: BucketCommandStatusType

Table 28: BucketCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.9.3 setBucket

Description: This operation is used to control the bucket position for forward or reverse thrust of the unmanned surface platform.

Namespace: UMAA::EO::BucketControl

Topic: BucketCommand

Data Type: BucketCommandType

Table 29: BucketCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
bucketPosition	AnglePosition	The desired position of the bucket to control forward or reserve thrust

6.1.10 BucketStatus

The purpose of this service is to provide the operations and interfaces to report bucket positions. A water jet driven boat typically has a bucket for controlling the forward/reverse thrust.

Table 30: BucketStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBucket⊕	reportBucket

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.10.1 reportBucket

Description: This operation is used to report the current position of the bucket on the unmanned surface platform.

Namespace: UMAA::EO::BucketStatus

Topic: BucketReport

Data Type: BucketReportType

Table 31: BucketReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAAStatus	
bucketPosition	AnglePosition	The current bucket position

6.1.11 CameraConfig

The purpose of this service is to provide basic configuration of the camera on board of the unmanned platform.

Table 32: CameraConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryCameraConfig⊕	reportCameraConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.11.1 reportCameraConfig

Description: This operation is used to report the current configuration of the camera.

Namespace: UMAA::SEM::CameraControl

Topic: CameraConfigReport

Data Type: CameraConfigReportType

Table 33: CameraConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
aperture	Distance_Precise	The current aperture of the camera.
exposureMode	ExposureModeEnumType	The current exposure mode.
focalLength	Distance_Precise	The current focal length of the sensor.
focusMode	AutomationEnumType	The current focus mode of the camera.
focusValue	FocusValue	The current focus value that allows incremental changes in focus position based on the percentage of the entire focal range of the camera.
horizontalFOV	FieldOfView_LineOfSightFocalPlane	The current horizontal Field of View (FOV) of the camera.
imageStabilization	BooleanEnumType	The current image stabilization.
imagingMode	ImagingModeEnumType	The current imaging mode.
lightSensitivity	LightSensitivityEnumType	The current light sensitivity of the camera.
maxZoomLevel	ContinuousSize	The current maximum zoom level
meteringMode	MeteringModeEnumType	The current metering mode.
minZoomLevel	ContinuousSize	The current minimum zoom level
shutterSpeed	Duration_Seconds	The current shutter speed of the camera.
status	PowerStatusEnumType	The current power status.
verticalFOV	FieldOfView_LineOfSightFocalPlane	The current vertical Field of View (FOV) of the camera.
whiteBalance	WhiteBalanceEnumType	The current white balance.
zoomLevel	ContinuousSize	The current zoom level (0.0 to 100.0)
zoomMode	ZoomModeEnumType	The current zoom mode.

6.1.12 CameraControl

The purpose of this service is to provide control of the camera on board of the unmanned platform.

Table 34: CameraControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setCamera	reportCameraCommandStatus
queryCameraCommandAck⊕	reportCameraCommandAck
cancelCameraCommand⊕	reportCameraCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.12.1 reportCameraCommandAck

Description: This operation is used to report the current set camera command.

Namespace: UMAA::SEM::CameraControl

Topic: CameraCommandAckReport

Data Type: CameraCommandAckReportType

Table 35: CameraCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
builtInTest†	BuiltInTestStatusType_All	Used to run a built-in test to the payload.
filtered	BooleanEnumType	The desired image filtered..
imageCenterLocation	Position2D	The desired center location of the image
mode	IRPolarityEnumType	The desired image polarity of the camera.

6.1.12.2 reportCameraCommandStatus

Description: This operation is used to report the current status of the set camera command.

Namespace: UMAA::SEM::CameraControl

Topic: CameraCommandStatus

Data Type: CameraCommandStatusType

Table 36: CameraCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.12.3 setCamera

Description: This operation is used to set image center, mode and filtered of a camera.

Namespace: UMAA::SEM::CameraControl

Topic: CameraCommand

Data Type: CameraCommandType

Table 37: CameraCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
builtInTest†	BuiltInTestStatusType_All	Used to run a built-in test to the payload.
filtered	BooleanEnumType	The desired image filtered..
imageCenterLocation	Position2D	The desired center location of the image
mode	IRPolarityEnumType	The desired image polarity of the camera.

6.1.13 CameraSelectionControl

The purpose of this service is to provide a mechanism to select a camera on a particular multiplexer.

Table 38: CameraSelectionControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setCameraSelection	reportCameraSelectionCommandStatus
queryCameraSelectionCommand⊕	reportCameraSelectionCommand
cancelCameraSelectionCommand⊕	reportCameraSelectionCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.13.1 reportCameraSelectionCommand

Description: This operation is used to report the current camera command.

Namespace: UMAA::SEM::CameraSelectionControl

Topic: CameraSelectionCommandAckReport

Data Type: CameraSelectionCommandAckReportType

Table 39: CameraSelectionCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
cameraID	NumericGUID	An unique identification of the selecting camera
muxID*	NumericGUID	An unique identification of the multiplexer in the unmanned platform

6.1.13.2 reportCameraSelectionCommandStatus

Description: This operation is used to report the status of the camera command.

Namespace: UMAA::SEM::CameraSelectionControl

Topic: CameraSelectionCommandStatus

Data Type: CameraSelectionCommandStatusType

Table 40: CameraSelectionCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.13.3 setCameraSelection

Description: This operation is used to select the camera with the associated multiplexer.

Namespace: UMAA::SEM::CameraSelectionControl

Topic: CameraSelectionCommand

Data Type: CameraSelectionCommandType

Table 41: CameraSelectionCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
cameraID	NumericGUID	An unique identification of the selecting camera
muxID*	NumericGUID	An unique identification of the multiplexer in the unmanned platform

6.1.14 CameraSelectionSpecs

The purpose of this service is to report the capabilities of a camera on a particular multiplexer.

Table 42: CameraSelectionSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryCameraSelectionSpecs⊕	reportCameraSelectionSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.14.1 reportCameraSelectionSpecs

Description: This operation is used to report the capability of the camera with the associated multiplexer.

Namespace: UMAA::SEM::CameraSelectionSpecs

Topic: CameraSelectionSpecsReport

Data Type: CameraSelectionSpecsReportType

Table 43: CameraSelectionSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
muxID	NumericGUID	An unique identification for the multiplexers in the unmanned platform
cameraID*	NumericGUID	An unique identification of the selected camera

6.1.15 CameraSelectionStatus

The purpose of this service is to report the current camera selection.

Table 44: CameraSelectionStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryCameraSelection⊕	reportCameraSelection

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.15.1 reportCameraSelection

Description: This operation is used to report the current selection of the camera with the associated multiplexer.

Namespace: UMAA::SEM::CameraSelectionStatus

Topic: CameraSelectionReport

Data Type: CameraSelectionReportType

Table 45: CameraSelectionReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
cameraID	NumericGUID	An unique identification of the currently selected camera associated with the defined multiplexer
muxID*	NumericGUID	An unique identification for the multiplexers in the unmanned platform

6.1.16 CameraSpecs

The purpose of this service is to provide the capabilities of the camera on board of the unmanned platform.

Table 46: CameraSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryCameraSpecs⊕	reportCameraSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.16.1 reportCameraSpecs

Description: This operation is used to report the capabilities of the camera.

Namespace: UMAA::SEM::CameraSpecs

Topic: CameraSpecsReport

Data Type: CameraSpecsReportType

Table 47: CameraSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
exposureModeAperture	BooleanEnumType	The unsupported/supported exposure aperture priority mode
exposureModeAuto	BooleanEnumType	The unsupported/supported exposure auto mode
exposureModeManual	BooleanEnumType	The unsupported/supported exposure manual mode.
exposureModeShutter	BooleanEnumType	The unsupported/supported exposure shutter priority mode

Attribute Name	Attribute Type	Attribute Description
focusModeAuto	BooleanEnumType	The unsupported/supported auto focus mode
focusModeManual	BooleanEnumType	The unsupported/supported manual focus mode
imageStabilization	BooleanEnumType	The true/false image stabilization/vibration reduction
imagingModeColor	BooleanEnumType	The unsupported/supported imaging color mode
imagingModeGreyscale	BooleanEnumType	The unsupported/supported imaging greyscale mode
imagingModeInfrared	BooleanEnumType	The unsupported/supported imaging infrared mode
imagingModeLowlight	BooleanEnumType	The unsupported/supported imaging lowlight mode
lightSensitivity_ISO100	BooleanEnumType	The unsupported/supported light sensitivity level
lightSensitivity_ISO1600	BooleanEnumType	The unsupported/supported light sensitivity level
lightSensitivity_ISO200	BooleanEnumType	The unsupported/supported light sensitivity level
lightSensitivity_ISO3200	BooleanEnumType	The unsupported/supported light sensitivity level
lightSensitivity_ISO400	BooleanEnumType	The unsupported/supported light sensitivity level
lightSensitivity_ISO800	BooleanEnumType	The unsupported/supported light sensitivity level
lightSensitivityAuto	BooleanEnumType	The unsupported/supported light sensitivity level
maxAperture	Distance_Precise	The aperture diameter (0.1 to 128)
maxFocalLength	Distance_Precise	The focal length (0.0 to 2.0)
maxShutterSpeed	Duration_Seconds	The shutter speed (0.0 to 60.0)
meteringModeAuto	BooleanEnumType	The unsupported/supported exposure metering auto mode
meteringModeCenterWeighted	BooleanEnumType	The unsupported/supported exposure metering manual mode
meteringModeSpot	BooleanEnumType	The unsupported/supported exposure metering spot mode
minAperture	Distance_Precise	The aperture diameter (0.1 to 128)
minFocalLength	Distance_Precise	The focal length (0.0 to 2.0)
minShutterSpeed	Duration_Seconds	The shutter speed (0.0 to 60.0)
name	StringShortDescription	Description of the camera
stateActive	BooleanEnumType	The unsupported/supported active state
stateOff	BooleanEnumType	The unsupported/supported off state
stateStandby	BooleanEnumType	The unsupported/supported standby state
whiteBalanceAuto	BooleanEnumType	The unsupported/supported auto white balance
whiteBalanceCloudy	BooleanEnumType	The unsupported/supported cloudy white balance
whiteBalanceDaylight	BooleanEnumType	The unsupported/supported daylight white balance
whiteBalanceFlash	BooleanEnumType	The unsupported/supported flash white balance
whiteBalanceFluorescent	BooleanEnumType	The unsupported/supported fluorescent white balance
whiteBalanceShade	BooleanEnumType	The unsupported/supported shade white balance
whiteBalanceTungsten	BooleanEnumType	The unsupported/supported tungsten white balance
zoomModeAnalog	BooleanEnumType	The unsupported/supported zoom AnalogOnly mode
zoomModeDigital	BooleanEnumType	The unsupported/supported zoom DigitalOnly mode.
zoomModeMixed	BooleanEnumType	The unsupported/supported zoom mixed mode
zoomModeOff	BooleanEnumType	The unsupported/supported None zoom mode

6.1.17 CameraStatus

The purpose of this service is to provide the status of the camera on board of the unmanned platform.

Table 48: CameraStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryCamera⊕	reportCamera

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.17.1 reportCamera

Description: This operation is used to report the current status of the camera.

Namespace: UMAA::SEM::CameraStatus

Topic: CameraReport

Data Type: CameraReportType

Table 49: CameraReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
filtered	BooleanEnumType	The current image filtered
imageCenterLocation	Position2D	The current center location of the image
mode	IRPolarityEnumType	The current image polarity of the camera

6.1.18 CommsLostPolicy

The purpose of this service is to provide a mechanism to retrieve the unmanned platform communication system's capability and configuration and for setting the behavior for a lost communication event.

Table 50: CommsLostPolicy Operations

Service Requests (Inputs)	Service Responses (Outputs)
setCommsLostPolicy	reportCommsLostPolicyCommandStatus
queryCommsLostPolicyCommand⊕	reportCommsLostPolicyCommand
cancelCommsLostPolicyCommand⊕	reportCommsLostPolicyCancelCommandStatus⊕
queryCommsLostPolicy⊕	reportCommsLostPolicy
queryCommsLostPolicyConfig⊕	reportCommsLostPolicyConfig
queryCommsLostPolicySpecs⊕	reportCommsLostPolicySpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.18.1 reportCommsLostPolicy

Description: This operation is used to report the current lost communication status of the unmanned platform.

Namespace: UMAA::MM::CommsLostPolicy

Topic: CommsLostPolicyReport

Data Type: CommsLostPolicyReportType

Table 51: CommsLostPolicyReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
status	LostCommsStatusEnumType	The policy status and execution state.

6.1.18.2 reportCommsLostPolicyCommand

Description: This operation is used to report the current lost communication policy command.

Namespace: UMAA::MM::CommsLostPolicy

Topic: CommsLostPolicyCommandReport

Data Type: CommsLostPolicyCommandReportType

Table 52: CommsLostPolicyCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
commsLostTimeOut	Duration_Seconds	Number of seconds of no contact after which communication is considered lost and the policy is initiated.
commsRegainedStop	BooleanEnumType	If true, the robot will stop when comms are regained. Otherwise it will do nothing.
commsRegainedTimeOut	Duration_Seconds	Number of seconds of communication with previously controlling client to consider communications regained.
continueMission†	BooleanEnumType	If present, the robot will continue its mission when comms are lost.
moveToPos†	MoveToPosPolicyType	If present, the robot will move to a specific position when comms are lost.
retrotraverse†	RetrotraversePolicyType	If present, the robot will retrotraverse as specified when comms are lost.
stopMotion†	BooleanEnumType	If present, the robot will stop moving when comms are lost.

6.1.18.3 reportCommsLostPolicyCommandStatus

Description: This operation is used to report a response whether the unmanned platform support the lost communication behaviors.

Namespace: UMAA::MM::CommsLostPolicy

Topic: CommsLostPolicyCommandStatus

Data Type: CommsLostPolicyCommandStatusType

Table 53: CommsLostPolicyCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.18.4 reportCommsLostPolicyConfig

Description: This operation is used to report the current lost communication configuration of the unmanned platform.

Namespace: UMAA::MM::CommsLostPolicy

Topic: CommsLostPolicyConfigReport

Data Type: CommsLostPolicyConfigReportType

Table 54: CommsLostPolicyConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
commsLostTimeOut	Duration_Seconds	Number of seconds of no contact after which communication is considered lost and the policy is initiated.
commsRegainedStop	BooleanEnumType	If true, the robot will stop when comms are regained. Otherwise it will do nothing.
commsRegainedTimeOut	Duration_Seconds	Number of seconds of communication with previously controlling client to consider communications regained.
continueMission†	BooleanEnumType	If present, the robot will continue its mission when comms are lost.
moveToPos†	MoveToPosPolicyType	If present, the robot will move to a specific position when comms are lost.
retrotraverse†	RetrotraversePolicyType	If present, the robot will retrotraverse as specified when comms are lost.
stopMotion†	BooleanEnumType	If present, the robot will stop moving when comms are lost.

6.1.18.5 reportCommsLostPolicySpecs

Description: This operation is used to report the lost communication capability of the unmanned platform.

Namespace: UMAA::MM::CommsLostPolicy

Topic: CommsLostPolicySpecsReport

Data Type: CommsLostPolicySpecsReportType

Table 55: CommsLostPolicySpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
continueMission	BooleanEnumType	Flag indicating support for continue policy.
moveToPose	BooleanEnumType	Flag indicating support for move to pose policy.
retroTraverse	BooleanEnumType	Flag indicating support for retrograde traverse policy.
stop	BooleanEnumType	Flag indicating support for stop policy.

6.1.18.6 setCommsLostPolicy

Description: This operation is used to set the active behavior for lost communication event for the unmanned platform.

Namespace: UMAA::MM::CommsLostPolicy

Topic: CommsLostPolicyCommand

Data Type: CommsLostPolicyCommandType

Table 56: CommsLostPolicyCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
commsLostTimeOut	Duration_Seconds	Number of seconds of no contact after which communication is considered lost and the policy is initiated.
commsRegainedStop	BooleanEnumType	If true, the robot will stop when comms are regained. Otherwise it will do nothing.
commsRegainedTimeOut	Duration_Seconds	Number of seconds of communication with previously controlling client to consider communications regained.
continueMission†	BooleanEnumType	If present, the robot will continue its mission when comms are lost.
moveToPos†	MoveToPosPolicyType	If present, the robot will move to a specific position when comms are lost.
retrotraverse†	RetrotraversePolicyType	If present, the robot will retrotraverse as specified when comms are lost.
stopMotion†	BooleanEnumType	If present, the robot will stop moving when comms are lost.

6.1.19 ControlSystemID

The purpose of this service is to report the information of a control system and its client(s).

Table 57: ControlSystemID Operations

Service Requests (Inputs)	Service Responses (Outputs)
setControlSystemID	reportControlSystemIDCommandStatus
queryControlSystemIDCommand⊕	reportControlSystemIDCommand
cancelControlSystemIDCommand⊕	reportControlSystemIDCancelCommandStatus⊕
queryControlSystemID⊕	reportControlSystemID
queryClientID⊕	reportClientID

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.19.1 reportClientID

Description: This operation is used to report the information of client(s) within a control system.

Namespace: UMAA::MM::ControlSystemID

Topic: ClientIDReport

Data Type: ClientIDReportType

Table 58: ClientIDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	A name to describe a user or a subset of a control system.

6.1.19.2 reportControlSystemID

Description: This operation is used to report the information of a control system.

