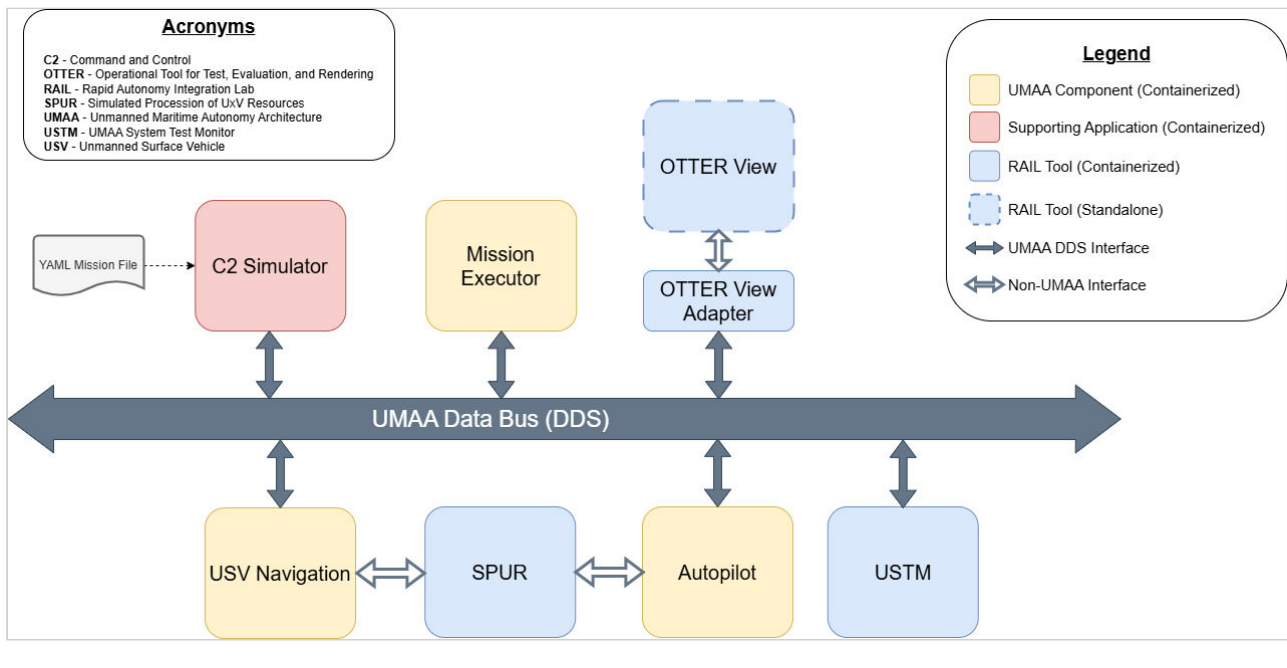


UMAA SDK Distro A Homepage

- [Overview](#)
 - [Running the SDK](#)
 - [Quickstart](#)
 - [Source Code](#)
 - [UMAA Components](#)
 - [Autopilot](#)
 - [USV Navigation](#)
 - [Mission Executor](#)
 - [Supporting Applications](#)
 - [C2 Simulator](#)
 - [RAIL Tools](#)
 - [OTTER View](#)
 - [USTM](#)
 - [SPUR](#)
-

Overview

The Unmanned Maritime Autonomy Architecture (UMAA) Software Development Kit (SDK) provides resources for developers creating their own UMAA components. It includes software components built using UMAA v6.0, example Component Specification documents and support documentation, and RAIL tools showcasing an example UMAA Mission Management simulation. The SDK offers C++ source code examples for components developed by Penn State University Applied Research Laboratory (PSU/ARL) and Johns Hopkins University Applied Physics Laboratory (JHU/APL), leveraging the Eclipse Cyclone DDS middleware, and distributed as Docker images. The SDK is composed of both unmanned underwater vehicle (UUV) and unmanned surface vehicle (USV) software components. Due to the navigation and autopilot components being geared to a USV, the simulated vehicle is only designed to follow desired speed and heading (GlobalVectorControl) commands on the water's surface, and cannot be commanded to dive. The SDK Component Specification documents detail the project-specific requirements implemented by these components. They also serve as templates for creating similar documentation for other components. Note that these requirements are fictional and not based on any real project or platform.



Running the SDK

Quickstart

The UMA SDK consists of several Docker containers that are managed using Docker Compose. The Docker Compose file as well as the dependencies list, setup and running instructions are available in the Quickstart Guide. Before running the Quickstart guide, download the sources listed below.

Download and extract the source for the UMA SDK Quickstart at **umaa-sdk-quickstart-v1.0.0-DistroA.tar.gz**

Download and extract OTTER View at **OTTER_View-2.2.0-DELTA.zip**

Download and load the SDK Docker container images at **UMAA-SDK-images-v1.0.0-DistroA.tar.gz**

```
# Load container images (Requires Docker to be installed)
docker load -i UMAA-SDK-images-v1.0.0-DistroA.tar.gz
```

Source Code

SDK source code can be downloaded through the attachment **UMAA-SDK-sources-v1.0.0-DistroA.tar.gz**

This bundle includes the projects for **Autopilot**, **C2 Sim**, **Mission Execution Manager**, **Route Objective Executor**, and **USV Navigation**.

