**Contract #: DTFH6116D00035**

**TOPR #: HOIT212116217**

**JPO**

Operational Data Environment

**User Guide**

**Submitted to:**

U.S. Department of Transportation (USDOT)

Federal Highway Administration ITS JPO

**May 4, 2017**

**Prepared by:**

Booz Allen Hamilton  
8283 Greensboro Drive  
McLean, VA 22102

This page intentionally blank

**Booz Allen Hamilton**

8283 Greensboro Drive

McLean, VA 22102-3838

Tel 703-902-5000

Fax 703-902-3333

[www.boozallen.com](http://www.boozallen.com)

Contents

[Version History 6](#_Toc481749633)

[1 Introduction 7](#_Toc481749634)

[2 Project Overview 7](#_Toc481749635)

[3 System Overview 8](#_Toc481749636)

[4 Audience 8](#_Toc481749637)

[5 Glossary 9](#_Toc481749638)

[6 ODE DEVELOPMENT ENVIRONMENT 11](#_Toc481749639)

[6.1 Java Development Tools 11](#_Toc481749640)

[6.2 Java 11](#_Toc481749641)

[6.3 Eclipse IDE 11](#_Toc481749642)

[6.4 Maven 11](#_Toc481749643)

[6.5 Git Version Control 11](#_Toc481749644)

[6.6 Building ODE Software Artifacts 12](#_Toc481749645)

[6.6.1 Open-Source Repository 12](#_Toc481749646)

[6.6.2 Private Repository 12](#_Toc481749647)

**[6.6.3](#_Toc481749648)** [ODE Application Properties 13](#_Toc481749648)

[6.6.4 ODE Logging Properties 17](#_Toc481749649)

[7 ODE Features 18](#_Toc481749650)

[7.1 Managing SNMP Devices 18](#_Toc481749651)

[7.1.1 Query Parameters 18](#_Toc481749652)

[7.1.2 API Details 19](#_Toc481749653)

[7.1.3 Web Based View 19](#_Toc481749654)

[7.1.4 Additional Features/ Discussion Points 19](#_Toc481749655)

[7.2 Logging Events 19](#_Toc481749656)

[7.2.1 Log Levels 19](#_Toc481749657)

[7.2.2 Logging setup 20](#_Toc481749658)

[7.2.3 Steps to turn on/off logging during application runtime. 20](#_Toc481749659)

[7.3 IEEE 1609.2 Compliance 21](#_Toc481749660)

[7.4 SCMS Certificate Management 21](#_Toc481749661)

[7.5 Inbound BSM Distribution 21](#_Toc481749662)

[7.6 Inbound Probe Data Distribution 21](#_Toc481749663)

[7.7 Outbound Probe Data Management 21](#_Toc481749664)

[7.7.1 PDM Broadcast Request Quick Start Guide 21](#_Toc481749665)

[7.8 Outbound TIM Broadcast 22](#_Toc481749666)

[7.8.1 Outbound TIM Setup 22](#_Toc481749667)

[7.8.2 TIM Broadcast Request Quick Start Guide 23](#_Toc481749668)

[7.9 Inbound TIM Distribution 24](#_Toc481749669)

[7.10 Data validation 24](#_Toc481749670)

[7.11 Data Sanitization 24](#_Toc481749671)

[7.12 VSD Deposit to SDC 24](#_Toc481749672)

[8 Appendix A: ODE Interface Specification 24](#_Toc481749673)

[8.1 File Copy Data Deposit 25](#_Toc481749674)

[8.1.1 Messages and Alerts 25](#_Toc481749675)

[8.2 ODE REST API 28](#_Toc481749676)

[8.2.1 Upload BSM File 31](#_Toc481749677)

[8.2.2 MANAGE SNMP API 32](#_Toc481749678)

[8.2.3 Traveler Information Message (TIM) Interface 33](#_Toc481749679)

[8.2.4 Probe Data Management Messages (PDM) Interface 34](#_Toc481749680)

[8.3 ODE Streaming API 35](#_Toc481749681)