Namespace: UMAA::MM::ControlSystemID

Topic: ControlSystemIDReport

Data Type: ControlSystemIDReportType

Table 59: ControlSystemIDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	A name to describe a control station or a control system.

6.1.19.3 reportControlSystemIDCommand

Description: This operation is used to report the current set control system ID command

Namespace: UMAA::MM::ControlSystemID

Topic: ControlSystemIDCommandReport

Data Type: ControlSystemIDCommandReportType

Table 60: ControlSystemIDCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	The name of the control system

6.1.19.4 reportControlSystemIDCommandStatus

Description: This operation is used to report the status of the set control system ID command

Namespace: UMAA::MM::ControlSystemID

Topic: ControlSystemIDCommandStatus

Data Type: ControlSystemIDCommandStatusType

Table 61: ControlSystemIDCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.19.5 setControlSystemID

Description: This operation is used to set the control system ID for an service that might not know it.

Namespace: UMAA::MM::ControlSystemID

Topic: ControlSystemIDCommand

Data Type: ControlSystemIDCommandType

Table 62: ControlSystemIDCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
name	StringShortDescription	The name of the control system

6.1.20 ControlTransfer

The purpose of this service is to control and manage ownership of an unmanned platform, a system, or a payload.

Table 63: ControlTransfer Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryClientControl⊕	reportClientControl
queryControlSystemControl⊕	reportControlSystemControl
queryControlSystemTransfer⊕	reportControlSystemTransfer
queryControlTransfer⊕	reportControlTransfer

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.20.1 reportClientControl

Description: This operation is used to report which client within the control system is in control of an unmanned platform, a system or a payload.

Namespace: UMAA::MM::ControlTransfer

Topic: ClientControlReport

Data Type: ClientControlReportType

Table 64: ClientControlReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAAStatus	
authorityLevel	Count	Value between 0 and 255 with 0 indicating no control and 255 being root access. The control arbiter may use this value to automatically force a control release or this value may be displayed to the controlling client user as an indication of control urgency.

Attribute Name	Attribute Type	Attribute Description
clientID	NumericGUID	A unique identification of an operator or a subsystem that controls the unmanned platform or a payload. However, if status is AVAILABLE, clientID should not be set.
status	ResourceAllocationStatusEnumType	A control status of the unmanned platform, a system or a payload

6.1.20.2 reportControlSystemControl

Description: This operation is used to report which control system is in control of an unmanned platform, a system or a payload.

Namespace: UMAA::MM::ControlTransfer

Topic: ControlSystemControlReport

Data Type: ControlSystemControlReportType

Table 65: ControlSystemControlReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
authorityLevel	Count	Value between 0 and 255 with 0 indicating no control and 255 being root access. The control arbiter may use this value to automatically force a control release or this value may be displayed to the controlling client user as an indication of control urgency.
controlSystemID	NumericGUID	A unique identification of a system that controls the unmanned platform or a payload. However, if status is AVAILABLE, controlSystemID should not be set.
status	ResourceAllocationStatusEnumType	A control status of the unmanned platform, a system or a payload

6.1.20.3 reportControlSystemTransfer

Description: This operation is used to report a control request by a control system to control of an unmanned platform, a system or a payload.

Namespace: UMAA::MM::ControlTransfer

Topic: ControlSystemTransferReport

Data Type: ControlSystemTransferReportType

Table 66: ControlSystemTransferReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
authorityLevel	Count	Value between 0 and 255 with 0 indicating no control and 255 being root access. The control arbiter may use this value to automatically force a control release or this value may be displayed to the controlling client user as an indication of control urgency.
result	HandoverResultEnumType	Needs these values: GRANTED, NOT_AVAILABLE, TIMEOUT, DENIED, DEFERRED, INSUFFICIENT_AUTHORITY

6.1.20.4 reportControlTransfer

Description: This operation is used to report a control request by a client to control of an unmanned platform, a system or a payload.

Namespace: UMAA::MM::ControlTransfer

Topic: ClientControlTransferReport

Data Type: ClientControlTransferReportType

Table 67: ClientControlTransferReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
authorityLevel	Count	Value between 0 and 255 with 0 indicating no control and 255 being root access. The control arbiter may use this value to automatically force a control release or this value may be displayed to the controlling client user as an indication of control urgency.
result	HandoverResultEnumType	Needs these values: GRANTED, NOT_AVAILABLE, TIMEOUT, DENIED, DEFERRED, INSUFFICIENT_AUTHORITY

6.1.21 DigitalAudioConfig

The purpose of this service is to provide a means of configuring a digital audio stream, often from a microphone or other source. Note that the transport of the digitized audio stream itself is not covered by this service, and may use existing audio networking standards such as Real Time Streaming Protocol (RTSP).

Table 68: DigitalAudioConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryDigitalAudioConfig⊕	reportDigitalAudioConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.21.1 reportDigitalAudioConfig

Description: This operation is used to report the configuration of a digital audio stream.

Namespace: UMAA::SEM::DigitalAudioConfig

Topic: DigitalAudioConfigReport

Data Type: DigitalAudioConfigReportType

Table 69: DigitalAudioConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
bitDepth	BitDepthEnumType	Audion stream bit depth.
digitalFormat	DigitalAudioFormatEnumType	Audio encoding format.
encodingQuality	AudioEncodingQualityEnumType	Audio quality setting.
maxBitRate	CommsRate_BitsPerSecond	The maximum bit rate (0 to 65535 kbps)
minBitRate	CommsRate_BitsPerSecond	The minimum bit rate (0 to 65535 kbps)
numberOfChannels	Count	Number of audio channels to be sent to Annunciator in stream.
sampleRate	Count	The samples per second
sensitivity	Count	The sensitivity as a percentage (0 to 100)

6.1.22 DigitalAudioSpecs

The purpose of this service is to provide a means to request the capabilities of a digital audio stream, often from a microphone or other source. Note that the transport of the digitized audio stream itself is not covered by this service, and may use existing audio networking standards such as Real Time Streaming Protocol (RTSP).

Table 70: DigitalAudioSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryDigitalAudioSpecs⊕	reportDigitalAudioSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.22.1 reportDigitalAudioSpecs

Description: This operation is used to report the capabilities of a digital audio stream.

Namespace: UMAA::SEM::DigitalAudioSpecs

Topic: DigitalAudioSpecsReport

Data Type: DigitalAudioSpecsReportType

Table 71: DigitalAudioSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
maxAllowedBitRate	CommsRate_BitsPerSecond	The maximum allowed bit rate (0 to 65535 Kbps)
maxChannels	Count	The maximum number of audio channels supported
maxSampleRate	Count	The maximum available sample rate (samples per second)
minAllowedBitRate	CommsRate_BitsPerSecond	The minimum allowed bit rate (0 to 65535 Kbps)
minSampleRate	Count	The minimum available sample rate (samples per second)
supported10bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported12bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported14bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported16bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported20bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported24bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported32bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported40bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported48bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported56bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported64bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supported8bits	BooleanEnumType	The supported/unsupported digital audio bit depth
supportedAAC_MPEG2	BooleanEnumType	The supported/unsupported digital audio format
supportedAAC_MPEG4	BooleanEnumType	The supported/unsupported digital audio format
supportedAIFF	BooleanEnumType	The supported/unsupported digital audio format
supportedALAC	BooleanEnumType	The supported/unsupported digital audio format
supportedAverage	BooleanEnumType	The supported/unsupported digital audio quality level
supportedBest	BooleanEnumType	The supported/unsupported digital audio quality level
supportedBetter	BooleanEnumType	The supported/unsupported digital audio quality level
supportedDolbyDigital	BooleanEnumType	The supported/unsupported digital audio format
supportedDTS	BooleanEnumType	The supported/unsupported digital audio format
supportedFLAC	BooleanEnumType	The supported/unsupported digital audio format
supportedGood	BooleanEnumType	The supported/unsupported digital audio quality level
supportedLess	BooleanEnumType	The supported/unsupported digital audio quality level
supportedLPCM_PCM	BooleanEnumType	The supported/unsupported digital audio format

Attribute Name	Attribute Type	Attribute Description
supportedMP2	BooleanEnumType	The supported/unsupported digital audio format
supportedMP3	BooleanEnumType	The supported/unsupported digital audio format
supportedPoor	BooleanEnumType	The supported/unsupported digital audio quality level
supportedRealAudio	BooleanEnumType	The supported/unsupported digital audio format
supportedSpeex	BooleanEnumType	The supported/unsupported digital audio format
supportedTrueAudio	BooleanEnumType	The supported/unsupported digital audio format
supportedULAW	BooleanEnumType	The supported/unsupported digital audio format
supportedVORBIS	BooleanEnumType	The supported/unsupported digital audio format
supportedWAV	BooleanEnumType	The supported/unsupported digital audio format
supportedWMA	BooleanEnumType	The supported/unsupported digital audio format
supportedWMA9_Lossless	BooleanEnumType	The supported/unsupported digital audio format
supportedWorst	BooleanEnumType	The supported/unsupported digital audio quality level

6.1.23 DigitalAudioStreamControl

The purpose of this service is to allow a client to specify an RTSP stream as an audio source.

Table 72: DigitalAudioStreamControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setDigitalAudioStream	reportDigitalAudioStreamCommandStatus
queryDigitalAudioStreamCommandAck⊕	reportDigitalAudioStreamCommandAck
cancelDigitalAudioStreamCommand⊕	reportDigitalAudioStreamCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.23.1 reportDigitalAudioStreamCommandAck

Description: This operation is used to report the current command that set the source of a digital audio stream.

Namespace: UMAA::SEM::DigitalAudioStreamControl

Topic: DigitalAudioStreamCommandAckReport

Data Type: DigitalAudioStreamCommandAckReportType

Table 73: DigitalAudioStreamCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
repeat	BooleanEnumType	The on/off value indicating if a finite stream should be replayed once it finishes (true == repeat until new stream is specified)

Attribute Name	Attribute Type	Attribute Description
url	StringShortDescription	The URL source of the stream. This URL should not require a DNS to resolve; hence, and IP address should be substituted for a host name.

6.1.23.2 reportDigitalAudioStreamCommandStatus

Description: This operation is used to report the status of the command that set the source of a digital audio stream.

Namespace: UMAA::SEM::DigitalAudioStreamControl

Topic: DigitalAudioStreamCommandStatus

Data Type: DigitalAudioStreamCommandStatusType

Table 74: DigitalAudioStreamCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.23.3 setDigitalAudioStream

Description: This operation is used to set the source of a digital audio stream.

Namespace: UMAA::SEM::DigitalAudioStreamControl

Topic: DigitalAudioStreamCommand

Data Type: DigitalAudioStreamCommandType

Table 75: DigitalAudioStreamCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
repeat	BooleanEnumType	The on/off value indicating if a finite stream should be replayed once it finishes (true == repeat until new stream is specified)
url	StringShortDescription	The URL source of the stream. This URL should not require a DNS to resolve; hence, and IP address should be substituted for a host name.

6.1.24 DigitalAudioStreamStatus

The purpose of this service is to allow a client to request the source of digital audio stream.

Table 76: DigitalAudioStreamStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryDigitalAudioStream⊕	reportDigitalAudioStream

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.24.1 reportDigitalAudioStream

Description: This operation is used to report the source of a digital audio stream.

Namespace: UMAA::SEM::DigitalAudioStreamStatus

Topic: DigitalAudioStreamReport

Data Type: DigitalAudioStreamReportType

Table 77: DigitalAudioStreamReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
repeat	BooleanEnumType	The on/off value indicating if a finite stream should be replayed once it finishes (true == repeat until new stream is specified)
url	StringShortDescription	The URL source of the stream. This URL should not require a DNS to resolve; hence, and IP address should be substituted for a host name.

6.1.25 DigitalVideoConfig

The purpose of this service is to provide access to the configuration of the digital camera.

Table 78: DigitalVideoConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryDigitalVideoConfig⊕	reportDigitalVideoConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.25.1 reportDigitalVideoConfig

Description: This operation is used to report the configuration of the digital video sensor.

Namespace: UMAA::SEM::DigitalVideoConfig

Topic: DigitalVideoConfigReport

Data Type: DigitalVideoConfigReportType

Table 79: DigitalVideoConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
commsProtocol	NetworkProtocolEnumType	The current communications protocol
commsRate	CommsRate_MegabitsPerSecond	The current communications rate
digitalVideoError	DigitalSensorErrorType	Digital video error code and description
format	DataEncodingEnumType	The current format of data encoding
frameRate	FrameRateFPS	The current frame rate of a digital video stream
frameSize	FrameSizeEnumType	The current frame size of a digital video stream
IPaddress	StringShortDescription	The current IP address to access the digital video stream
IPPort	IPPort_Counting	The current IP port to access the digital video stream
maxBitRate	DataTransferRate	The current bit rate of a digital video stream
minBitRate	DataTransferRate	The current bit rate of a digital video stream
transportEncoding	TransportEncodingEnumType	The current encoding method for transport
URI	UniformResourceIdentifier	The current location of the digital video stream

6.1.26 DigitalVideoControl

The purpose of this service is to provide access to control video and associated audio stream.

Table 80: DigitalVideoControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setDigitalVideo	reportDigitalVideoCommandStatus
queryDigitalVideoCommandAck⊕	reportDigitalVideoCommandAck
cancelDigitalVideoCommand⊕	reportDigitalVideoCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.26.1 reportDigitalVideoCommandAck

Description: This operation is used to report the current command to the digital video or audio sensor.

Namespace: UMAA::SEM::DigitalVideoControl

Topic: DigitalVideoCommandAckReport

Data Type: DigitalVideoCommandAckReportType

Table 81: DigitalVideoCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
builtInTest†	BuiltInTestStatusType_All	Used to run a built-in test to the payload.
state	StreamStateEnumType	The desired state of the stream of the playback video or audio.

6.1.26.2 reportDigitalVideoCommandStatus

Description: This operation is used to report the status of the digital video command.

Namespace: UMAA::SEM::DigitalVideoControl

Topic: DigitalVideoCommandStatus

Data Type: DigitalVideoCommandStatusType

Table 82: DigitalVideoCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.26.3 setDigitalVideo

Description: This operation is used to set the configuration of a video or audio sensor.

Namespace: UMAA::SEM::DigitalVideoControl

Topic: DigitalVideoCommand

Data Type: DigitalVideoCommandType

Table 83: DigitalVideoCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
builtInTest†	BuiltInTestStatusType_All	Used to run a built-in test to the payload.
state	StreamStateEnumType	The desired state of the stream of the playback video or audio.

6.1.27 DigitalVideoSpecs

The purpose of this service is to provide access to the capabilities of the digital camera.

Table 84: DigitalVideoSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryDigitalVideoSpecs⊕	reportDigitalVideoSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.27.1 reportDigitalVideoSpecs

Description: This operation is used to report the capabilities of a video sensor.

Namespace: UMAA::SEM::DigitalVideoSpecs

Topic: DigitalVideoSpecsReport

Data Type: DigitalVideoSpecsReportType

Table 85: DigitalVideoSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
maxBitRate	DataTransferRate	The maximum bit rate of the digital video sensor
maxFrameRate	FrameRateFPS	The maximum frame rate of the digital video sensor
minBitRate	DataTransferRate	The minimum bit rate of the digital video sensor
minFrameRate	FrameRateFPS	The minimum frame rate of the digital video sensor
supportedAVI	BooleanEnumType	The unsupported/supported AVI format
supportedCGA_320x200	BooleanEnumType	The unsupported/supported frame size
supportedCIF_1408x1152	BooleanEnumType	The unsupported/supported frame size
supportedCIF_352x288	BooleanEnumType	The unsupported/supported frame size
supportedCIF_704x576	BooleanEnumType	The unsupported/supported frame size
supportedEGA_640x350	BooleanEnumType	The unsupported/supported frame size
supportedH262	BooleanEnumType	The unsupported/supported H_262 format
supportedH263	BooleanEnumType	The unsupported/supported H_263 format
supportedH263PLUS	BooleanEnumType	The unsupported/supported H_263PLUS format
supportedH264	BooleanEnumType	The unsupported/supported H_264 format
supportedHD1080_1920x1080	BooleanEnumType	The unsupported/supported frame size
supportedHD480_852x480	BooleanEnumType	The unsupported/supported frame size
supportedHD720_1280x720	BooleanEnumType	The unsupported/supported frame size
supportedHSXGA_5120x4096	BooleanEnumType	The unsupported/supported frame size
supportedMJPEG	BooleanEnumType	The unsupported/supported MJPEG format
supportedMPEG4	BooleanEnumType	The unsupported/supported MPEG_4 format
supportedQCIF_176x144	BooleanEnumType	The unsupported/supported frame size

Attribute Name	Attribute Type	Attribute Description
supportedQQVGA_160x120	BooleanEnumType	The unsupported/supported frame size
supportedQSXGA_2560x2048	BooleanEnumType	The unsupported/supported frame size
supportedQVGA_320x240	BooleanEnumType	The unsupported/supported frame size
supportedQXGA_2048x1536	BooleanEnumType	The unsupported/supported frame size
supportedSQCIF_128x96	BooleanEnumType	The unsupported/supported frame size
supportedSVGA_800x600	BooleanEnumType	The unsupported/supported frame size
supportedSXGA_1280x1024	BooleanEnumType	The unsupported/supported frame size
supportedUXGA_1600x1200	BooleanEnumType	The unsupported/supported frame size
supportedVGA_640x480	BooleanEnumType	The unsupported/supported frame size
supportedWHSXGA_6400x4096	BooleanEnumType	The unsupported/supported frame size
supportedWHUXGA_7680x4800	BooleanEnumType	The unsupported/supported frame size
supportedWOXGA_2560x1600	BooleanEnumType	The unsupported/supported frame size
supportedWQSXGA_3200x2048	BooleanEnumType	The unsupported/supported frame size
supportedWQUXGA_3840x2400	BooleanEnumType	The unsupported/supported frame size
supportedWSXGA_1600x1024	BooleanEnumType	The unsupported/supported frame size
supportedWUXGA_1920x1200	BooleanEnumType	The unsupported/supported frame size
supportedWVGA_852x480	BooleanEnumType	The unsupported/supported frame size
supportedWXGA_1366x768	BooleanEnumType	The unsupported/supported frame size
supportedXGA_1024x768	BooleanEnumType	The unsupported/supported frame size

6.1.28 ExtendedPayloadStatus

The purpose of this service is to report additional per-joint status information and allow for the specification of a manipulator, pan/tilt unit, or end effector mounted on a manipulator. This covers both ExtendedPrimativeManipulator and ExtendedPrimitivePanTilt services.