- Note: The contents of the common library (umaa-sdk-common) can be copied into the umaa-sdk-common folder in the C2 Sim, Mission Execution Manager, and Route Objective Executor projects.



* In UMAA v6.0, the Mission Executor component was officially released, and definitions for Mission Execution Manager (MEM) and Route Objective Executor (OE) were included as experimental. SDK v1.0.0 has separate source code repositories for MEM and Route OE, but combines these into the aggregate Mission Executor component.

UMAA Components

Autopilot

The Autopilot component provides for vehicle control. The component definition states that Autopilot can be commanded using GlobalVectorControl, GlobalHoverControl, or PrimitiveDriverControl services. For the SDK, Autopilot utilizes the GlobalVectorControl service to receive vector commands from the Mission Executor, and sends vehicle control commands to SPUR to maneuver the simulated vehicle.

USV Navigation

The USV Navigation component receives navigation data of the simulated vehicle from SPUR, and provides the own-ship information, including position and speed, onto the UMAA DDS bus.

Mission Executor

The Mission Executor component is responsible for managing and monitoring the execution of objectives. The SDK's implementation of the Mission Executor executes Route Objectives. It receives navigation data published by USV Navigation and publishes a vector control command to the Autopilot to control the simulated vehicle along the planned route.

Supporting Applications

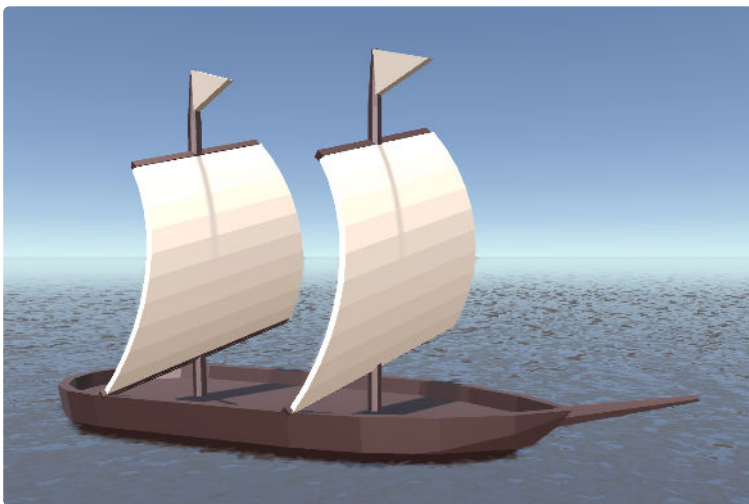
C2 Simulator

The Command and Control (C2) Simulator is used to send mission information to the Mission Executor component given a YAML Mission File. Note: this is not a UMAA-defined component.

RAIL Tools

OTTER View

OTTER (Operational Test Tool for Evaluation and Rendering) View is a 3D visualization tool that allows developers to view results from in-water tests and simulations. SDK missions use the USV vehicle and operate in San Francisco, California.



The image above depicts a ship in OTTER View.

USTM

USTM (UMAA System Test Monitor) is an automated tool which monitors DDS messages to aid in compliance checking to the UMAA standard. The SDK demonstrates USTM compliance feedback via a localhost web GUI, but USTM is also capable of producing J-UNIT XML outputs as part of a CI/CD pipeline.



Passing USTM alone does not indicate UMAA compliance.

For the release of **SDK v1.0.0**, USTM supports the following verification for all UMAA v6.0 services:

- Topic Name validation
- Type Name validation
- Strict Type Matching
- Range-boundary validation for all UMAA types with range-boundaries defined
- Ability to set custom range-boundary values for stricter testing
- UMAA Command State Transition Flow Control
- Support for LargeList and LargeSet Validation
- Component Level grouping / validation of services

SPUR

SPUR (Simulated Procession of UxV Resources) is a multi-UxV simulation tool based on GTRI's SCRIMMAGE Sim and is used to simulate interactions with UMAA-compliant components and vehicle behavior.

Plugins used by the SDK

Plugin Type	Plugin Name	Notes
Autonomy	GenericUdpVectorCommand	
Controller	Iver3Pilot	If used with any autonomy other than RemoteHelm, fin commands must not be allowed. This controller clamps speed to 4 knots.
Motion Model	Iver3	This motion model can only be used with the Iver3Pilot controller.
Sensors	UuvRpmVelocity	

Plugin Type	Plugin Name	Notes
	UuvGps	
	UuvCtd	
	UuvDvl	
	UuvCompass	
	UuvImu	
	UuvIns	
	GenericUdpNav	The UuvCtd, UuvGps, UuvDvl, and UuvCompass plugins should be included on the same entity when using this plugin.