[8.3.1 Direct Kafka Interface 35](#_Toc481749682)

[8.3.2 ODE WebSocket Interface 36](#_Toc481749683)

[8.4 ODE Request Schemas 38](#_Toc481749684)

[8.4.1 ODE Request Message Metadata 38](#_Toc481749685)

[8.5 ODE Response Schemas 39](#_Toc481749686)

[8.5.1 ODE Data Message 39](#_Toc481749687)

[8.5.2 ODE Message Metadata 39](#_Toc481749688)

[8.5.3 ODE Payload Violation 41](#_Toc481749689)

[8.5.4 ODE GET TOKEN Response 41](#_Toc481749690)

[8.5.5 ODE Status Message 42](#_Toc481749691)

[8.5.6 ODE Control Message 42](#_Toc481749692)

[8.5.7 ODE Data Message Payload 43](#_Toc481749693)

[8.5.8 ODE Data Message Supporting Data Structures 44](#_Toc481749694)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Implemented By | Revision Date | What Changed? |
| 1.0 | Hamid Musavi |  | Initial draft |
|  | Hamid Musavi | 3/6/2017 | Updated document for [ODE-146](https://usdotjpoode.atlassian.net/browse/ODE-146) |
|  |  |  |  |
|  |  |  |  |

# Introduction

The JPO Operational Data Environment (ODE) product is being developed under Agile Development Methodologies, using an open architecture approach, in an open source environment. This document describes the preliminary architectural design of the JPO ODE and its interfaces with external systems including the TMC applications, field devices and center services.

Note: This is a living document and will be updated throughout the life of the JPO ODE project to reflect the most recent changes in the ODE design and stakeholder feedback. All stakeholders are invited to provide input to this document. Stakeholders may direct all input to the JPO Product Owner at DOT, FHWA, JPO. To provide feedback, we recommend that you create an “[issue](https://github.com/usdot-jpo-ode/jpo-ode/issues)” in the project’s GitHub repository (<https://github.com/usdot-jpo-ode/jpo-ode/issues>). You will need a GitHub account to create an issue. If you don’t have an account, a dialog will be presented to you to create one at no cost.

# Project Overview

An Operational Data Environment is a real-time data acquisition and distribution software system that processes and routes data from Connected-X devices – including connected vehicles (CV), personal mobile devices, infrastructure components, and sensors – to subscribing applications to support the operation, maintenance, and use of the transportation system, as well as related research and development efforts.

The ODE is intended to complement a connected vehicle infrastructure by brokering, processing and routing data from various data sources, including connected vehicles, field devices, Transportation Management Center (TMC) applications and a variety of other data users. Data users include but not limited to transportation software applications, Research Data Exchange (RDE), US DOT Situation Data Warehouse.

As a data provisioning service, the ODE can provision data from disparate data sources to software applications that have placed data subscription requests to the ODE. On the other direction, the ODE can accept data from CV applications and broadcast them to field devices through Road Side Units (RSU) and US DOT Situation Data Warehouse which in turn will transmit the data to Sirius XM satellites for delivery to the connected vehicles in the field.

While provisioning data from data sources to data users, the ODE also will perform necessary security / credential checks and, as needed, data validation and sanitization.

* Data validation is the process of making a judgment about the quality of the data and handling invalid data as prescribed by the system owners.
* Data sanitization is the modification of data as originally received to reduce or eliminate the possibility that the data can be used to compromise the privacy of the individual(s) that might be linked to the data.

# System Overview

JPO ODE is an open-sourced software application that will enable the transfer of data between field devices and backend TMC systems for operational, monitoring, and research purposes. The system will enable applications to submit data through a variety standard interfaces as illustrated in the figure below.

The mechanisms chosen for a specific deployment will depend on the infrastructure, technical resources, and applications available to an ODE environment.

The JPO-ODE will be designed to support the producers and consumers of CV data as illustrated in Figure 1 below. ***The implementation timeline for the identified interfaces will depend on the needs of the JPO ODE customers (Wyoming CV Pilot site, initially) and the priority of these capabilities to the JPO-ODE product owner.***



Figure - ODE System Data Producers and Consumers

# Audience

This document is intended for use by the ODE client applications.