Table 86: ExtendedPayloadStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryHostManipulator⊕	reportHostManipulator
queryJointOperationalParams⊕	reportJointOperationalParams
queryPanTiltOperationalParams⊕	reportPanTiltOperationalParams

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.28.1 reportHostManipulator

Description: This operation is used to report the manipulator ID and joint number on which this subsystem is mounted.

Namespace: UMAA::SEM::ExtendedPayloadStatus

Topic: HostManipulatorReport

Data Type: HostManipulatorReportType

Table 87: HostManipulatorReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
jointID	NumericGUID	ID of joint on which this subsystem is mounted
manipulatorID	NumericGUID	Subsystem ID of manipulator on which this subsystem is mounted

6.1.28.2 reportJointOperationalParams

Description: This operation is used to report the operational parameters of a manipulator joint.

Namespace: UMAA::SEM::ExtendedPayloadStatus

Topic: JointOperationalParamsReport

Data Type: JointOperationalParamsReportType

Table 88: JointOperationalParamsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
jointParameters	sequence<JointOperationalParamsType>	List of joints with their operational parameters

6.1.28.3 reportPanTiltOperationalParams

Description: This operation is used to report the operational parameters of both pan/tilt joints at the same time.

Namespace: UMAA::SEM::ExtendedPayloadStatus

Topic: PanTiltOperationalParamsReport

Data Type: PanTiltOperationalParamsReportType

Table 89: PanTiltOperationalParamsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
panParams	OperationalParamsType	Operational parameters for pan joint
tiltParams	OperationalParamsType	Operational parameters for tilt joint

6.1.29 FinControl

The purpose of this service is to provide the control of the deflection of an individual fin (or set of logical fins) for stabilization and mobility of the unmanned platform.

Table 90: FinControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setFin	reportFinCommandStatus
queryFinCommandAck⊕	reportFinCommandAck
cancelFinCommand⊕	reportFinCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.29.1 reportFinCommandAck

Description: This operation is used to report the commanded fin deflection of an individual fin (or set of fins) of the unmanned platform.

Namespace: UMAA::EO::FinControl

Topic: FinCommandAckReport

Data Type: FinCommandAckReportType

Table 91: FinCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
deflection	Angle	The desired fin deflection
deflectionRate	AngleRate	The desired fin deflection rate

6.1.29.2 reportFinCommandStatus

Description: This operation is used to report the status of the fin deflection command.

Namespace: UMAA::EO::FinControl

Topic: FinCommandStatus

Data Type: FinCommandStatusType

Table 92: FinCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.29.3 setFin

Description: This operation is used to control the fin deflection of an individual fin (or set of logical fins) of the unmanned platform.

Namespace: UMAA::EO::FinControl

Topic: FinCommand

Data Type: FinCommandType

Table 93: FinCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
deflection	Angle	The desired fin deflection
deflectionRate	AngleRate	The desired fin deflection rate

6.1.30 FinSpecs

The purpose of this service is to report the specifications of an individual fin (or set of logical fins) for stabilization and mobility of the unmanned platform.

Table 94: FinSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryFinSpecs⊕	reportFinSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.30.1 reportFinSpecs

Description: This operation is used to report the specifications of an individual fin (or set of fins) of the unmanned platform.

Namespace: UMAA::EO::FinSpecs

Topic: FinSpecsReport

Data Type: FinSpecsReportType

Table 95: FinSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
maxDeflectionRate	AngleRate	The maximum rate of fin deflection
maxNegativeDeflection	Angle	The maximum amount of fin deflection in the negative direction
maxPositiveDeflection	Angle	The maximum amount of fin deflection in the positive direction
minDeflectionRate	AngleRate	The minimum rate of fin deflection
name	StringShortDescription	The name of the fin

6.1.31 FinStatus

The purpose of this service is to provide the position and deflection of an individual fin (or set of logical fins) for stabilization and mobility of the unmanned platform.

Table 96: FinStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryFin⊕	reportFin

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.31.1 reportFin

Description: This operation is used to report the current deflection of an individual fin (or set of fins) of the unmanned platform.

Namespace: UMAA::EO::FinStatus

Topic: FinReport

Data Type: FinReportType

Table 97: FinReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
deflection	Angle	The status of deflection direction of the fin on an unmanned undersea platform

Attribute Name	Attribute Type	Attribute Description
deflectionRate	AngleRate	The status of deflection rate of a fin on an unmanned undersea platform

6.1.32 GuardedTeleopConfig

The purpose of this service is to provide a mechanism for retrieving the guarded teleoperation configuration of a vehicle. Guarded teleoperation is normal teleoperation enhanced by obstacle detection and avoidance sensors where the commanded motion is altered based on the guarded teleoperation policy.

Table 98: GuardedTeleopConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryGuardedTeleopConfig⊕	reportGuardedTeleopConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.32.1 reportGuardedTeleopConfig

Description: This operation is used to report the guarded teleop configuration.

Namespace: UMAA::MO::GuardedTeleopConfig

Topic: GuardedTeleopConfigReport

Data Type: GuardedTeleopConfigReportType

Table 99: GuardedTeleopConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
obstacleAvoidance	ObstacleAvoidanceEnumType	Obstacle avoidance behavior.
pathTolerance	Distance	Maximum allowed deviation, in meters, for obstacle avoidance. A value of 0 indicates infinite tolerance.
stopOnPitchoverLimit	BooleanEnumType	Automatic stop when pitchover limit is reached
stopOnRolloverLimit	BooleanEnumType	Automatic stop when rollover limit is reached

6.1.33 GuardedTeleopControl

The purpose of this service is to provide a mechanism for setting the guarded teleoperation behavior of a vehicle. Guarded teleoperation is normal teleoperation enhanced by obstacle detection and avoidance sensors where the commanded motion is altered based on the guarded teleoperation policy.

Table 100: GuardedTeleopControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGuardedTeleop	reportGuardedTeleopCommandStatus
queryGuardedTeleopCommandAck⊕	reportGuardedTeleopCommandAck
cancelGuardedTeleopCommand⊕	reportGuardedTeleopCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.33.1 reportGuardedTeleopCommandAck

Description: This operation is used to report the current guarded teleop policy command.

Namespace: UMAA::MO::GuardedTeleopControl

Topic: GuardedTeleopCommandAckReport

Data Type: GuardedTeleopCommandAckReportType

Table 101: GuardedTeleopCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
obstacleAvoidance	ObstacleAvoidanceEnumType	Obstacle avoidance behavior.
pathTolerance	Distance	Maximum allowed deviation, in meters, for obstacle avoidance. A value of 0 indicates infinite tolerance.
stopOnPitchoverLimit	BooleanEnumType	Automatic stop when pitchover limit is reached
stopOnRolloverLimit	BooleanEnumType	Automatic stop when rollover limit is reached

6.1.33.2 reportGuardedTeleopCommandStatus

Description: This operation is used to report status of the guarded teleop policy command.

Namespace: UMAA::MO::GuardedTeleopControl

Topic: GuardedTeleopCommandStatus

Data Type: GuardedTeleopCommandStatusType

Table 102: GuardedTeleopCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.33.3 setGuardedTeleop

Description: This operation is used to set the guarded teleop policy configuration.

Namespace: UMAA::MO::GuardedTeleopControl

Topic: GuardedTeleopCommand

Data Type: GuardedTeleopCommandType

Table 103: GuardedTeleopCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from	UMAA::UMAACommand
obstacleAvoidance	ObstacleAvoidanceEnumType	Obstacle avoidance behavior.
pathTolerance	Distance	Maximum allowed deviation, in meters, for obstacle avoidance. A value of 0 indicates infinite tolerance.
stopOnPitchoverLimit	BooleanEnumType	Automatic stop when pitchover limit is reached
stopOnRolloverLimit	BooleanEnumType	Automatic stop when rollover limit is reached

6.1.34 GuardedTeleopSpecs

The purpose of this service is to provide a mechanism for retrieving the guarded teleoperation specifications of a vehicle. Guarded teleoperation is normal teleoperation enhanced by obstacle detection and avoidance sensors where the commanded motion is altered based on the guarded teleoperation policy.

Table 104: GuardedTeleopSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryGuardedTeleopSpecs⊕	reportGuardedTeleopSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.34.1 reportGuardedTeleopSpecs

Description: This operation is used to report the guarded teleop capabilities.

Namespace: UMAA::MO::GuardedTeleopSpecs

Topic: GuardedTeleopSpecsReport

Data Type: GuardedTeleopSpecsReportType

Table 105: GuardedTeleopSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
supportedAvoidObstacle	BooleanEnumType	Supports deviating from path to avoid obstacle
supportedPitchoverLimit	BooleanEnumType	Supports stopping when pitchover limit has been reached
supportedRolloverLimit	BooleanEnumType	Supports stopping when rollover limit has been reached
supportedStopOnObstacle	BooleanEnumType	Supports stopping when an obstacle is blocking path

6.1.35 GuardedTeleopStatus

The purpose of this service is to provide a mechanism for retrieving the guarded teleoperation behavior of a vehicle. Guarded teleoperation is normal teleoperation enhanced by obstacle detection and avoidance sensors where the commanded motion is altered based on the guarded teleoperation policy.

Table 106: GuardedTeleopStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryGuardedTeleop⊕	reportGuardedTeleop

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.35.1 reportGuardedTeleop

Description: This operation is used to report the guarded teleop status.

Namespace: UMAA::MO::GuardedTeleopStatus

Topic: GuardedTeleopReport

Data Type: GuardedTeleopReportType

Table 107: GuardedTeleopReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
status	GuardedTeleoperationStatusEnumType	Current guarded teleop status.

6.1.36 H264VideoEncodingConfig

The purpose of this service is to provide a mechanism for querying the configuration of the H264 encoding of the video sensors.

Table 108: H264VideoEncodingConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryH264VideoEncodingConfig⊕	reportH264VideoEncodingConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.36.1 reportH264VideoEncodingConfig

Description: This operation is used to report the H264 configuration of a video sensor.

Namespace: UMAA::SEM::H264VideoEncodingConfig

Topic: H264VideoEncodingConfigReport

Data Type: H264VideoEncodingConfigReportType

Table 109: H264VideoEncodingConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
gradualDecoderRefresh	BooleanEnumType	The on/off value of Gradual Decoder Refresh, also called Periodic Intra Refresh.
groupOfPictures	Count	Number of interim frames before sending a key frame
preset	H264PresetEnumType	Preconfigured settings for hardware specific H264 settings.
profile	H264EncodingEnumType	H264 profile setting.
regionOfInterest	BooleanEnumType	The on/off value of region of interest (foveation) encoding, if supported.
ROI_height	Count	Height of high bitrate window measured as a percent of total image height (0 to 65535)
ROI_highBitRate	CommsRate_MegabitsPerSecond	Bitrate of ROI window, measured in Mbps (0 to 65535)
ROI_lowBitRate	CommsRate_MegabitsPerSecond	Bitrate of remaining image, measured in Mbps (0 to 65535)
ROI_width	Count	Width of high bitrate window measured as percent of total image width (0 to 65535)
ROI_x	Count	Horizontal start location of high bitrate window (0 to 65535 pixels)
ROI_y	Count	Vertical start location of high bitrate window (0 to 65535 pixels)

6.1.37 H264VideoEncodingControl

The purpose of this service is to provide a mechanism to control H264 video encoding of one or more digital video sensors on the unmanned platform.

Table 110: H264VideoEncodingControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setH264VideoEncoding	reportH264VideoEncodingCommandStatus
queryH264VideoEncodingCommandAck⊕	reportH264VideoEncodingCommandAck
cancelH264VideoEncodingCommand⊕	reportH264VideoEncodingCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.37.1 reportH264VideoEncodingCommandAck

Description: This operation is used to report the current of the H264 encoding command.

Namespace: UMAA::SEM::H264VideoEncodingControl

Topic: H264VideoEncodingCommandAckReport

Data Type: H264VideoEncodingCommandAckReportType

Table 111: H264VideoEncodingCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
gradualDecoderRefresh	BooleanEnumType	The on/off value of Gradual Decoder Refresh, also called Periodic Intra Refresh.
groupOfPictures	Count	Number of interim frames before sending a key frame
preset	H264PresetEnumType	Preconfigured settings for hardware specific H264 settings.
profile	H264EncodingEnumType	H264 profile setting.
regionOfInterest	BooleanEnumType	The on/off value of region of interest (foveation) encoding, if supported.
ROI_height	Count	Height of high bitrate window measured as a percent of total image height (0 to 65535)
ROI_highBitRate	CommsRate_MegabitsPerSecond	Bitrate of ROI window, measured in Mbps (0 to 65535)
ROI_lowBitRate	CommsRate_MegabitsPerSecond	Bitrate of remaining image, measured in Mbps (0 to 65535)
ROI_width	Count	Width of high bitrate window measured as percent of total image width (0 to 65535)
ROI_x	Count	Horizontal start location of high bitrate window (0 to 65535 pixels)
ROI_y	Count	Vertical start location of high bitrate window (0 to 65535 pixels)

6.1.37.2 reportH264VideoEncodingCommandStatus

Description: This operation is used to report the status of the H264 encoding command.

Namespace: UMAA::SEM::H264VideoEncodingControl

Topic: H264VideoEncodingCommandStatus

Data Type: H264VideoEncodingCommandStatusType

Table 112: H264VideoEncodingCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.37.3 setH264VideoEncoding

Description: This operation is used to set the configuration of the H264 encoding of a video sensor.

Namespace: UMAA::SEM::H264VideoEncodingControl

Topic: H264VideoEncodingCommand

Data Type: H264VideoEncodingCommandType

Table 113: H264VideoEncodingCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
gradualDecoderRefresh	BooleanEnumType	The on/off value of Gradual Decoder Refresh, also called Periodic Intra Refresh.
groupOfPictures	Count	Number of interim frames before sending a key frame
preset	H264PresetEnumType	Preconfigured settings for hardware specific H264 settings.
profile	H264EncodingEnumType	H264 profile setting.
regionOfInterest	BooleanEnumType	The on/off value of region of interest (foveation) encoding, if supported.
ROI_height	Count	Height of high bitrate window measured as a percent of total image height (0 to 65535)
ROI_highBitRate	CommsRate_MegabitsPerSecond	Bitrate of ROI window, measured in Mbps (0 to 65535)
ROI_lowBitRate	CommsRate_MegabitsPerSecond	Bitrate of remaining image, measured in Mbps (0 to 65535)
ROI_width	Count	Width of high bitrate window measured as percent of total image width (0 to 65535)
ROI_x	Count	Horizontal start location of high bitrate window (0 to 65535 pixels)
ROI_y	Count	Vertical start location of high bitrate window (0 to 65535 pixels)

6.1.38 H264VideoEncodingSpecs

The purpose of this service is to provide a mechanism for querying the H264 capabilities the video sensors.

Table 114: H264VideoEncodingSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryH264VideoEncodingSpecs⊕	reportH264VideoEncodingSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.38.1 reportH264VideoEncodingSpecs

Description: This operation is used to report the H264 capabilities of a video sensor.

Namespace: UMAA::SEM::H264VideoEncodingSpecs

Topic: H264VideoEncodingSpecsReport

Data Type: H264VideoEncodingSpecsReportType

Table 115: H264VideoEncodingSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
supportedBaseline	BooleanEnumType	The supported/unsupported H264 profile
supportedBestQuality	BooleanEnumType	The supported/unsupported H264 preset
supportedCAVLC444	BooleanEnumType	The supported/unsupported H264 profile
supportedConstrainedBaseline	BooleanEnumType	The supported/unsupported H264 profile
supportedConstrainedHigh	BooleanEnumType	The supported/unsupported H264 profile
supportedDriveVision	BooleanEnumType	The supported/unsupported H264 preset
supportedExtended	BooleanEnumType	The supported/unsupported H264 profile
supportedGradualDecoderRefresh	BooleanEnumType	The indication of support for Gradual Decoder Refresh capability
supportedGroupOfPictures	BooleanEnumType	The indication of support for Group of Pictures capability
supportedHigh	BooleanEnumType	The supported/unsupported H264 profile
supportedHigh10	BooleanEnumType	The supported/unsupported H264 profile
supportedHigh10Intra	BooleanEnumType	The supported/unsupported H264 profile
supportedHigh422	BooleanEnumType	The supported/unsupported H264 profile
supportedHigh422Intra	BooleanEnumType	The supported/unsupported H264 profile
supportedHigh444Intra	BooleanEnumType	The supported/unsupported H264 profile
supportedHigh444Predictive	BooleanEnumType	The supported/unsupported H264 profile
supportedLevel1	BooleanEnumType	The supported/unsupported H264 level

Attribute Name	Attribute Type	Attribute Description
supportedLevel11	BooleanEnumType	The supported/unsupported H264 level
supportedLevel12	BooleanEnumType	The supported/unsupported H264 level
supportedLevel13	BooleanEnumType	The supported/unsupported H264 level
supportedLevel1B	BooleanEnumType	The supported/unsupported H264 level
supportedLevel2	BooleanEnumType	The supported/unsupported H264 level
supportedLevel21	BooleanEnumType	The supported/unsupported H264 level
supportedLevel22	BooleanEnumType	The supported/unsupported H264 level
supportedLevel3	BooleanEnumType	The supported/unsupported H264 level
supportedLevel31	BooleanEnumType	The supported/unsupported H264 level
supportedLevel32	BooleanEnumType	The supported/unsupported H264 level
supportedLevel4	BooleanEnumType	The supported/unsupported H264 level
supportedLevel41	BooleanEnumType	The supported/unsupported H264 level
supportedLevel42	BooleanEnumType	The supported/unsupported H264 level
supportedLevel5	BooleanEnumType	The supported/unsupported H264 level
supportedLevel51	BooleanEnumType	The supported/unsupported H264 level
supportedLevel52	BooleanEnumType	The supported/unsupported H264 level
supportedLowLatency	BooleanEnumType	The supported/unsupported H264 preset
supportedMain	BooleanEnumType	The supported/unsupported H264 profile
supportedManipulation	BooleanEnumType	The supported/unsupported H264 preset
supportedMultiviewHigh	BooleanEnumType	The supported/unsupported H264 profile
supportedPersistentStare	BooleanEnumType	The supported/unsupported H264 preset
supportedProgramSpecific1	BooleanEnumType	The supported/unsupported H264 preset
supportedProgramSpecific2	BooleanEnumType	The supported/unsupported H264 preset
supportedProgramSpecific3	BooleanEnumType	The supported/unsupported H264 preset
supportedProgramSpecific4	BooleanEnumType	The supported/unsupported H264 preset
supportedProgressiveHigh	BooleanEnumType	The supported/unsupported H264 profile
supportedRegionOfInterest	BooleanEnumType	The indication of support for Region of Interest (foveation) capability
supportedScalableBaseline	BooleanEnumType	The supported/unsupported H264 profile
supportedScalableConstrainedBaseline	BooleanEnumType	The supported/unsupported H264 profile
supportedScalableConstrainedHigh	BooleanEnumType	The supported/unsupported H264 profile
supportedScalableHigh	BooleanEnumType	The supported/unsupported H264 profile
supportedScalableHighIntra	BooleanEnumType	The supported/unsupported H264 profile
supportedSlowComms	BooleanEnumType	The supported/unsupported H264 preset
supportedStereoHigh	BooleanEnumType	The supported/unsupported H264 profile

6.1.39 HeartbeatPulseStatus

The purpose of this service is to provide a means to maintain the periodic communication connection with the unmanned platform.

Table 116: HeartbeatPulseStatus Operations

Service Requests (Inputs)	**Service Responses (Outputs)**
queryHeartbeatPulse⊕	reportHeartbeatPulse

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.39.1 reportHeartbeatPulse

Description: This operation is used to report the heartbeat pulse status of the unmanned platform.

Namespace: UMAA::SO::HeartbeatPulseStatus

Topic: HeartbeatPulseReport

Data Type: HeartbeatPulseReportType

Table 117: HeartbeatPulseReportType Message Definition

Attribute Name	**Attribute Type**	**Attribute Description**
Additional fields included from UMAA::UMAAStatus		
heartBeat	Count	The current heartbeat pulse to report the HeartbeatPulse connection of the unmanned platform

6.1.40 ManagementState

The purpose of this service is to provide the control and report of the system or subsystems life-cycle.

Table 118: ManagementState Operations

Service Requests (Inputs)	**Service Responses (Outputs)**
setEmergencyState	reportEmergencyStateCommandStatus
queryEmergencyStateCommand⊕	reportEmergencyStateCommand
cancelEmergencyStateCommand⊕	reportEmergencyStateCancelCommandStatus⊕
setManagementState	reportManagementStateCommandStatus
queryManagementStateCommand⊕	reportManagementStateCommand
cancelManagementStateCommand⊕	reportManagementStateCancelCommandStatus⊕
queryManagementState⊕	reportManagementState

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.40.1 reportEmergencyStateCommand

Description: This operation is used to report the commanded emergency state of the system or subsystem life-cycle.

Namespace: UMAA::SO::ManagementState

Topic: EmergencyStateCommandReport

Data Type: EmergencyStateCommandReportType

Table 119: EmergencyStateCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
emergencyState	BooleanEnumType	An alert safety critical condition

6.1.40.2 reportEmergencyStateCommandStatus

Description: This operation is used to report the status of the emergency state of the system or subsystem life-cycle command.

Namespace: UMAA::SO::ManagementState

Topic: EmergencyStateCommandStatus

Data Type: EmergencyStateCommandStatusType

Table 120: EmergencyStateCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.40.3 reportManagementState

Description: This operation is used to report the current state of the system or subsystem life-cycle.

Namespace: UMAA::SO::ManagementState

Topic: ManagementStateReport

Data Type: ManagementStateReportType

Table 121: ManagementStateReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
state	CoreStateEnumType	The current state of the system or subsystem life-cycle

6.1.40.4 reportManagementStateCommand

Description: This operation is used to report the commanded state of the system or subsystem life-cycle.

Namespace: UMAA::SO::ManagementState

Topic: ManagementStateCommandReport

Data Type: ManagementStateCommandReportType

Table 122: ManagementStateCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
state	CoreStateEnumType	A desired state (shutdown, standby, resume, reset, etc.)

6.1.40.5 reportManagementStateCommandStatus

Description: This operation is used to report the status of the state of the system or subsystem life-cycle command.

Namespace: UMAA::SO::ManagementState

Topic: ManagementStateCommandStatus

Data Type: ManagementStateCommandStatusType

Table 123: ManagementStateCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.40.6 setEmergencyState

Description: This operation is used to alert a safety critical condition of the system or subsystem life-cycle.

Namespace: UMAA::SO::ManagementState

Topic: EmergencyStateCommand

Data Type: EmergencyStateCommandType

Table 124: EmergencyStateCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from	UMAA::UMAACommand
emergencyState	BooleanEnumType	An alert safety critical condition

6.1.40.7 setManagementState

Description: This operation is used to set the desired state of the system or subsystem life-cycle.

Namespace: UMAA::SO::ManagementState

Topic: ManagementStateCommand

Data Type: ManagementStateCommandType

Table 125: ManagementStateCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from	UMAA::UMAACommand
state	CoreStateEnumType	A desired state (shutdown, standby, resume, reset, etc.)

6.1.41 PanTiltJointEffortControl

The purpose of this service is to provide control of the low level pant tilt mechanism.

Table 126: PanTiltJointEffortControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setPanTiltJointEffort	reportPanTiltJointEffortCommandStatus
queryPanTiltJointEffortCommandAck⊕	reportPanTiltJointEffortCommandAck
cancelPanTiltJointEffortCommand⊕	reportPanTiltJointEffortCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.41.1 reportPanTiltJointEffortCommandAck

Description: This operation is used to report the percent effort that is currently being applied to the two joints of the pan tilt mechanism.

Namespace: UMAA::SEM::PanTiltJointEffortControl

Topic: PanTiltJointEffortCommandAckReport

Data Type: PanTiltJointEffortCommandAckReportType

Table 127: PanTiltJointEffortCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
panEffort	Effort	The desired joint pan effort in percent of the pan mechanism (-100 to 100)
tiltEffort	Effort	The desired joint tilt effort in percent of the tilt mechanism (-100 to 100)

6.1.41.2 reportPanTiltJointEffortCommandStatus

Description: This operation is used to report the status of the joint effort command.

Namespace: UMAA::SEM::PanTiltJointEffortControl

Topic: PanTiltJointEffortCommandStatus

Data Type: PanTiltJointEffortCommandStatusType

Table 128: PanTiltJointEffortCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.41.3 setPanTiltJointEffort

Description: This operation is used to control the two joint actuators for a pan tilt mechanism. The consumer must perform a "cancel" of the command to initiate the end of command execution as this command has no determinate end of execution.

Namespace: UMAA::SEM::PanTiltJointEffortControl

Topic: PanTiltJointEffortCommand

Data Type: PanTiltJointEffortCommandType

Table 129: PanTiltJointEffortCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
panEffort	Effort	The desired joint pan effort in percent of the pan mechanism (-100 to 100)
tiltEffort	Effort	The desired joint tilt effort in percent of the tilt mechanism (-100 to 100)

6.1.42 PanTiltJointEffortStatus

The purpose of this service is to provide the status of the low level pant tilt mechanism.

Table 130: PanTiltJointEffortStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPanTiltJointEffort⊕	reportPanTiltJointEffort

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.42.1 reportPanTiltJointEffort

Description: This operation is used to report the current the percent effort that is currently being applied to the two joints of the pan tilt mechanism.

Namespace: UMAA::SEM::PanTiltJointEffortStatus

Topic: PanTiltJointEffortReport

Data Type: PanTiltJointEffortReportType

Table 131: PanTiltJointEffortReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
panEffort	Effort	The current percentage level of pan effort of the pan mechanism (-100 to 100)
tiltEffort	Effort	The current percentage level of tilt effort of the tilt mechanism (-100 to 100)

6.1.43 PanTiltJointPositionControl

The purpose of this service is to provide the closed-loop joint position control.

Table 132: PanTiltJointPositionControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setPanTiltJointPosition	reportPanTiltJointPositionCommandStatus
queryPanTiltJointPositionCommandAck⊕	reportPanTiltJointPositionCommandAck
cancelPanTiltJointPositionCommand⊕	reportPanTiltJointPositionCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.43.1 reportPanTiltJointPositionCommandAck

Description: This operation is used to report the commanded joint position for a pan tilt mechanism.

Namespace: UMAA::SEM::PanTiltJointPositionControl

Topic: PanTiltJointPositionCommandAckReport

Data Type: PanTiltJointPositionCommandAckReportType

Table 133: PanTiltJointPositionCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
pan	Angle	The desired joint pan position (-8 pi to +8 pi).
tilt	Angle	The desired joint tilt position (-8 pi to +8 pi).

6.1.43.2 reportPanTiltJointPositionCommandStatus

Description: This operation is used to report the pan tilt join position command.

Namespace: UMAA::SEM::PanTiltJointPositionControl

Topic: PanTiltJointPositionCommandStatus

Data Type: PanTiltJointPositionCommandStatusType

Table 134: PanTiltJointPositionCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.43.3 setPanTiltJointPosition

Description: This operation is used to set the desired joint position for a pan tilt mechanism.

Namespace: UMAA::SEM::PanTiltJointPositionControl

Topic: PanTiltJointPositionCommand

Data Type: PanTiltJointPositionCommandType

Table 135: PanTiltJointPositionCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
pan	Angle	The desired joint pan position (-8 pi to +8 pi).
tilt	Angle	The desired joint tilt position (-8 pi to +8 pi).

6.1.44 PanTiltJointPositionStatus

The purpose of this service is to report the two joint angles of the pan tilt mechanism.

Table 136: PanTiltJointPositionStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPanTiltJointPositionState⊕	reportPanTiltJointPositionState

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.44.1 reportPanTiltJointPositionState

Description: This operation is used to report the current position of the two joint angles of the pan tilt mechanism.

Namespace: UMAA::SEM::PanTiltJointPositionStatus

Topic: PanTiltJointPositionStateReport

Data Type: PanTiltJointPositionStateReportType

Table 137: PanTiltJointPositionStateReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
pan	Angle	The current pan position (-8pi to +8pi)
tilt	Angle	The current tilt distance (-8pi to +8pi)

6.1.45 PanTiltJointVelocityControl

The purpose of this service is to provide a closed-loop joint velocity control.

Table 138: PanTiltJointVelocityControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setPanTiltJointVelocity	reportPanTiltJointVelocityCommandStatus
queryPanTiltJointVelocityCommandAck⊕	reportPanTiltJointVelocityCommandAck
cancelPanTiltJointVelocityCommand⊕	reportPanTiltJointVelocityCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.45.1 reportPanTiltJointVelocityCommandAck

Description: This operation is used to report the commanded joint velocity values for a pan tilt mechanism.

Namespace: UMAA::SEM::PanTiltJointVelocityControl

Topic: PanTiltJointVelocityCommandAckReport

Data Type: PanTiltJointVelocityCommandAckReportType

Table 139: PanTiltJointVelocityCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
panVelocity	AngleRate	The current joint velocity of a pan mechanism. (-10 pi to +10 pi radian per second)
tiltVelocity	AngleRate	The current joint velocity of a tilt mechanism. (-10 pi to +10 pi radian per second)

6.1.45.2 reportPanTiltJointVelocityCommandStatus

Description: This operation is used to report the status of the joint velocity command.

Namespace: UMAA::SEM::PanTiltJointVelocityControl

Topic: PanTiltJointVelocityCommandStatus

Data Type: PanTiltJointVelocityCommandStatusType

Table 140: PanTiltJointVelocityCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.45.3 setPanTiltJointVelocity

Description: This operation is used to control the joint velocity for a pan tilt mechanism. The consumer must perform a "cancel" of the command to initiate the end of command execution as this command has no determinate end of execution.

Namespace: UMAA::SEM::PanTiltJointVelocityControl

Topic: PanTiltJointVelocityCommand

Data Type: PanTiltJointVelocityCommandType

Table 141: PanTiltJointVelocityCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
panVelocity	AngleRate	The current joint velocity of a pan mechanism. (-10 pi to +10 pi radian per second)
tiltVelocity	AngleRate	The current joint velocity of a tilt mechanism. (-10 pi to +10 pi radian per second)

6.1.46 PanTiltJointVelocityStatus

The purpose of this service is to report the velocity of the two joint angles of the pan tilt mechanism.

Table 142: PanTiltJointVelocityStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPanTiltJointVelocityState⊕	reportPanTiltJointVelocityState

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.46.1 reportPanTiltJointVelocityState

Description: This operation is used to report the current joint velocity values for a pan tilt mechanism.

Namespace: UMAA::SEM::PanTiltJointVelocityStatus

Topic: PanTiltJointVelocityReport

Data Type: PanTiltJointVelocityReportType

Table 143: PanTiltJointVelocityReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
panVelocity	AngleRate	The current pan velocity of a pan tilt mechanism. (-10 pi to +10 pi radian per second)
tiltVelocity	AngleRate	The current tilt rate of a pan tilt mechanism. (-10 pi to +10 pi radian per second)

6.1.47 PanTiltSpecs

The purpose of this service is to report the physical specifications of a pan tilt unit on board of the unmanned platform.

Table 144: PanTiltSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPanTiltSpecs⊕	reportPanTiltSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.47.1 reportPanTiltSpecs

Description: This operation is used report the physical specifications of the pan tilt unit on board of the unmanned platform.

Namespace: UMAA::SEM::PanTiltSpecs

Topic: PanTiltSpecsReport

Data Type: PanTiltSpecsReportType

Table 145: PanTiltSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
baseOffset	Position3D_PlatformXYZ	The pan tilt coordinate system (X, Y, Z) measured with respect to vehicle coordinate system (-30m to +30m).
baseOrientation	OrientationQuaternion	The orientation of the pan tilt coordinate system measured with respect to vehicle coordinate system (-1 to +1).
panMax	Angle	The maximum pan angle (-8 pi to +8 pi)
panMaxSpeed	AngleRate	The maximum pan speed (0 to +10 pi radian per second).
panMin	Angle	The minimum pan angle (-8 pi to +8 pi)
tiltMax	Angle	The maximum tilt distance (-8 pi to +8 pi)
tiltMaxSpeed	AngleRate	The maximum tilt speed (0 to +10 pi radian per second)

Attribute Name	Attribute Type	Attribute Description
tiltMin	Angle	The minimum tilt distance (-8 pi to +8 pi)

6.1.48 PlatformModeControl

The purpose of this service is to provide a mechanism to manage the unmanned platform's mode of operation.

Table 146: PlatformModeControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setPlatformMode	reportPlatformModeCommandStatus
queryPlatformModeCommandAck⊕	reportPlatformModeCommandAck
cancelPlatformModeCommand⊕	reportPlatformModeCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.48.1 reportPlatformModeCommandAck

Description: This operation is used to report the current platform mode command of the vehicle.

Namespace: UMAA::SO::PlatformModeControl

Topic: PlatformModeCommandAckReport

Data Type: PlatformModeCommandAckReportType

Table 147: PlatformModeCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
mode	PlatformModeEnumType	Platform Mode.

6.1.48.2 reportPlatformModeCommandStatus

Description: This operation is used to report the status of the platform mode command of the vehicle.

Namespace: UMAA::SO::PlatformModeControl

Topic: PlatformModeCommandStatus

Data Type: PlatformModeCommandStatusType

Table 148: PlatformModeCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.48.3 setPlatformMode

Description: This operation is used to set the platform mode of a vehicle.

Namespace: UMAA::SO::PlatformModeControl

Topic: PlatformModeCommand

Data Type: PlatformModeCommandType

Table 149: PlatformModeCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
mode	PlatformModeEnumType	Platform Mode.

6.1.49 PlatformModeSpecs

The purpose of this service is to provide a mechanism to report the modes of operation the unmanned platform supports.