# Glossary

|  |  |
| --- | --- |
| Term | Description |
| API | Application Program Interface |
| ASN.1 | Abstract Syntax Notation One (ASN.1) is a standard and notation that describes rules and structures for representing, encoding, transmitting, and decoding data in telecommunications and computer networking |
| Git | Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. <https://git-scm.com/> |
| JDK | Java Development Kit |
| JPO | Joint Program Office |
| JRE | Java Runtime Environment |
| JVM | Java Virtual Machine |
| Kafka | Apache Kafka is publish-subscribe messaging rethought as a distributed commit log. |
| POJO | Plain Old Java Object |
| SAE | SAE International is a global association of more than 128,000 engineers and related technical experts in the aerospace, automotive and commercial-vehicle industries. |
| J2735 | This SAE Standard specifies a message set, and its data frames and data elements specifically for use by applications intended to utilize the 5.9 GHz Dedicated Short Range Communications for Wireless Access in Vehicular Environments (DSRC/WAVE, referenced in this document simply as “DSRC”), communications systems. |
| SCP | Secure Copy |
| SDW | Situation Data Warehouse |
| TIM | Traveler Information Message |
| US DOT | Unites States Department of Transportation |
| WebSocket | WebSocket is designed to be implemented in web browsers and web servers, but it can be used by any client or server application. The WebSocket Protocol is an independent TCP-based protocol. Its only relationship to HTTP is that its handshake is interpreted by HTTP servers as an Upgrade request. |
| ZooKeeper | Apache ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. |

This page intentionally left blank

# ODE DEVELOPMENT ENVIRONMENT

## Java Development Tools

The ODE team uses Java as the primary programming language.

Tools:

* Java
* Eclipse IDE
* Git
* Maven
* GitHub: <https://github.com/usdot-jpo-ode/jpo-ode>

## Java

Install Java Development Kit (JDK) 1.8

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

## Eclipse IDE

Download and install Eclipse.

<https://eclipse.org>

Configure Eclipse to use Java 1.8 JDK. Local installation of Tomcat can integrate with Eclipse and can help with prototyping or debugging the application.

## Maven

Maven is a build and dependency management tool. It is recommended that a Maven plug-in is installed with your IDE so that your IDE is Maven "aware". Newer versions of eclipse (Luna and later versions) comes pre-installed with a Maven plug-in.

Download and install Maven: <https://maven.apache.org/>

## Git Version Control

The ODE software is maintained and version controlled using GIT version control system.

Recommend clients:

* Tortoise Git
* Source Tree
* GitHub Windows Desktop Application
* Git Extensions

It is recommended that GIT plug-ins are installed with your IDE so that your IDE is Git "aware". Newer versions of eclipse (Luna and later versions) comes pre-installed with a Git plug-in.

## Building ODE Software Artifacts

The ODE source code is maintained in two separate Git repositories.

1. The public Git repository, <https://github.com/usdot-jpo-ode/jpo-ode>, houses the entirety of the JPO ODE open-source software source code.
2. The private Git repository, <https://bitbucket.org/usdot-jpo-ode/jpo-ode-private>, holds the components of the software that may be proprietary and/or hold commercial license and therefore not permitted to be open to public use.

### Open-Source Repository

The ODE deployment artifact consists of one of more jar files that make up the collection of software modules and service components. Initially, there will be only one executable jar file (one micros service) but in the future as the ODE functionality expands it is envisioned that additional services be introduced in separate jar files. Each service component jar file will be a standalone “uber-jar” that contains all necessary dependent jar files. The jar file will be deployable to a physical or virtual server as well as within a Docker container.

The following components make up the JPO ODE software:

* jpo-ode-common: this component contains all the common classes used by other jpo-ode components. *This component is the lowest common denominator and never depends on any other jpo-ode component.*
* jpo-ode-core: this component contains the core functions carried out by the jpo-ode.
* jpo-ode-plugins: this component contains the plug-in modules.
* jpo-ode-svcs: this component and similar future components are the actual service components. This component is always a Spring Framework application and implements a specific service.