Table 150: PlatformModeSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPlatformModeSpecs⊕	reportPlatformModeSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.49.1 reportPlatformModeSpecs

Description: This operation is used to report the platform modes supported by the vehicle.

Namespace: UMAA::SO::PlatformModeSpecs

Topic: PlatformModeSpecsReport

Data Type: PlatformModeSpecsReportType

Table 151: PlatformModeSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
supportedMaintenance	BooleanEnumType	The supported/unsupported platform mode
supportedStandardOperating	BooleanEnumType	The supported/unsupported platform mode
supportedTraining	BooleanEnumType	The supported/unsupported platform mode

6.1.50 PlatformModeStatus

The purpose of this service is to provide a mechanism to report the unmanned platform's mode of operation.

Table 152: PlatformModeStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPlatformMode⊕	reportPlatformMode

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.50.1 reportPlatformMode

Description: This operation is used to report the platform mode of the vehicle.

Namespace: UMAA::SO::PlatformModeStatus

Topic: PlatformModeReport

Data Type: PlatformModeReportType

Table 153: PlatformModeReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
mode	PlatformModeEnumType	Platform Mode.
status	PlatformModeTransitionEnumType	Status of platform mode transition.

6.1.51 RenderUselessControl

The purpose of this service is to provide a mechanism to destroy, disable, or make the system and/or subsystems of the unmanned platform less valuable if the unmanned platform were to be captured by an adversary. Depending on the implementation, some or all operations in this service may not be sent successfully.

Table 154: RenderUselessControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setRenderUseless	reportRenderUselessCommandStatus
cancelRenderUselessCommand⊕	reportRenderUselessCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.51.1 reportRenderUselessCommandStatus

Description: This operation is used to report the status of the render-useless command.

Namespace: UMAA::SO::RenderUselessControl

Topic: RenderUselessCommandStatus

Data Type: RenderUselessCommandStatusType

Table 155: RenderUselessCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.51.2 setRenderUseless

Description: This operation is used to render the target system or subsystem inoperable.

Namespace: UMAA::SO::RenderUselessControl

Topic: RenderUselessCommand

Data Type: RenderUselessCommandType

Table 156: RenderUselessCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		

6.1.52 RenderUselessStatus

The purpose of this service is to provide a mechanism to report whether the system and/or subsystems of the unmanned platform have been intentionally rendered useless.

Table 157: RenderUselessStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRenderUseless⊕	reportRenderUseless

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.52.1 reportRenderUseless

Description: This operation is used to report if a target has been rendered inoperable.

Namespace: UMAA::SO::RenderUselessStatus

Topic: RenderUselessReport

Data Type: RenderUselessReportType

Table 158: RenderUselessReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
errorString	StringShortDescription	Description associated with failed render-useless target system or subsystem. No description will be filled if the state is Normal or RenderedUseless.
state	RenderUselessStateEnumType	A render-useless state of the target system or subsystem

6.1.53 ResourceIdentification

The purpose of this service is to report the information of the unmanned platform and its subsystems.

Table 159: ResourceIdentification Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryResourceAuthorization⊕	reportResourceAuthorization
querySubsystemID⊕	reportSubsystemID
queryVehicleID⊕	reportVehicleID

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.53.1 reportResourceAuthorization

Description: This operation is used to report a list of levels of authorization of the system or subsystem(s).

Namespace: UMAA::MM::ResourceIdentification

Topic: ResourceAuthorizationReport

Data Type: ResourceAuthorizationReportType

Table 160: ResourceAuthorizationReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
authorizationLevel*	SpecificLOIEnumType	A list of authorized control stations

6.1.53.2 reportSubsystemID

Description: This operation is used to report the information of the subsystem(s) on-board or off-board an unmanned platform.

Namespace: UMAA::MM::ResourceIdentification

Topic: SubsystemIDReport

Data Type: SubsystemIDReportType

Table 161: SubsystemIDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
isControlTransferCapable	BooleanEnumType	An subsystem or a subset of a system can be transferred control between control systems.
name	StringShortDescription	A name to describe a payload or a subsystem.
type	StringShortDescription	A name to describe the type of payload or subsystem (e.g. cameras, sonar, batteries, GPS, etc.).

6.1.53.3 reportVehicleID

Description: This operation is used to report the information of an unmanned platform or a system.

Namespace: UMAA::MM::ResourceIdentification

Topic: VehicleIDReport

Data Type: VehicleIDReportType

Table 162: VehicleIDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
domain	DomainEnumType	The domain that the unmanned platform is operated under.
isControlTransferCapable	BooleanEnumType	An unmanned platform or a system can be transferred control between control systems.
make	StringShortDescription	The manufacture of the unmanned platform.
model	StringShortDescription	The model of the unmanned platform.
name	StringShortDescription	The name of the unmanned platform or a system.
protocol	StringShortDescription	The protocol used to communicate to the unmanned platform.
type	StringShortDescription	The type of the unmanned platform or a system.
vehicleNumber	NumericGUID	a unique number which specifies the string designation for unmanned platform. It's a tail number for air vehicle, hull number for maritime vehicle, or registration number for ground vehicle.

6.1.54 SelfCollisionAvoidanceConfig

The purpose of this service is to provide a mechanism to configure avoidance behaviors on the unmanned platform.

Table 163: SelfCollisionAvoidanceConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySelfCollisionAvoidanceConfig⊕	reportSelfCollisionAvoidanceConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.54.1 reportSelfCollisionAvoidanceConfig

Description: This operation is used to report the current configuration of the self-collision avoidance of the unmanned platform.

Namespace: UMAA::MO::SelfCollisionAvoidanceConfig

Topic: SelfCollisionAvoidanceConfigReport

Data Type: SelfCollisionAvoidanceConfigReportType

Table 164: SelfCollisionAvoidanceConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		

Attribute Name	Attribute Type	Attribute Description
priority	Count	The relative priority used to establish precedence when multiple moving objects support self-collision avoidance. Lower values should yield to objects with higher values. Fixed objects, or moving objects that do not support self-collision avoidance, are considered to have maximum priority (255). If two objects have the same priority, the behavior is non-deterministic.
state	CollisionAvoidStateEnumType	An currently configured behavior for self-collision avoidance

6.1.55 SelfCollisionAvoidanceControl

The purpose of this service is to provide a mechanism to control supported self-collision avoidance capabilities on the unmanned platform.

Table 165: SelfCollisionAvoidanceControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setSelfCollisionAvoidance	reportSelfCollisionAvoidanceCommandStatus
querySelfCollisionAvoidanceCommandAck⊕	reportSelfCollisionAvoidanceCommandAck
cancelSelfCollisionAvoidanceCommand⊕	reportSelfCollisionAvoidanceCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.55.1 reportSelfCollisionAvoidanceCommandAck

Description: This operation is used to report the current self-collision avoidance command.

Namespace: UMAA::MO::SelfCollisionAvoidanceControl

Topic: SelfCollisionAvoidanceCommandAckReport

Data Type: SelfCollisionAvoidanceCommandAckReportType

Table 166: SelfCollisionAvoidanceCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
priority	Count	The relative priority used to establish precedence when multiple moving objects support self-collision avoidance. Lower values should yield to objects with higher values. Fixed objects, or moving objects that do not support self-collision avoidance, are considered to have maximum priority (255). If two objects have the same priority, the behavior is non-deterministic.
state	CollisionAvoidStateEnumType	An active behavior for self-collision avoidance

6.1.55.2 reportSelfCollisionAvoidanceCommandStatus

Description: This operation is used to report a response whether the unmanned platform supports the self-collision avoidance.

Namespace: UMAA::MO::SelfCollisionAvoidanceControl

Topic: SelfCollisionAvoidanceCommandStatus

Data Type: SelfCollisionAvoidanceCommandStatusType

Table 167: SelfCollisionAvoidanceCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.55.3 setSelfCollisionAvoidance

Description: This operation is used to set the active behavior for self-collision avoidance of the unmanned platform.

Namespace: UMAA::MO::SelfCollisionAvoidanceControl

Topic: SelfCollisionAvoidanceCommand

Data Type: SelfCollisionAvoidanceCommandType

Table 168: SelfCollisionAvoidanceCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
priority	Count	The relative priority used to establish precedence when multiple moving objects support self-collision avoidance. Lower values should yield to objects with higher values. Fixed objects, or moving objects that do not support self-collision avoidance, are considered to have maximum priority (255). If two objects have the same priority, the behavior is non-deterministic.
state	CollisionAvoidStateEnumType	An active behavior for self-collision avoidance

6.1.56 SelfCollisionAvoidanceSpecs

The purpose of this service is to provide a mechanism to determine supported self-collision avoidance specifications on the unmanned platform.

Table 169: SelfCollisionAvoidanceSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySelfCollisionAvoidanceSpecs⊕	reportSelfCollisionAvoidanceSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.56.1 reportSelfCollisionAvoidanceSpecs

Description: This operation is used to report the capabilities of the self-collision avoidance of the unmanned platform.

Namespace: UMAA::MO::SelfCollisionAvoidanceSpecs

Topic: SelfCollisionAvoidanceSpecsReport

Data Type: SelfCollisionAvoidanceSpecsReportType

Table 170: SelfCollisionAvoidanceSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
avoidObstacle	BooleanEnumType	One (1) means the unmanned platform support avoid-obstacle behavior
stopOnObstacle	BooleanEnumType	One (1) means the unmanned platform support stop-on-obstacle behavior

6.1.57 SelfCollisionAvoidanceStatus

The purpose of this service is to provide a mechanism to solicit status information on the unmanned platform.

Table 171: SelfCollisionAvoidanceStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySelfCollisionAvoidance⊕	reportSelfCollisionAvoidance

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.57.1 reportSelfCollisionAvoidance

Description: This operation is used to report the current status of the self-collision avoidance of the unmanned platform.

Namespace: UMAA::MO::SelfCollisionAvoidanceStatus

Topic: SelfCollisionAvoidanceReport

Data Type: SelfCollisionAvoidanceReportType

Table 172: SelfCollisionAvoidanceReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
status	CollisionAvoidStatusEnumType	A current status of the self-collision avoidance

6.1.58 SoftwareVersionStatus

The purpose of this service is to provide a mechanism to report the software versions of the system or its subsystem.

Table 173: SoftwareVersionStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySoftwareVersion⊕	reportSoftwareVersion

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.58.1 reportSoftwareVersion

Description: This operation is used to report the software version of a specified system or the subsystem.

Namespace: UMAA::SO::SoftwareVersionStatus

Topic: SoftwareVersionReport

Data Type: SoftwareVersionReportType

Table 174: SoftwareVersionReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
buildID	StringShortDescription	The build ID of this software unit
copyright	StringShortDescription	The copy write info for this software unit
name	NumericGUID	The name of this software unit
softwareDescription	StringShortDescription	A description of this software unit
version	StringShortDescription	The version of this software unit
softwareID*	NumericGUID	The ID of this software unit (could be an executable, script, shared library, etc.)

6.1.59 TamperDetectionControl

The purpose of this service is to provide a mechanism to enable or disable tamper detection.

Table 175: TamperDetectionControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setTamperDetection	reportTamperDetectionCommandStatus
queryTamperDetectionCommandAck⊕	reportTamperDetectionCommandAck
cancelTamperDetectionCommand⊕	reportTamperDetectionCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.59.1 reportTamperDetectionCommandAck

Description: This operation is used to report the current tamper detection command.

Namespace: UMAA::SO::TamperDetectionControl

Topic: TamperDetectionCommandAckReport

Data Type: TamperDetectionCommandAckReportType

Table 176: TamperDetectionCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
state	TamperDetectionStateEnumType	A desired state of tamper detection

6.1.59.2 reportTamperDetectionCommandStatus

Description: This operation is used to report the status of the tamper detection command.

Namespace: UMAA::SO::TamperDetectionControl

Topic: TamperDetectionCommandStatus

Data Type: TamperDetectionCommandStatusType

Table 177: TamperDetectionCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.59.3 setTamperDetection

Description: This operation is used to enable or disable tamper detection and to clear the previously reported activities.

Namespace: UMAA::SO::TamperDetectionControl

Topic: TamperDetectionCommand

Data Type: TamperDetectionCommandType

Table 178: TamperDetectionCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
state	TamperDetectionStateEnumType	A desired state of tamper detection

6.1.60 TamperDetectionStatus

The purpose of this service is to provide a mechanism to report when tampering has occurred.

Table 179: TamperDetectionStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryTamperDetection⊕	reportTamperDetection

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.60.1 reportTamperDetection

Description: This operation is used to report the current status of anti-tamper detection. Detected activities are continuously reported until cleared by a setTamperDetectionState message.

Namespace: UMAA::SO::TamperDetectionStatus

Topic: TamperDetectionReport

Data Type: TamperDetectionReportType

Table 180: TamperDetectionReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAAStatus	
descriptor	StringShortDescription	Description of tamper detection activities
electricalTamper	BooleanEnumType	Current result of electrical tamper detection.

Attribute Name	Attribute Type	Attribute Description
hardwareTamper	BooleanEnumType	Current result of hardware tamper detection.
networkInstrusion	BooleanEnumType	Current result of network tamper detection.
otherTamper	BooleanEnumType	Current result of tamper detection.
state	TamperDetectionStateEnumType	The current state of tamper detection.

6.1.61 VideoIlluminatorConfig

The purpose of this service is to enable setting and reading the configuration of Illuminators associated with video sensors.

Table 181: VideoIlluminatorConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryVideoIlluminatorConfig⊕	reportVideoIlluminatorConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.61.1 reportVideoIlluminatorConfig

Description: This operation is used to report the configuration of a video illuminator.

Namespace: UMAA::SEM::VideoIlluminatorConfig

Topic: VideoIlluminatorConfigReport

Data Type: VideoIlluminatorConfigReportType

Table 182: VideoIlluminatorConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
associatedSensorIDs	sequence<NumericGUID>	List of video sensor IDs that are to be associated with this illuminator (this should be a sequence in the IDL)
mode	ImagingModeEnumType	The illumination mode - this enum already exists, I just don't know what it's called here - it contains: Color Greyscale Infrared Lowlight

6.1.62 VideoIlluminatorControl

The purpose of this service is to control Illuminators associated with video sensors. The service enables selection of an illumination mode, and setting the illumination intensity and the illuminator beam width for each illumination device.

Table 183: VideoIlluminatorControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setVideoIlluminator	reportVideoIlluminatorCommandStatus
queryVideoIlluminatorCommandAck⊕	reportVideoIlluminatorCommandAck
cancelVideoIlluminatorCommand⊕	reportVideoIlluminatorCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.62.1 reportVideoIlluminatorCommandAck

Description: This operation is used to report the current video illuminator command.

Namespace: UMAA::SEM::VideoIlluminatorControl

Topic: VideoIlluminatorCommandAckReport

Data Type: VideoIlluminatorCommandAckReportType

Table 184: VideoIlluminatorCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
beamWidth	Angle	The horizontal field of illumination (0 to 2pi radians)
level	Count	The intensity level as a percentage (0 to 100)

6.1.62.2 reportVideoIlluminatorCommandStatus

Description: This operation is used to report the status of a video illuminator command.

Namespace: UMAA::SEM::VideoIlluminatorControl

Topic: VideoIlluminatorCommandStatus

Data Type: VideoIlluminatorCommandStatusType

Table 185: VideoIlluminatorCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.62.3 setVideoIlluminator

Description: This operation is used to set the intensity and beam width of a video illuminator.

Namespace: UMAA::SEM::VideoIlluminatorControl

Topic: VideoIlluminatorCommand

Data Type: VideoIlluminatorCommandType

Table 186: VideoIlluminatorCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
beamWidth	Angle	The horizontal field of illumination (0 to 2pi radians)
level	Count	The intensity level as a percentage (0 to 100)

6.1.63 VideoIlluminatorSpecs

The purpose of this service is to report the capabilities of Illuminators associated with video sensors.

Table 187: VideoIlluminatorSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryVideoIlluminatorSpecs⊕	reportVideoIlluminatorSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.63.1 reportVideoIlluminatorSpecs

Description: This operation is used to report the capabilities of a video illuminator.

Namespace: UMAA::SEM::VideoIlluminatorSpecs

Topic: VideoIlluminatorSpecsReport

Data Type: VideoIlluminatorSpecsReportType

Table 188: VideoIlluminatorSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAAStatus	
maxBeamWidth	BooleanEnumType	The maximum horizontal field of illumination
name	StringShortDescription	The name of the video illuminator
supportedColor	BooleanEnumType	The supported/unsupported illuminator mode
supportedGreyscale	BooleanEnumType	The supported/unsupported illuminator mode
supportedInfrared	BooleanEnumType	The supported/unsupported illuminator mode

Attribute Name	Attribute Type	Attribute Description
supportedLowlight	BooleanEnumType	The supported/unsupported illuminator mode

6.1.64 VideoIlluminatorStatus

The purpose of this service is to report the intensity and beam width of Illuminators associated with video sensors.

Table 189: VideoIlluminatorStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryVideoIlluminator⊕	reportVideoIlluminator

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.64.1 reportVideoIlluminator

Description: This operation is used to report the intensity and beam width of a video illuminator.

Namespace: UMAA::SEM::VideoIlluminatorStatus

Topic: VideoIlluminatorReport

Data Type: VideoIlluminatorReportType

Table 190: VideoIlluminatorReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
beamWidth	Angle	The horizontal field of illumination (0 to 2pi radians)
level	Count	The intensity level as a percentage (0 to 100)

6.2 Common Data Types

Common data types define DDS types that are referenced throughout the UMAA model. These DDS types are considered common because they can be re-used as the data type for many attributes defined in service interface topics, interface topics, and other common data types. These data types are not intended to be directly published to/subscribed as DDS topics.

6.2.1 UCSMDEInterfaceSet

Namespace: UMAA::UCSMDEInterfaceSet

Description: Defines the common UCSMDE Interface Set Message Fields.

Table 191: UCSMDEInterfaceSet Structure Definition

Attribute Name	Attribute Type	Attribute Description
timeStamp	DateTime	The time at which the data was derived.

6.2.2 UMAACommand

Namespace: UMAA::UMAACommand

Description: Defines the common UMAA Command Message Fields.

Table 192: UMAACommand Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UCSMDEInterfaceSet		
source*	NumericGUID	The unique identifier of the originating source of the command interface.
destination*	NumericGUID	The unique identifier of the destination of the command interface.
sessionID*	NumericGUID	The identifier of the session.

6.2.3 UMAAStatus

Namespace: UMAA::UMAAStatus

Description: Defines the common UMAA Status Message Fields.

Table 193: UMAAStatus Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UCSMDEInterfaceSet		
source*	NumericGUID	The unique identifier of the originating source of the status interface.

6.2.4 UMAACommandStatusBase

Namespace: UMAA::UMAACommandStatusBase

Description: Defines the common UMAA Command Status Base Message Fields.

Table 194: UMAACommandStatusBase Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UCSMDEInterfaceSet		
source*	NumericGUID	The unique identifier of the originating source of the command status interface.
sessionID*	NumericGUID	The identifier of the session.

6.2.5 UMAACommandStatus

Namespace: UMAA::UMAACommandStatus

Description: Defines the common UMAA Command Status Message Fields.

Table 195: UMAACommandStatus Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
commandStatus	CommandStatusEnumType	The status of the command
commandStatusReason	CommandStatusReasonEnumType	The reason for the status of the command
logMessage	StringLongDescription	Human-readable description related to response. Systems should not parse or use any information from this for processing purposes.

6.2.6 DateTime

Namespace: UMAA::Measurement::DateTime

Description: Describes an absolute time. Conforms with POSIX time standard (IEEE Std 1003.1-2017) epoch reference point of January 1st, 1970 00:00:00 UTC.

Table 196: DateTime Structure Definition

Attribute Name	Attribute Type	Attribute Description
seconds	DateTimeSeconds	The number of seconds offset from the standard POSIX (IEEE Std 1003.1-2017) epoch reference point of January 1st, 1970 00:00:00 UTC.
nanoseconds	DateTimeNanoSeconds	The number of nanoseconds elapsed within the current DateTimeSecond

6.2.7 Altitude_AGL

Namespace: UMAA::Common::Measurement::Altitude_AGL

Description: Altitude_AGL specifies the entity's height above terrain, as reported by a radar system.

Table 197: Altitude_AGL Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	RadarHeight	Specifies the entity's height above terrain, as reported by a radar system.

6.2.8 Altitude_MSL

Namespace: UMAA::Common::Measurement::Altitude_MSL

Description: Altitude_MSL specifies the entity's height above the geoid.

Table 198: Altitude_MSL Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	MSLHeight	Specifies the entity's height above the geoid.

6.2.9 AnalogSensorErrorType

Namespace: UMAA::SEM::SensorManagement::AnalogSensorErrorType

Description: This structure is used to report error code when the specified analog sensor configuration is set to be invalid.

Table 199: AnalogSensorErrorType Structure Definition

Attribute Name	Attribute Type	Attribute Description
errorCode	AnalogSensorErrorCodeEnumType	Error code reports when the specified analog sensor configuration is set to be invalid.
errorMessage	StringShortDescription	A description of the invalid analog sensor configuration setting

6.2.10 BuiltInTestStatusType_All

Namespace: UMAA::Common::Measurement::BuiltInTestStatusType_All

Description: Realizes BuiltInTestStatusType: the Selector for the BuiltInTestStatusEnumType.

Table 200: BuiltInTestStatusType_All Structure Definition

Attribute Name	Attribute Type	Attribute Description
domain	sequence<BuiltInTestStatusEnumType>	An EnumerationSet which defines the allowable values for the Selector.
setPoint	BuiltInTestStatusEnumType	An EnumerationSet which specifies the desired value of the Selector.
value	BuiltInTestStatusEnumType	An EnumerationSet which specifies the actual value of the Selector.

6.2.11 DigitalSensorErrorType

Namespace: UMAA::SEM::SensorManagement::DigitalSensorErrorType

Description: This structure is used to report error code when the specified digital sensor configuration is set to be invalid.

Table 201: DigitalSensorErrorType Structure Definition

Attribute Name	Attribute Type	Attribute Description
errorCode	DigitalSensorErrorCodeEnumType	Error code reports when the specified digital sensor configuration is set to be invalid.
errorMessage	StringShortDescription	A description of the invalid digital sensor configuration setting

6.2.12 GeodeticLatitude

Namespace: UMAA::Common::Measurement::GeodeticLatitude

Description: GeodeticLatitude specifies the angle between the normal and the equatorial plane of the ellipsoid. The Latitude specifies the north-south position of a point.

Table 202: GeodeticLatitude Structure Definition

Attribute Name	Attribute Type	Attribute Description
latitude	GeodeticLatitude	GeodeticLatitude specifies the angle between the normal and the equatorial plane of the ellipsoid. The Latitude specifies the north-south position of a point.

6.2.13 GeodeticLongitude

Namespace: UMAA::Common::Measurement::GeodeticLongitude

Description: GeodeticLongitude specifies the angular measurement of a location east or west of the prime meridian of the reference ellipsoid.

Table 203: GeodeticLongitude Structure Definition

Attribute Name	Attribute Type	Attribute Description
longitude	GeodeticLongitude	GeodeticLongitude specifies the angular measurement of a location east or west of the prime meridian of the reference ellipsoid.

6.2.14 JointOperationalParamsType

Namespace: UMAA::SEM::ExtendedPayloadStatus::JointOperationalParamsType

Description: This structure is used to report the operational parameters of a single manipulator joint.

Table 204: JointOperationalParamsType Structure Definition

Attribute Name	Attribute Type	Attribute Description
params	OperationalParamsType	The operational parameters of this joint
jointID*	NumericGUID	Unique Identifier of the manipulator joint (within the manipulator)

6.2.15 MoveToPosPolicyType

Namespace: UMAA::MM::CommsLostPolicy::MoveToPosPolicyType

Description: This structure is used to report the move-to-position policy in case of lost communications of the unmanned platform.

Table 205: MoveToPosPolicyType Structure Definition

Attribute Name	Attribute Type	Attribute Description
distance	Distance	Travel distance while trying to regain comms
mode	VehicleSpeedModeEnumType	A speed mode.
position	WaypointType	Travel to a waypoint position while trying to regain comms
speed	Speed	A speed of the unmanned platform with reference to the medium, the ground, the air, or true speed.
speedReference	VehicleSpeedReferenceEnumType	A speed reference.

6.2.16 OperationalParamsType

Namespace: UMAA::SEM::ExtendedPayloadStatus::OperationalParamsType

Description: This structure is used to report operational parameters of a subsystem.

Table 206: OperationalParamsType Structure Definition

Attribute Name	Attribute Type	Attribute Description
busCurrent	PowerBusCurrent	Bus current of a subsystem
busVoltage	PowerBusVoltage	Bus voltage of a subsystem
phaseCurrent	PowerBusCurrent	Phase current of a subsystem
temp	Temperature	Subsystem temperature

6.2.17 Orientation3D

Namespace: UMAA::Common::Measurement::Orientation3D

Description: Orientation3D specifies the orientation of the platform in the order yaw, pitch, roll. The angles are given in a locally level, North-East-Down coordinate system centered on the platform.

Table 207: Orientation3D Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchY	Pitch_HalfAngle	pitchY specifies the platform's rotation about the lateral axis (e.g. the axis parallel to the wings) in a locally level, North-East-Down coordinate system centered on the platform. Pitch is zero when the platform is "nose to tail" level in the North-East plane. The measurement is stated in radians between -0.5 pi and 0.5 pi.
rollX	Roll_Angle	rollX specifies the platform's rotation about the longitudinal axis (e.g. the axis through the body of an aircraft from tail to nose) in a locally level, North-East-Down coordinate system centered on the platform. Roll is zero when the platform is "wing-tip to wing-tip" level in the North-East plane. The measurement is stated in radians between -pi and pi.
yawZ	Yaw_PosAngle	yawZ specifies the platform's rotation about the vertical axis (e.g. the axis from top to bottom through an aircraft) in a locally level, North-East-Down coordinate system centered on the platform. By this definition, yaw is zero when the platform is oriented toward true North and is equivalent to true North referenced heading. The measurement is stated in radians between -pi and pi.

6.2.18 Position2D

Namespace: UMAA::Common::Measurement::Position2D

Description: Position2D specifies a location on the surface of the Earth.

Table 208: Position2D Structure Definition

Attribute Name	Attribute Type	Attribute Description
geodeticLatitude	GeodeticLatitude	geodeticLatitude specifies the north-south coordinate of the position.
geodeticLongitude	GeodeticLongitude	geodeticLongitude specifies the east-west coordinate of the position.

6.2.19 Position3D_PlatformXYZ

Namespace: UMAA::Common::Measurement::Position3D_PlatformXYZ

Description: Position3D_PlatformXYZ specifies a location on a Cartesian coordinate system relative to the origin of the platform.

Table 209: Position3D_PlatformXYZ Structure Definition

Attribute Name	Attribute Type	Attribute Description
xA	Forward	xA specifies the X-axis position which is in the forward (toward the nose) direction.
yA	Right	yA specifies the Y-axis position which is in the right (starboard) direction.
zA	Down	zA specifies the Z-axis position which is in the down (toward the center of the Earth) direction.

6.2.20 Quaternion

Namespace: BasicTypes::Quaternion

Description: Defines a four-element vector that can be used to encode any rotation in a 3D coordinate system.

Table 210: Quaternion Structure Definition

Attribute Name	Attribute Type	Attribute Description
a	double	Real number a.
b	double	Real number b.
c	double	Real number c.
d	double	Real number d.

6.2.21 RetrotraversePolicyType

Namespace: UMAA::MM::CommsLostPolicy::RetrotraversePolicyType

Description: This structure is used to report the travel speed policy in case of lost communication of the unmanned platform.

Table 211: RetrotraversePolicyType Structure Definition

Attribute Name	Attribute Type	Attribute Description
retrotraversePolicy	RetrotraverseType	Retrotraverse action policy while trying to regain comms
waypointPolicy†	WaypointType	An optional travel to a waypoint while trying to regain comms

6.2.22 RetrotraverseType

Namespace: UMAA::MM::Retrotraverse::RetrotraverseType

Description: This structure is used to report the retrotraverse action of an unmanned platform.

Table 212: RetrotraverseType Structure Definition

Attribute Name	Attribute Type	Attribute Description
distance	Distance	Distance along the path to retrotraverse. In the case of retrotraverse to a point, this distance represents the maximum allowed overall travel distance; after this distance, the vehicle should stop even if the target point has not been achieved.
maxSpeed	Speed	The maximum traversal speed. A value of zero means the unmanned platform should not exceed the speed of original travel.
mode	VehicleSpeedModeEnumType	A speed mode.
speedReference	VehicleSpeedReferenceEnumType	A speed reference.
standoffDistance	Distance	Exclusion radius around the destination point. The unmanned platform may not approach closer than this distance.
travelDirection	BooleanEnumType	This field defines the direction of motion used in retrotraverse. If the field is set, the unmanned platform will retrotraverse with positive velocity (conventional direction of travel). If the field is not set, the vehicle travel with negative velocity.
travelMethod	BooleanEnumType	When this field is set, the unmanned platform will travel along a straight line to a point on the originally traveled path that is X distance away from the current position, where X is specified in travelDirection. Otherwise, the unmanned platform will travel a path i.

6.2.23 WaypointType

Namespace: UMAA::MM::BaseType::WaypointType

Description: This structure is used to define attributes of a waypoint including position, depth, and speed.

Table 213: WaypointType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude†	Altitude_MSL	Specifies the distance along the vehicle path to the waypoint
altitudeAGL†	Altitude_AGL	Specifies the distance above ground level
altitudeASF†	Distance_ASF	Specifies the distance above sea level
attitude†	Orientation3D	includes yaw, pitch, roll that vehicle should assume at the given waypoint
depth†	Distance_BSL	Specifies the distance of the waypoint below sea level
endTime†	DateTime	Specifies the end of the valid time period
globalPosition†	Position2D	Specifies the global location of the waypoint (in latitude/longitude)
localPosition†	Position3D_PlatformXYZ	Specifies the local position of the waypoint (in x, y, z)
mode†	HeightModeEnumType	describe whether the waypoint is an altitude or depth mode
pathTolerance†	Distance	the current tolerance of the path measured by distance
speed†	Speed_LocalWaterMass	Specifies the speed to be maintained traveling to the waypoint
startTime†	DateTime	Specifies the beginning of the valid time period
waypointTolerance†	Distance	the current tolerance of the waypoint measured by distance.
waypointType	WaypointKindEnumType	A type of waypoint
waypointID*	NumericGUID	An unique identification of the waypoint

6.3 Enumerations

Enumerations are used extensively throughout UMAA. This section lists the values associated with each enumeration defined in UCS-UMAA.

6.3.1 AnalogSensorErrorCodeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::AnalogSensorErrorCodeEnumType

Description: A mutually exclusive set of values that defines the types of error reporting from the camera.

Table 214: AnalogSensorErrorCodeEnumType Enumeration

Enumeration Value	Description
INVALID_FORMAT	Invalid format
UNKNOWN_ERROR_FAULT	Unknown error fault
UNKNOWN_SENSOR_ID	Unknown sensor ID

6.3.2 AudioEncodingQualityEnumType

Namespace: UMAA::Common::MaritimeEnumeration::AudioEncodingQualityEnumType

Description: A mutually exclusive set of values that defines the types of audio encoding quality.

Table 215: AudioEncodingQualityEnumType Enumeration

Enumeration Value	Description
AVERAGE	Average
BEST	Best
BETTER	Better
GOOD	Good
LESS	Less
POOR	Poor
WORST	Worst

6.3.3 AutomationEnumType

Namespace: UMAA::Common::Enumeration::AutomationEnumType

Description: AutomationEnumTypeLDM is a Realization of AutomationEnumType which is a mutually exclusive set of values that defines automation modes that can be applied to devices and subsystems.

Table 216: AutomationEnumType Enumeration

Enumeration Value	Description
AUTOMATIC	Control over the corresponding element is completely automatic.
MANUAL	Control over the corresponding element is completely manual.
SEMI_AUTOMATIC	Control over the corresponding element is semi-automatic.

6.3.4 BilgeControlEnumType

Namespace: UMAA::Common::MaritimeEnumeration::BilgeControlEnumType

Description: A mutually exclusive set of values that defines the controlling mode of operation of each bilge pump on the unmanned platform.

Table 217: BilgeControlEnumType Enumeration

Enumeration Value	Description
AUTO	Bilge will automatically be turned on by the service when flood is detected in its responsible area
OFF	Off
ON	On

6.3.5 BilgeStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::BilgeStateEnumType

Description: A mutually exclusive set of values that defines the states of each bilge pump on the unmanned platform.

Table 218: BilgeStateEnumType Enumeration

Enumeration Value	Description
FAULT	Fault
OFF	Off
ON	On

6.3.6 BitDepthEnumType

Namespace: UMAA::Common::MaritimeEnumeration::BitDepthEnumType

Description: A mutually exclusive set of values that defines the types of bit depth.

Table 219: BitDepthEnumType Enumeration

Enumeration Value	Description
TEN_BITS	10 bit
TWELVE_BITS	12 bit
FOURTEEN_BITS	14 bit
SIXTEEN_BITS	16 bit
TWENTY_BITS	20 bit
TWENTY_FOUR_BITS	24 bit
THIRTY_TWO_BITS	32 bit
FOURTY_BITS	40 bit
FOURTY_EIGHT_BITS	48 bit

Enumeration Value	Description
FIFTY_SIX_BITS	56 bit
SIXTY_FOUR_BITS	64 bit
EIGHT_BITS	8 bit

6.3.7 BuiltInTestStatusEnumType

Namespace: UMAA::Common::Enumeration::BuiltInTestStatusEnumType

Description: BuiltInTestStatusEnumTypeLDM is a Realization of BuiltInTestStatusEnumType which is a mutually exclusive set of values that defines the state of a Built-In Test.

Table 220: BuiltInTestStatusEnumType Enumeration

Enumeration Value	Description
BIT_SUSPENDED	The built-in test (BIT) has been suspended.
BIT_FAILED	The built-in test (BIT) has failed.
BIT_PASSED	The built-in test (BIT) has passed.
RUNNING_BIT	The built-in test (BIT) is current executing.
OFF_ABORT	The built-in test (BIT) is off or has been aborted.

6.3.8 CollisionAvoidStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CollisionAvoidStateEnumType

Description: A mutually exclusive set of values that defines the self-collision avoidance states supported by the unmanned platform.

Table 221: CollisionAvoidStateEnumType Enumeration

Enumeration Value	Description
DEVIATE_FROM_PATH	Deviate from path to avoid collisions
DO_NOTHING	Do nothing
STOP_ON_PATH	Stop on path

6.3.9 CollisionAvoidStatusEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CollisionAvoidStatusEnumType

Description: A mutually exclusive set of values that defines the self-collision avoidance current status supported by the unmanned platform.