### Private Repository

The Private repository is intended to house the 3rd-party non-open source components of the software. Therefore, by nature this repository is quite static and once built and installed on a given build machine, it does not need to be rebuilt as long as no updates have been made to that source code. Currently this repository will contain only the [OSS Nokalva](http://www.oss.com/) ASN.1 compiler and runtime libraries.

#### ASN.1 Java API

The data uploaded or deposited to the ODE from the connected vehicles (CV) and the road-side units (RSU) is encoded in ASN.1 format. In order for the ODE to utilize the data, it must be able to decode the data from ASN.1 format into a more generic format, in this case Plain Old Java Objects (POJOs). We have acquired ASN.1 compiler and run-time libraries from [OSS Nokalva](http://www.oss.com/) to accomplish the decoding/encoding of data from ASN.1 to POJO and vice-versa.

Due to the commercial license associated with the ASN.1 compiler software, the source code generated by the compiler and the runtime libraries required to build and run the software are maintained in our private Git repository.

#### Build and Deploy Procedure

Follow the steps in jpo-ode/README.md Getting Started guide for building and deploying the JPO-ODE services.

### ODE Application Properties

JPO ODE configuration can be modified in a number of ways.

1. You can specify the configuration parameters in a file named *application.properties* located in the same directory from which the application is launched.
2. You may specify properties as command line options in the form of *--ode.propertyName=propertyValue*. For example, add ode.DdsCasUsername=fred.flintstone@stone.age
3. You may specify properties as system environment variables in the form of *ode.DdsCasUsername=fred.flintstone@stone.age.*

Other properties not specific to the ODE can also be defined in a similar way but without the *ode* prefix.

Current ODE properties and their default are defined in OdeProperties class. The property name is the name of the OdeProperties class instance parameter.

The following table describes all the ODE properties currently available.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Default Value | Required | Description |
| ode.kafkaBrokers | $DOCKER\_HOST\_IP:9092 | X | List of kafka brokers and ports |
| ode.uploadLocationRoot | ./uploads |  | Location of the shared directory where ODE monitors for files to ingest. |
| ode.uploadLocationBsm | ./uploads/bsm |  | Specific location for BSM files without message-frame header |
| ode.uploadLocationMessageFrame | ./uploads/messageframe |  | Specific location for BSM files with message-frame header |
| ode. pluginsLocations | ./plugins |  | Location of the jar files for ODE plugins. |
| ode. asn1CoderClassName | us.dot.its.jpo.ode.plugin.j2735.oss.OssAsn1Coder |  | How Kafka producers write to topics: valid values are async and sync for asynchronous writes and synchronous writes, respectively. |
| ode. kafkaProducerType | async |  |  |
| ode. ddsCasUsername |  | X | Username to be used for authentication when interfacing with Situation Data Warehouse |
| ode. ddsCasPassword |  | X | Password to be used for authentication when interfacing with Situation Data Warehouse (SDW) |
| ode. ddsCasUrl | https://cas.connectedvcs.com/accounts/v1/tickets |  | URL of the US DOT security server. |
| ode. ddsWebsocketUrl | wss://webapp2.connectedvcs.com/whtools23/websocket |  | URL of the US DOT SDW WebSockets API |
| ode. kafkaTopicBsmSerializedPOJO | topic.J2735Bsm |  | Name of the serial Kafka topic |
| ode. kafkaTopicBsmJSON | topic.J2735BsmRawJSON |  | Name of the JSON Kafka topic |

### ODE Logging Properties

ODE produces two log files:

1. The application log file: for overall application health monitoring
2. Events log file: for tracking and monitoring major data events such as the flow of data files through the system

The configuration of the loggers is done via *logback.xm*l file. The default logback.xml is located in the *src/main/resources* directory of the source code as well as in the *BOOT-INF\classes\* directory of the executable jar file. To modify the default values, you can modify the source *src/main/resources/logback.xml* file before building the software or place a different *logback.xml* file with the modified values in the working directory of the application.

# ODE Features

JPO ODE provides the following features and functions to TMC applications:

1. Managing SNMP Devices
2. Logging Events
3. IEEE 1609.2 Compliance
4. SCMS Certificate Management
5. Inbound BSM Distribution
6. Inbound Probe Data Distribution
7. Outbound Probe Device Management
8. Outbound TIM Broadcast
9. Inbound TIM Distribution
10. Data Validation
11. Data Sanitization

## Managing SNMP Devices

Over SNMP Protocol, the ODE can ping and assess the health of an existing Road Side Unit to ensure the system is up and running. To trigger a specific heartbeat call, the ODE provides two separate interfaces to deploy a message to an RSU.

### Query Parameters

To make a heartbeat call, a user must provide two pieces of information to identify the device and the information the user is attempting to capture.

**IP Address:** The published ip address of the device.

**SNMP OID Value:** The numeric OID of the desired information.

The OIDs for the RSUs are specified in the DSRC Roadside Unit (RSU) Specifications Document v4.1. The units also respond to ISO standard OIDs, as demonstrated in the screenshot below.

### API Details

To get the results from the SNMP protocol, submit a RESTful GET request to the route listed below.

/rsuHeartbeat?ip=<ip\_address>&oid=<oid\_string>

You should receive a detailed plain text response that looks like the following example. If the device is off, a 4 second timeout will occur and the ODE will indicate this with an "[ERROR] Empty response" message. (This specific OID returns the amount of time since the device was last powered on)

[1.3.6.1.2.1.1.3.0 = 0:05:12.59]

### Web Based View

An additional method way to interact with the heartbeat service is through the existing web interface located at the root of the application. On it, a user will see a section for RSU SNMP Query and may enter in the same IP and OID information as the API Endpoint.

### Additional Features/ Discussion Points

* SNMP v3 discussion needed surrounding v2, v1 support
  + V3 username/password
* Should the responses from the application be in a standard format? (JSON)

## Logging Events

ODE uses Logback logging framework to log application and data events.

### Log Levels

1. ALL - Logger reports to all levels below
2. DEBUG - Logger reports debug information
3. ERROR - Logger reports error events that may still allow the application to continue running
4. FATAL - Logger reports fatal errors that will cause the application to abort
5. INFO - Logger reports informational messages
6. OFF - Turns off the logger
7. TRACE - Logger reports more specific debug information
8. WARN - Logger reports application warnings

### Logging setup

* As it stands, the current logging framework has two separate log files. The first log file is for application output called ode.log. Application debug information and backend service messages are output to this file. The second log file, Events.log contains informational messages pertaining to the services a message goes through inside of the system.
* The current setup of the logging framework is very minimal. It contains four loggers and two appenders for the respective files. The logback framework has the ability to set time based file deletion, and rolling archive file naming. For the full list of features visit this URL: <https://logback.qos.ch/manual/>

### Steps to turn on/off logging during application runtime.

1. Start ode, Kafka, and Zookeeper as normal.
2. In a new terminal window run "jconsole".
3. After the dialog box comes up asking for connection, click on the remote access button at the bottom.
4. Input the ip address you set to be your DOCKER\_HOST\_IP:9090 (ex. 0.0.0.0:9090).
5. Click connect.
6. Select insecure connection.
7. Select the MBeans tab at the top.
8. Expand the folder ch.qos.logback.classic until you get to Attributes and Operations.
9. Open the operations Tab.
10. Select the reloadbyfilename option.
11. In the dialog box input the name of your logging configuration file. (Currently logback.xml)
12. Edit logback.xml inside of the docker container for ode and modifiy the log level for whatever logger you wish to turn off to "OFF".
13. Save the file and go back to the jconsole and click the button reloadbyfilename to submit changes.

## IEEE 1609.2 Compliance

TBD

## SCMS Certificate Management

TBD

## Inbound BSM Distribution

ODE accepts Inbound BSMs via File Copy Data Deposit mechanism as described in section 8.1.