Table 222: CollisionAvoidStatusEnumType Enumeration

Enumeration Value	Description
ACTIVE_AVOID_OBSTACLE	Active, avoid obstacle
ACTIVE_STOP_ON_OBSTACLE	Active, stop on obstacle
DISABLED	Disabled
ENABLED	Enabled

6.3.10 CommandStatusReasonEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CommandStatusReasonEnumType

Description: Defines a mutually exclusive set of reasons why a command status state transition has occurred.

Table 223: CommandStatusReasonEnumType Enumeration

Enumeration Value	Description
CANCELED	Indicates a transition to the CANCELED state when the command is canceled successfully.
VALIDATION_FAILED	Indicates a transition to the FAILED state when the command contains missing, out-of-bounds, or otherwise invalid parameters.
OBJECTIVE_FAILED	Indicates a transition to the FAILED state when the commanded resource is unable to achieve the command's objective due to external factors.
SERVICE_FAILED	Indicates a transition to the FAILED state when the commanded resource is unable to achieve the command's objective due to processing failure.
RESOURCE_FAILED	Indicates a transition to the FAILED state when the commanded resource is unable to achieve the command's objective due to resource or platform failure.
RESOURCE_REJECTED	Indicates a transition to the FAILED state when the commanded resource rejects the command for some reason.
INTERRUPTED	Indicates a transition to the FAILED state when the command has been interrupted by a higher priority process.
TIMEOUT	Indicates a transition to the FAILED state when the command is not acknowledged within some defined time bound.
SUCCEEDED	Indicates the conditions to proceed to this state have been met and a normal state transition has occurred.

6.3.11 CoreStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CoreStateEnumType

Description: A mutually exclusive set of values that defines the states of the systems, subsystems, or components.

Table 224: CoreStateEnumType Enumeration

Enumeration Value	Description
EMERGENCY	Emergency
FAILURE	Failure

Enumeration Value	Description
INITIAL	Initial
READY	Ready
RESET	Reset
RESUME	Resume
SHUTDOWN	Shutdown
STANDBY	Standby

6.3.12 DigitalAudioFormatEnumType

Namespace: UMAA::Common::MaritimeEnumeration::DigitalAudioFormatEnumType

Description: A mutually exclusive set of values that defines the types of digital audio format.

Table 225: DigitalAudioFormatEnumType Enumeration

Enumeration Value	Description
AAC_MPEG2	AAC_MPEG2
AAC_MPEG4	AAC_MPEG4
AIFF	AIFF
ALAC	ALAC
DOLBY_DIGITAL	DolbyDigital
DTS	DTS
FLAC	FLAC
LPCM_PCM	LPCM_PCM
MP2	MP2
MP3	MP3
REAL_AUDIO	RealAudio
SPEEX	Speex
TRUE_AUDIO	TrueAudio
ULAW	uLaw (from DigitalAudioAnnunciator service)
VORBIS	VORBIS
WAV	WAV
WMA	WMA
WMA9_LOSS_LESS	WMA9_LOSSLESS

6.3.13 DigitalSensorErrorCodeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::DigitalSensorErrorCodeEnumType

Description: A mutually exclusive set of values that defines the types of error reporting from the camera.

Table 226: DigitalSensorErrorCodeEnumType Enumeration

Enumeration Value	**Description**
INVALID_FORMAT	Invalid format
INVALID_FRAME_SIZE	Invalid frame size
INVALID_MAX_BIT_RATE	Invalid maximum bit rate
INVALID_MAX_FRAME_RATE	Invalid maximum frame rate
INVALID_MIN_BIT_RATE	Invalid minimum bit rate
INVALID_MIN_FRAME_RATE	Invalid minimum frame rate
MULTIPLE_INVALID_PARAMETERS	Multiple invalid parameters
UNKNOWN_ERROR_FAULT	Unknown error fault
UNKNOWN_SENSOR_ID	Unknown sensor ID

6.3.14 DomainEnumType

Namespace: UMAA::Common::MaritimeEnumeration::DomainEnumType

Description: A mutually exclusive set of values that defines the area or region in which an unmanned platform operates in.

Table 227: DomainEnumType Enumeration

Enumeration Value	**Description**
AIR	Air
GROUND	Surface, ground
SURFACE	Surface, water
UNDERSEA	Undersea

6.3.15 ExposureModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ExposureModeEnumType

Description: A mutually exclusive set of values that defines the exposure mode settings on the camera.

Table 228: ExposureModeEnumType Enumeration

Enumeration Value	**Description**
AUTO_DEFAULT	Auto default
MANUAL	Manual
APETURE_PRIORITY	Manual aperture, automatic shutter speed
SHUTTER_PRIORITY	Manual shutter speed, automatic aperture

6.3.16 FrameSizeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::FrameSizeEnumType

Description: A mutually exclusive set of values that defines the resolution settings for the digital video frame that can be streamed down from the camera.

Table 229: FrameSizeEnumType Enumeration

Enumeration Value	Description
CGA_320x200	CGA_320x200
CIF_1408x1152	CIF_1408x1152
CIF_352x288	CIF_352x288
CIF_704x576	CIF_704x576
EGA_640x350	EGA_640x350
HD1080_1920x1080	HD1080_1920x1080
HD480_852x480	HD480_852x480
HD720_1280x720	HD720_1280x720
HSXGA_5120x4096	HSXGA_5120x4096
QCIF_176x144	QCIF_176x144
QQVGA_160x120	QQVGA_160x120
QSXGA_2560x2048	QSXGA_2560x2048
QVGA_320x240	QVGA_320x240
QXGA_2048x1536	QXGA_2048x1536
SQCIF_128x96	SQCIF_128x96
SVGA_800x600	SVGA_800x600
SXGA_1280x1024	SXGA_1280x1024
UXGA_1600x1200	UXGA_1600x1200
VGA_640x480	VGA_640x480
WHSXGA_6400x4096	WHSXGA_6400x4096
WHUXGA_7680x4800	WHUXGA_7680x4800
WOXGA_2560x1600	WOXGA_2560x1600
WQSXGA_3200x2048	WQSXGA_3200x2048
WQUXGA_3840x2400	WQUXGA_3840x2400
WSXGA_1600x1024	WSXGA_1600x1024
WUXGA_1920x1200	WUXGA_1920x1200
WVGA_852x480	WVGA_852x480
WXGA_1366x768	WXGA_1366x768
XGA_1024x768	XGA_1024x768

6.3.17 GuardedTeleoperationStatusEnumType

Namespace: UMAA::Common::MaritimeEnumeration::GuardedTeleoperationStatusEnumType

Description: A mutually exclusive set of values that defines the status of the guarded teleoperation manager.

Table 230: GuardedTeleoperationStatusEnumType Enumeration

Enumeration Value	Description
DISABLED	Disabled
ENABLED	Enabled but not active
ACTIVE_AVOID_OBSTACLE	Vehicle has deviated from path to avoid obstacle
PITCHOVER_LIMIT	Vehicle has stopped because pitchover limit was reached
ROLLOVER_LIMIT	Vehicle has stopped because rollover limit was reached
ACTIVE_STOP_ON_OBSTACLE	Vehicle has stopped on path due to an obstacle

6.3.18 H264EncodingEnumType

Namespace: UMAA::Common::MaritimeEnumeration::H264EncodingEnumType

Description: A mutually exclusive set of values that defines different type of H264.

Table 231: H264EncodingEnumType Enumeration

Enumeration Value	Description
BASELINE	Baseline
CAVLC444	CAVLC 444
CONSTRAINED_BASELINE	Constrained baseline
CONSTRAINED_HIGH	Constrained high
EXTENDED	Extended
HIGH	High
HIGH10	High 10
HIGH10_INTRA	High 10 Intra
HIGH422	High 422
HIGH422_INTRA	High 422 Intra
HIGH444_INTRA	High 444 Intra
HIGH444_PREDICTIVE	High 444 predictive
MAIN	Main
MULTIVIEW_HIGH	Multiview high
PROGRESSIVE_HIGH	Progressive high
SCALABLE_BASELINE	Scalable baseline
SCALABLE_CONSTRAINED_BASELINE	Scalable constrained baseline
SCALABLE_CONSTRAINED_HIGH	Scalable constrained high
SCALABLE_HIGH	Scalable high
SCALABLE_HIGH_INTRA	Scalable high intra
STEREO_HIGH	Stereo high

6.3.19 H264PresetEnumType

Namespace: UMAA::Common::MaritimeEnumeration::H264PresetEnumType

Description: A mutually exclusive set of values that defines the preset type of H264.

Table 232: H264PresetEnumType Enumeration

Enumeration Value	Description
BEST_QUALITY	Best quality
DRIVE_VISION	Drive vision
LOW_LATENCY	Low latency
MANIPULATION	Manipulation
PERSISTENT_STARE	Persistent stare
PROGRAM_SPECIFIC_1	Program Specific 1
PROGRAM_SPECIFIC_2	Program Specific 2
PROGRAM_SPECIFIC_3	Program Specific 3
PROGRAM_SPECIFIC_4	Program Specific 4
SLOW_COMMS	Slow comms

6.3.20 HandoverResultEnumType

Namespace: UMAA::Common::MaritimeEnumeration::HandoverResultEnumType

Description: A mutually exclusive set of values that defines the status of the mode transition on a vehicle platform.

Table 233: HandoverResultEnumType Enumeration

Enumeration Value	Description
DEFERRED	Control handover deferred temporarily by current controller
DENIED	Control handover denied by current controller
GRANTED	Control handover granted by current controller
TIMEOUT	Control handover timed out because current controller did not respond in the allotted time
INSUFFICIENT_AUTHORITY	Control transfer requestor had insufficient authority to take control from current controller
NOT_AVAILABLE	Vehicle is unavailable for control handover

6.3.21 HeightModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::HeightModeEnumType

Description: An enumeration that is used to set which height mode the unmanned vehicle should be operated in under the specified route.

Table 234: HeightModeEnumType Enumeration

Enumeration Value	Description
ALTITUDE	Height value is distance above sea-floor.
DEPTH	Height value is distance below sea surface.

6.3.22 ImagingModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ImagingModeEnumType

Description: A mutually exclusive set of values that defines intensity of image settings on the camera.

Table 235: ImagingModeEnumType Enumeration

Enumeration Value	Description
COLOR	Default color mode
GREYSCALE	Greyscale
INFRARED	Infrared
LOWLIGHT	Lowlight

6.3.23 IRPolarityEnumType

Namespace: UMAA::Common::Enumeration::IRPolarityEnumType

Description: IRPolarityEnumTypeLDM is a Realization of IRPolarityEnumType which is a mutually exclusive set of values that defines the image polarity of an infrared sensor.

Table 236: IRPolarityEnumType Enumeration

Enumeration Value	Description
BLACK_HOT	In the infrared (IR) image, the black regions indicate hot and white regions indicate cool.
WHITE_HOT	In the infrared (IR) image, the white regions indicate hot and black regions indicate cool.

6.3.24 LightSensitivityEnumType

Namespace: UMAA::Common::MaritimeEnumeration::LightSensitivityEnumType

Description: A mutually exclusive set of values that defines the level of sensitivity of a camera to available light (ISO level).

Table 237: LightSensitivityEnumType Enumeration

Enumeration Value	Description
AUTO_DEFAULT	Auto default

Enumeration Value	Description
ISO_100	ISO 100
ISO_1600	ISO 1600
ISO_200	ISO 200
ISO_3200	ISO 3200
ISO_400	ISO 400
ISO_800	ISO 800

6.3.25 LostCommsStatusEnumType

Namespace: UMAA::Common::MaritimeEnumeration::LostCommsStatusEnumType

Description: A mutually exclusive set of values that defines the lost communications policy status of the unmanned platform.

Table 238: LostCommsStatusEnumType Enumeration

Enumeration Value	Description
ACTIVE	Active
DISABLED	Disabled
ENABLED	Enabled

6.3.26 CommandStatusEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CommandStatusEnumType

Description: Defines a mutually exclusive set of values that defines the states of a command as it progresses towards completion.

Table 239: CommandStatusEnumType Enumeration

Enumeration Value	Description
FAILED	The command has been attempted, but was not successful.
COMPLETED	The command has been completed successfully.
ISSUED	The command has been issued to the resource (typically a sensor or streaming device), but processing has not yet commenced.
COMMANDED	The command has been placed in the resource's command queue but has not yet been accepted.
EXECUTING	The command is being performed by the resource and has not yet been completed.
CANCELED	The command was canceled by the requestor before the command completed successfully.

6.3.27 DataEncodingEnumType

Namespace: UMAA::Common::MaritimeEnumeration::DataEncodingEnumType

Description: A mutually exclusive set of values that defines the types of specialized format used for encoding an elementary data stream within a transport stream.

Table 240: DataEncodingEnumType Enumeration

Enumeration Value	Description
AVI	AVI
H_261	H.261
H_262	H.262
H_263	H.263
H_263PLUS	H.263 Plus
H_264	H.264
MJPEG	MJPEG
MPEG_1	MPEG-1
MPEG_2	MPEG-2
MPEG_4	MPEG-4
NONE	None

6.3.28 TransportEncodingEnumType

Namespace: UMAA::Common::MaritimeEnumeration::TransportEncodingEnumType

Description: A mutually exclusive set of values that defines the encoding methods used for video transport.

Table 241: TransportEncodingEnumType Enumeration

Enumeration Value	Description
MPEG_2	MPEG-2
MP4	MPEG-4
OTHER	Unspecified or unknown format

6.3.29 WaypointKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::WaypointKindEnumType

Description: A mutually exclusive set of values that defines the types of waypoints in the route.

Table 242: WaypointKindEnumType Enumeration

Enumeration Value	Description
APPROACH_FINAL_POINT	Approach final point
APPROACH_INITIAL_POINT	Approach initial point
LAUNCH	Launch
LOITER	Loiter

Enumeration Value	Description
NAV_ONLY	Navigation only
NAV_TARGET	Navigation target
OTHER	Other
RECOVERY	Recovery
RENDEZVOUS	Rendezvous

6.3.30 MeteringModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::MeteringModeEnumType

Description: A mutually exclusive set of values that defines the metering mode settings on the camera.

Table 243: MeteringModeEnumType Enumeration

Enumeration Value	Description
AUTO_DEFAULT	Auto default
CENTER_WEIGHTED	Center weighted
SPOT	Spot

6.3.31 NetworkProtocolEnumType

Namespace: UMAA::Common::Enumeration::NetworkProtocolEnumType

Description: NetworkProtocolEnumTypeLDM is a Realization of NetworkProtocolEnumType which is a mutually exclusive set of values that defines network protocols.

Table 244: NetworkProtocolEnumType Enumeration

Enumeration Value	Description
TCP_IPV4	The network protocol is transmission control protocol (TCP) internet protocol (IP) version 4. This is a reliable, connection based protocol that uses four octets to specify network addresses.
TCP_IPV6	The network protocol is transmission control protocol (TCP) internet protocol (IP) version 6. This is a reliable, connection based protocol that uses 128-bits to specify network addresses.
UDP_IPV4	The network protocol is user datagram protocol (UDP) internet protocol (IP) version 4. This is an unreliable, connectionless protocol that uses four octets to specify network addresses.
UDP_IPV6	The network protocol is user datagram protocol (UDP) internet protocol (IP) version 4. This is an unreliable, connectionless protocol that uses four octets to specify network addresses.

6.3.32 ObstacleAvoidanceEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ObstacleAvoidanceEnumType

Description: A mutually exclusive set of values that defines the obstacle avoidance policy of a given vehicle.

Table 245: ObstacleAvoidanceEnumType Enumeration

Enumeration Value	Description
DEVIATE_FROM_PATH	Deviate from path to avoid obstacle
DO_NOTHING	Do not try to avoid obstacle
STOP_ON_PATH	Stop on path before hitting obstacle

6.3.33 PlatformModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PlatformModeEnumType

Description: A mutually exclusive set of values that defines the mode of a vehicle platform.

Table 246: PlatformModeEnumType Enumeration

Enumeration Value	Description
MAINTENANCE	Maintenance
STANDARD_OPERATING	Standard_Operating
TRAINING	Training

6.3.34 PlatformModeTransitionEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PlatformModeTransitionEnumType

Description: A mutually exclusive set of values that defines the status of the mode transition on a vehicle platform.

Table 247: PlatformModeTransitionEnumType Enumeration

Enumeration Value	Description
ACTIVE	Active
EXITING	Exiting
INITIALIZING	Initializing

6.3.35 PowerStatusEnumType

Namespace: UMAA::Common::Enumeration::PowerStatusEnumType

Description: PowerStatusEnumTypeLDM is a Realization of PowerStatusEnumType which is a mutually exclusive set of values that defines the power state or status of device or subsystem.

Table 248: PowerStatusEnumType Enumeration

Enumeration Value	Description
POWER_OFF	Power for the device is requested/reported to be off.
POWER_ON	Power for the device is requested/reported to be on.
EMERGENCY_POWER	Power for the device is requested/reported to use emergency power.
POWER_STANDBY	Power for the device is requested/reported to use standby power.

6.3.36 RenderUselessStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::RenderUselessStateEnumType

Description: A mutually exclusive set of values that defines the render states of the target system or subsystem.

Table 249: RenderUselessStateEnumType Enumeration

Enumeration Value	Description
NORMAL	Normal
RENDERED_USELESS	Rendered Useless
RENDERED_USELESS_FAILED	Render Useless Failed

6.3.37 ResourceAllocationStatusEnumType

Namespace: UMAA::Common::Enumeration::ResourceAllocationStatusEnumType

Description: ResourceAllocationStatusEnumTypeLDM is a Realization of ResourceAllocationStatusEnumType which is a mutually exclusive set of values that defines allocation status for domain resources.