The ODE propagates BSM data to applications via a subscription service provided by Kafka messaging hub. The ODE offers two Kafka BSM subscription formats, JSON and serialized Java objects (also referred to as POJO). ODE uses Kryo serializer for serializing POJOs before publishing. See section 8.3.1 for the topic names to which applications can subscribe.

## Inbound Probe Data Distribution

TBD

## Outbound Probe Data Management

ODE accepts PDM messages and other metadata parameters for broadcasting PDM messages via the REST API interface. The ODE accepts data elements in JSON which are then sent via SNMP to an array of Roadside Units (RSUs) which are also specified in that same JSON string.

### PDM Broadcast Request Quick Start Guide

To run a local test of the PDM message API, please follow these instructions.

1. Start the ODE.
2. Reference the Swagger documentation located in the /docs folder of the repo to view the specifications for the API call. If needed, paste the YAML file into http://editor.swagger.io to see a rendered webpage for the documentation.
3. Use a web based REST tool such as Postman to send the PDM broadcast request to the ODE. Make sure the REST request body contains the “snmp” and “rsus” elements with valid IP addresses of the RSUs that you intend to send the message to.
4. The REST interface will return a response indicating the request was executed successfully: {success: true}. If the request fails, you will receive an error message such as:  
   {  
    "timestamp": 1489415494755,  
    "status": 400,  
    "error": "Bad Request",  
    "exception": "us.dot.its.jpo.ode.traveler.TimMessageException",  
    "message": "us.dot.its.jpo.ode.traveler.TimMessageException: Empty response from RSU 127.0.0.1",  
    "path": "/tim"  
   }

## Outbound TIM Broadcast

ODE accepts TIM messages and other metadata parameters for broadcasting TIM messages via the REST API interface. The ODE accepts data elements in JSON format from which a fully formed ASN.1 compliant J2735 TravelerInformation message will be constructed and sent to an array of RSUs. The RSUs must be specified in the TIM broadcast message received by the ODE. In addition to the RSU devices, the TIM message is also deposited to the US DOT Situation Data Warehouse (SDW) from which the SiriusXM satellites will pull from and broadcast to vehicles that are not within range of RSUs. SDW parameters are also specified in the TIM REST interface. Please refer to the Swagger file documentation for details of a TIM REST interface.

### Outbound TIM Setup

1. ODE **Configuration**: Update the effective application.properties file with username and password for Webapp2/sdw. Substitute your username and password for <SDWUSERNAME> and <SDWPASSWORD>, respectively.

ode.ddsCasUsername=<SDWUSERNAME>

ode.ddsCasPassword=<SDWPASSWORD>

OR defined the following command line arguments while launching jpo-ode-svcs

--ode.ddsCasUsername=<SDWUSERNAME>, \

--ode.ddsCasPassword=<SDWPASSWORD>

Or define the following system properties / environment variables

ode.ddsCasUsername=<SDWUSERNAME>

ode.ddsCasPassword=<SDWPASSWORD>

* **RSU Enablement**: /tim REST service sends the TIM messages to RSUs if both “rsus” and “snmp” elements of the request body are defined and valid. If either “rsus” or “snmp” are missing, the request will not be sent to the RSUs.
* **SDW Enablement**: /tim REST service sends the TIM messages to SDW if the “sdw” element of the request body is defined and valid. If “sdw” element is missing, the request will not be sent to the SDW.

### TIM Broadcast Request Quick Start Guide

To run a local test of the TIM Message API, please follow these instructions:

1. Start the ODE with valid ode.ddsCasUsername and ode.ddsCasPassword in the effective application.properties file.
2. Reference the Swagger documentation located in the /docs folder of the repo to view the specifications for the API call. If needed, paste the YAML file into http://editor.swagger.io to see a rendered webpage for the documentation.
3. Copy the curl command, run the python script, or use a web based REST tool such as Postman to send the TIM broadcast request to the ODE. Make sure the REST request body contains the “snmp” and “rsus” elements with valid IP addresses of the RSUs that you intend to send the message to as well as the required SDW parameters.
4. The REST interface will return a response indicating the request was executed successfully: {success: true}. If the request fails, you will receive an error message such as:  
   {  
    "timestamp": 1489415494755,  
    "status": 400,  
    "error": "Bad Request",  
    "exception": "us.dot.its.jpo.ode.traveler.TimMessageException",  
    "message": "us.dot.its.jpo.ode.traveler.TimMessageException: Empty response from RSU 127.0.0.1",  
    "path": "/tim"  
   }