Table 250: ResourceAllocationStatusEnumType Enumeration

Enumeration Value	Description
FORCED_ALLOCATION	The resource allocation is forced.
FORCED_ALLOCATION_W_LAUNCH_RECOVERY	The resource allocation with launch and recovery is forced.
ALLOCATED	The resource is allocated.
ALLOCATED_W_LAUNCH_RECOVERY	The resource is allocated with launch and recovery.
AVAILABLE	The resource is available.
RELEASED	The resource is released.
TEMPORARILY_UNAVAILABLE	The resource is temporarily unavailable.
UNAVAILABLE	The resource is unavailable.
FAULT	There was a fault in the resource allocation.

6.3.38 SpecificLOIEnumType

Namespace: UMAA::Common::Enumeration::SpecificLOIEnumType

Description: SpecificLOIEnumTypeLDM is a Realization of SpecificLOIEnumType which is a mutually exclusive set of values that defines the Level Of Interoperability (LOI) of a UCS system.

Table 251: SpecificLOIEnumType Enumeration

Enumeration Value	Description
LOI_1	The data link level of interoperability (LOI)-1 is indirect receipt of UAV related data.
LOI_2	The data link level of interoperability (LOI)-2 is direct receipt of ISR or other data where "direct" covers reception of UAV data by the UCS when it has direct communication with the UAV.
LOI_3	The data link level of interoperability (LOI)-3 is control and monitoring of the UAV payload in additional to direct receipt of ISR or other data.
LOI_4	The data link level of interoperability (LOI)-4 is control and monitoring of the UAV, less launch and recovery.
LOI_5	The data link level of interoperability (LOI)-5 is control and monitoring of the UAV (LOI-4) plus launch and recovery functions.

6.3.39 StreamStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::StreamStateEnumType

Description: A mutually exclusive set of values that defines the playback state of the video stream.

Table 252: StreamStateEnumType Enumeration

Enumeration Value	Description
PAUSE	Paused
PLAY	Playing
STOP	Stopped

6.3.40 TamperDetectionStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::TamperDetectionStateEnumType

Description: A mutually exclusive set of values that defines the state of tamper detection.

Table 253: TamperDetectionStateEnumType Enumeration

Enumeration Value	Description
ALWAYS_ENABLED_OR_CLEAR	ALWAYS_ENABLED for reporting data; CLEAR for clearing the previous activities
DISABLED	Disabled
ENABLED	Enabled

6.3.41 VehicleSpeedModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::VehicleSpeedModeEnumType

Description: A mutually exclusive set of values that defines the type of performance speed of the unmanned platform.

Table 254: VehicleSpeedModeEnumType Enumeration

Enumeration Value	Description
LRC	Long-Range Cruise
MEC	Maximum Endurance Cruise
MRC	Maximum Range Cruise
SLOW	Slow speed
VEHICLE_SPECIFIC	Vehicle Specific

6.3.42 VehicleSpeedReferenceEnumType

Namespace: UMAA::Common::MaritimeEnumeration::VehicleSpeedReferenceEnumType

Description: A mutually exclusive set of values that defines the type of referenced speed of the unmanned platform.

Table 255: VehicleSpeedReferenceEnumType Enumeration

Enumeration Value	Description
SPEED_THROUGH_AIR	Air speed
OTHER	Other
RPM	RPM
SPEED_OVER_GROUND	Speed relative to ground
SPEED_THROUGH_WATER	Speed relative to surrounding water

6.3.43 VideoFormatEnumType

Namespace: UMAA::Common::MaritimeEnumeration::VideoFormatEnumType

Description: A mutually exclusive set of values that defines the video formats.

Table 256: VideoFormatEnumType Enumeration

Enumeration Value	Description
NTSCJ	NTSC-J
NTSCM	NTSC-M (default)
PALM	PAL-M
PALN	PAL-N
SECAMBG	SECAM-B/G
SECAML	SECAM-L

6.3.44 WhiteBalanceEnumType

Namespace: UMAA::Common::MaritimeEnumeration::WhiteBalanceEnumType

Description: A mutually exclusive set of values that defines the white balance settings on the camera.

Table 257: WhiteBalanceEnumType Enumeration

Enumeration Value	Description
AUTO_DEFAULT	Auto default
CLOUDY	Cloudy
DAYLIGHT	Daylight
FLASH	Flash
FLUORESCENT	Fluorescent
SHADE	Shade
TUNGSTEN	Tungsten

6.3.45 ZoomModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ZoomModeEnumType

Description: A mutually exclusive set of values that defines the operational mode of camera.

Table 258: ZoomModeEnumType Enumeration

Enumeration Value	Description
ANALOG_ONLY	Analog zoom only
DIGITAL_ONLY	Digital zoom only
OFF	No zoom
MIXED_DEFAULT	System default mixture of analog and digital zoom

6.4 Type Definitions

This section describes the type definitions for UMAA. The table below lists how UMAA defined types are mapped to the DDS primitive types.

Table 259: Type Definitions

Type Name	Primitive Type	Range of Values	Description

Angle	double	fractionDigits=3
		maxInclusive=3.1415926535897932384626433832795
		minInclusive=-3.1415926535897931264626433832795
		units=Radian
		referenceFrame=Counting
			Angle specifies the amount of turning necessary to bring one ray, line or plane into coincidence with or parallel to another. The measurement is stated in radians between -pi and pi.

AnglePosition	double	fractionDigits=3
		maxInclusive=100
		minInclusive=-100
		units=Percent
		referenceFrame=PlatformXYZ
			Represents the position angle measured in percent.

AngleRate	double	units=RadianPerSecond
		minInclusive=-62.831
		maxInclusive=62.831
		fractionDigits=3
			Represents the rate of change of angular displacement measured in radians per second.

BooleanEnumType	boolean	units=N/A
		minInclusive=N/A
		maxInclusive=N/A
		fractionDigits=N/A
		length=N/A
			BooleanEnumTypeLDM is a Realization of BooleanEnumType which is a mutually exclusive set of values that defines the truth values of logical algebra.

CommsRate_BitsPerSecond	long	units=BitPerSecond
		referenceFrame=Counting
			Realizes CommsRateType: an Entity that describes a measure of the data rate over a communications link.

CommsRate_MegabitsPerSecond	double	units=MegabitsPerSecond
		minInclusive=0
		maxInclusive=1125000
		fractionDigits=3
			Represents the number of bits that are conveyed or processed per unit of time measured in megabits per second.

ContinuousSize	double	units=None
		minInclusive=0
		maxInclusive=32767
		fractionDigits=7
			Represents a general size measured to the nearest 0.0000001.

Count	long	units=N/A
		minInclusive=-2147483648
		maxInclusive=2147483647
		fractionDigits=0
			Represents a whole (non-fractional) number that can be positive, negative or zero.

DataTransferRate	double	fractionDigits=6
		maxInclusive=1,125,000,000
		minInclusive=0
		units=KilobitsPerSecond
		referenceFrame=Counting
			Represents the number of bits that are conveyed or processed per unit of time measured in kilobits per second.

DateTimeNanoseconds	long	units=Nanoseconds
		minInclusive=0
		maxInclusive=999999999
		fractionDigits=0
			number of nanoseconds elapsed within the current second.

Type Name	Primitive Type	Range of Values	Description

DateTimeSeconds	longlong	units=Seconds
		minInclusive=0
		maxInclusive=18446744073709500000
		fractionDigits=0
			seconds offset from the standard POSIX (IEEE Std 1003.1-2017) epoch reference point of January 1st, 1970 00:00:00 UTC.

Distance	double	units=Meter
		minInclusive=0
		maxInclusive=401056000
		fractionDigits=3
			This type stores a distance in meters.

Distance__ASF	double	units=Meter
		minInclusive=0
		maxInclusive=401056000
		fractionDigits=3
			The altitude or distance above the sea floor in meters.

Distance__BSL	double	units=Meter
		minInclusive=0
		maxInclusive=10000
		fractionDigits=3
			The distance below sea level in meters.

Distance__Precise	double	units=Meter
		minInclusive=0
		maxInclusive=401056000
		fractionDigits=6
			Represents a distance measured in meters to the nearest 0.000001.

Down	double	axisAbbrev=Z
		axisDirection=down
		axisUnit=Meter
		maximumValue=50000
		minimumValue=-50000
		rangeMeaning=exact
		resolution=0.001
			The Down axis is used for measuring position and increases in magnitude as values extend toward the center of the Earth. Down measurements are expressed in meters.

Duration__Seconds	double	units=Seconds
		minInclusive=0
		maxInclusive=37817280
		fractionDigits=6
			Represents a time duration in seconds.

Effort	double	fractionDigits=3
		maxInclusive=100
		minInclusive=-100
		units=Percent
		referenceFrame=PlatformXYZ
			Represents the level of effort measured in percent.

FieldOfView__LineOfSightFocalPlane	double	fractionDigits=3
		maxInclusive=3.1415926535897932384626433832795
		minInclusive=-3.1415926535897931264626433832795
		units=Radian
		referenceFrame=LineOfSightFocalPlane
			FieldOfView__LineOfSightFocalPlane specifies the solid angle along the Y axis within which a sensor provides detection. The measurement is stated in radians between -pi and pi.

FocusValue	double	maxInclusive=100
		minInclusive=0
		referenceFrame=Counting
			Realizes SizeType: an entity that describes the magnitude or number of a measurable or countable entity.

Type Name	Primitive Type	Range of Values	Description

Forward	double	axisAbbrev=X
		axisDirection=fore
		axisUnit=Meter
		maximumValue=20000000
		minimumValue=-20000000
		rangeMeaning=exact
		resolution=0.001
			The Forward axis is used for measuring position and increases in magnitude as position extends out the "front" of the reference body. Forward measurements are expressed in meters.

FrameRateFPS	double	units=FramesPerSecond
		minInclusive=0
		maxInclusive=1000
		fractionDigits=0
			Represents the number of images transferred or recorded per second.

GeodeticLatitude	double	axisAbbrev=Latitude
		axisDirection=north/south
		axisUnit=Degrees
		maximumValue=90.0
		minimumValue=-90.0
		rangeMeaning=exact
		resolution=0.0000000001
			The Latitude axis is used for measuring position and increases in magnitude as position extends from the south pole to the north pole. Latitude measurements are expressed in degrees.

GeodeticLongitude	double	axisAbbrev=Longitude
		axisDirection=east
		axisUnit=Degrees
		maximumValue=180.0
		minimumValue=-180.0
		rangeMeaning=wraparound
		resolution=0.0000000001
			The Longitude axis is used for measuring position and increases in magnitude as position extends eastward. Longitude measurements are expressed in degrees. Longitude measurements are periodic and whose limits (min and max), while mathematically discontinuous, represent a continuous range.

IPPort_Counting	long	maxInclusive=65535
		minInclusive=0
		units=None
		referenceFrame=Counting
			Realizes AddressType: an Entity that describes a logical location, e.g. an IP address, or port.

MSLHeight	double	axisDirection=up
		axisUnit=Meter
		maximumValue=700000
		minimumValue=-10000
		rangeMeaning=exact
		resolution=0.001
			The MSLHeight axis is used for measuring position and increases in magnitude as values extend away from the center of the Earth. MSLHeight measurements are expressed in meters.

NumericGUID	octet[16]	units=N/A
		minInclusive=0
		maxInclusive=(2^128)-1
		fractionDigits=0
			Represents a 128-bit number according to RFC 4122 variant 2

OrientationQuaternion	BasicTypes::Quaternion	fractionDigits=6
		maxInclusive=1
		minInclusive=-1
		units=N/A
		referenceFrame=Counting
			Represents a unit quaternion (a, b, c, d) which specifies the axis and angle of rotation.

Type Name	Primitive Type	Range of Values	Description

Pitch_HalfAngle	double	fractionDigits=3
		maxInclusive=1.5707963267948966192313216916398
		minInclusive=-1.5707963267948966192313216916398
		units=Radian
		referenceFrame=PlatformNED
			Pitch_HalfAngle specifies the platform's rotation about the lateral axis (e.g. the axis parallel to the wings) in a locally level, North-East-Down coordinate system centered on the platform. Pitch is zero when the platform is "nose to tail level" in the North-East plane. The measurement is stated in radians between -0.5 pi and 0.5 pi.

PowerBusCurrent	double	units=Ampere
		minInclusive=-100000
		maxInclusive=100000
		fractionDigits=3
			Represents the time rate of flow of electric charge measured in amperes.

PowerBusVoltage	double	units=Volt
		minInclusive=-100000
		maxInclusive=100000
		fractionDigits=3
			Represents the potential difference in charge between two points in an electrical field measured in volts.

RadarHeight	double	axisDirection=up
		axisUnit=Meter
		maximumValue=700000
		minimumValue=-10000
		rangeMeaning=exact
		resolution=0.001
			The RadarHeight axis is used for measuring position and increases in magnitude as values extend away from the center of the Earth. RadarHeight measurements are expressed in meters.

Right	double	axisAbbrev=Y
		axisDirection=starboard
		axisUnit=Meter
		maximumValue=20000000
		minimumValue=-20000000
		rangeMeaning=exact
		resolution=0.001
			The Right axis is used for measuring position and increases in magnitude as position extends out the "right" of the reference body. Right measurements are expressed in meters.

Roll_Angle	double	fractionDigits=3
		maxInclusive=3.1415926535897932384626433832795
		minInclusive=-3.1415926535897931264626433832795
		units=Radian
		referenceFrame=PlatformNED
			Roll_Angle specifies a platform's rotation about the longitudinal axis (e.g. the axis through the body of the vehicle from tail to nose) in a locally level, North-East-Down coordinate system centered on the vehicle. Roll is zero when the platform is "wing-tip to wing-tip" level in the North-East plane. The measurement is stated in radians between -pi and pi.

Speed	double	units=MeterPerSecond
		minInclusive=0
		maxInclusive=299792458
		fractionDigits=6
			This type stores speed in meters/s.

Speed_LocalWaterMass	double	units=MeterPerSecond
		minInclusive=0
		maxInclusive=299792458
		fractionDigits=6
			This type stores speed in meters/s.

Type Name	Primitive Type	Range of Values	Description

StringLongDescription	string	fractionDigits=N/A
		length=4095
		maxExclusive=N/A
		maxInclusive=N/A
		minExclusive=N/A
		minInclusive=N/A
		units=N/A
			Represents a long format description.

StringShortDescription	string	fractionDigits=N/A
		length=1023
		maxExclusive=N/A
		maxInclusive=N/A
		minExclusive=N/A
		minInclusive=N/A
		units=N/A
			Represents a short format description.

Temperature	double	units=Celsius
		minInclusive=-273
		maxInclusive=1000
		fractionDigits=3
			Represents the degree or intensity of warmness or coldess presence in a substance. Measured in Celsius.

UniformResourceIdentifier	string	fractionDigits=N/A
		length=2047
		maxExclusive=N/A
		maxInclusive=N/A
		minExclusive=N/A
		minInclusive=N/A
		units=N/A
			Represents a Uniform Resource Identifier (URI).

Yaw_PosAngle	double	fractionDigits=3
		maxInclusive=6.283185307179586364925286766559
		minInclusive=0
		units=Radian
		referenceFrame=PlatformNED
			Yaw_PosAngle specifies the platform's rotation about the Z axis of its body axis system (PlatformXYZ) relative to its velocity vector in the X-Y plane of its body axis system. Yaw is positive in a clockwise direction. The measurement is stated in radians between 0 and 2 pi.

A Appendices

A.1 Acronyms

Note: This acronym list is included in every ICD and covers the complete UMAA specification. Not every acronym appears in every ICD.

ADD	Architecture Design Description
AGL	Above Sea Level
ASF	Above Sea Floor
BSL	Below Sea Level
BWL	Beam at Waterline
C2	Command and Control
CMD	Command
CO	Comms Operations
CPA	Closest Point of Approach
CTD	Conductivity, Temperature and Depth
DDS	Data Distribution Service
EO	Engineering Operations
FB	Feedback
GUID	Globally Unique Identifier
HM&E	Hull, Mechanical, & Electrical
ICD	Interface Control Document
ID	Identifier
IDL	Interface Definition Language Specification
IMO	International Maritime Organization
INU	Inertial Navigation Unit
LDM	Logical Data Model
LOA	Length Over All
LRC	Long Range Cruise
LWL	Length at Waterline
MDE	Maritime Domain Extensions
MEC	Maximum Endurance Cruise
MM	Mission Management
MMSI	Maritime Mobile Service Identity
MO	Maneuver Operations
MRC	Maximum Range Cruise
MSL	Mean Sea Level
OMG	Object Management Group
PIM	Platform Independent Model
PMC	Primary Mission Control
PNT	Precision Navigation and Timing
PO	Processing Operations
PSM	Platform Specific Model
RMS	Root-Mean-Square
RPM	Revolutions per minute
RTPS	Real Time Publish Subscribe
RTSP	Real Time Streaming Protocol

SA	Situational Awareness
SEM	Sensor and Effector Management
SO	Support Operations
SoaML	Service-oriented architecture Modeling Language
STP	Standard Temperature and Pressure
UCS	Unmanned Systems Control Segment
UMAA	Unmanned Maritime Autonomy Architecture
UML	Unified Modeling Language
UMS	Unmanned Maritime System
UMV	Unmanned Maritime Vehicle
UxS	Unmanned System
WGS84	Global Coordinate System
WMO	World Meteorological Organization