## Inbound TIM Distribution

TBD

## Data validation

TBD

## Data Sanitization

TBD

## VSD Deposit to SDC

ODE sends VSD message to SDC using UDP protocol rather than Web Socket protocol. UDP protocol supports transfer of high volume data and is more scalable whereas web socket protocol is not intended for transferring high volume data. Unlike TCP, UDP is a best effort delivery service which means that the protocol does not worry about acknowledging whether the receiver actually received the message or not. The VSD depositor is implemented as a module in the ODE app and follows the VSD dialog mentioned below for depositing VSD message to SDC.

**VSD Deposit Dialog**

* 1. ODE ------ServiceRequest-----> SDC // ODE sends service request to SDC
  2. ODE <----ServiceResponse----- SDC // ODE receives service response from SDC
  3. ODE --------VsdMessage-------> SDC // ODE sends the actual VSD message to SDC

**Requirements for send VSD message to SDC over UDP**

1. IP address or Domain Name of the SDC server
2. Port number of the SDC server

# Appendix A: ODE Interface Specification

Field devices and TMC applications interface with the ODE for both sending and receiving data to and from the ODE

Ode provides two methods of accepting data from field devices:

* File copy: described in section 7.1
* RESTful API: upload described in section 7.2

ODE provides several methods for the TMC applications (or any ODE client application) to send and receive data to and from the ODE

* RESTful API: upload described in section 7.2
* Streaming API: described in section 7.3.

All of the above interfaces can be secured using SSL encryption.

## File Copy Data Deposit

The File copy method is achieved by providing a configurable location on a shared file system where field devices will be able to deposit their data files and log files for processing. The upload location is specified by the application properties ode.uploadLocationRoot/ode.uploadLocationBsm and ode.uploadLocationRoot/ode.uploadLocationMessageFrame. If not specified, default locations would be uploads/bsm and uploads/messageframe sub-directories off of the location where ODE is launched. ODE creates the specified directories if they do not exist.

Once the ODE processes the received file, it moves it to the “backup” sub-directory under the respective directory. The backed up file is renamed with a timestamp in milliseconds.

### Messages and Alerts

This interface uses the file system to copy a file from source to destination. As a result, the messages and alerts generated by the copy command are platform dependent. The following table describes a sample set of exit codes returned by scp command but they may differ from the system on which ODE is deployed and running.

Table 1 - SCP Return Codes

|  |  |
| --- | --- |
| 0 | Operation was successful |
| 1 | General error in file copy |
| 2 | Destination is not directory, but it should be |
| 3 | Maximum symlink level exceeded |
| 4 | Connecting to host failed. |
| 5 | Connection broken |
| 6 | File does not exist |
| 7 | No permission to access file. |
| 8 | General error in sftp protocol |
| 9 | File transfer protocol mismatch |
| 10 | No file matches a given criteria |
| 65 | Host not allowed to connect |
| 66 | General error in ssh protocol |
| 67 | Key exchange failed |
| 68 | Reserved |
| 69 | MAC error |
| 70 | Compression error |
| 71 | Service not available |
| 72 | Protocol version not supported |
| 73 | Host key not verifiable |
| 74 | Connection failed |
| 75 | Disconnected by application |
| 76 | Too many connections |
| 77 | Authentication cancelled by user |
| 78 | No more authentication methods available |
| 79 | Invalid user name |

Table - File Copy Data Deposit Messages and Alerts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Message or Alert | Communication Method | Description | Algorithm | Criteria |
| See Table 1 - SCP Return Codes for “copy” function Messages and Alerts | Command exit code | See Table 1 - SCP Return Codes for “copy” function Messages and Alerts | Platform dependent | Platform dependent |
| Post-copy:   * “IMPORTER - Unable to open or process file: {}”   + FileNotFoundException | Application log file | When a data file is copied into one of the ODE upload folders, ODE will try to open the file and process its content. This error message is logged when ODE fails to open the file due to file not being present. | If the file does not exist when ODE starts to process it or for some other reason cannot be opened for reading, this message is logged in the application log file. | File does not exist when ODE starts to process the file. |
| * “IMPORTER - Unable to open or process file: {}”   + SecurityException | Application log file | When a data file is copied into one of the ODE upload folders, ODE will try to open the file and process its content. This error message is logged when ODE fails to read the file due to lack of Java security privileges. | If a security manager exists and its checkRead method denies read access to the file, a message will be logged to the application log file.” | ODE does not have permission to process the file. |
| * “IMPORTER - Unable to open or process file: {}”   + "Error decoding data." | Application log file | When a data file is copied into one of the ODE upload folders, ODE will try to open the file and process its content. This error message is logged when ODE fails to decode the data from ASN.1 format. | If the message is not encoded to the expected ASN.1 encoding, ODE will raise this error to indicate failure to decode the data. | Message is encoded incorrectly. |

## ODE REST API

ODE exposes a RESTful API for use by clients for security, administrative and data functions. Standard HTTP/HTTPS verbs such as GET, POST, PUT, DELETE, etc., will be deployed for various functions.

1. host: ip:port
2. root context path: ode/api/rest
3. schemes:
4. - http
5. - https

ODE REST API documentation utilizes <http://swagger.io/> API documentation framework. The file containing the API documentation can be found in our GitHub repository’s docs folder: <https://github.com/usdot-jpo-ode/jpo-ode/blob/develop/docs/ODESwagger.yaml>. A Swagger UI server will be stood up in the future to host this file but for the time being you may copy the file contents to the free [Swagger Editor](http://editor.swagger.io/#!/) at <http://editor.swagger.io/#!/> in order to view the REST API specifications. Figure 2 illustrates the outcome of this process.



Figure - ODE REST API Editor Tool

### Upload BSM File

|  |  |
| --- | --- |
| **Path** | upload/<type: bsm (or) messageFrame> |
| **Verb** | POST |
| **Summary** | Service to upload a file containing BSM records |
| **Consumes** | - multi-part/file |
| **Produces** | - application/json |
| **Headers** | |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Value** | **Description** | **Required**  **(T | F)** | | Token | Value of token | Valid security token | True | |
| **Parameters** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Value** | **Description** | **Required** | **In**  **(Q|B)** | | type | bsm | messageFrame | The message format of BSM in uploaded file. | True | Q | |
| **Responses** | |  |  |  | | --- | --- | --- | | **Code** | **Description** | **Schema** | | **200** | Upload was successful | {Status:success} | | **HTTP Status code** | Description of the response | Schema of the response body | |

Table - Upload BSM File API

STATUS: A prototype upload BSM File API has been implemented but requires minor modifications to comply with the above specifications.

### MANAGE SNMP API

Over an SNMP Protocol, the ODE can ping and assess the health of an existing Road Side Unit to ensure the system is up and running. To trigger a specific heartbeat call, the ODE provides two separate interfaces to deploy a message to an RSU.

#### MANAGE SNMP API - Web Based View

To connect heartbeat service is through the existing web interface located at the root of the application. On it, a user will see a section for RSU SNMP Query and may enter in the same IP and OID information as the API Endpoint.



#### MANAGE SNMP API – REST Interface

To interface with the ODE through its REST API, the client needs to use the following parameters:

host: <Host server IP address or DNS name>

basePath: /rsuHeartbeat?ip=<ip\_address>&oid=<oid\_string>

You should receive a detailed plain text response that looks like the following example. If the device is off, a 4 second timeout will occur and the ODE will indicate this with an "[ERROR] Empty response" message. (This specific OID returns the amount of time since the device was last powered on)

[1.3.6.1.2.1.1.3.0 = 0:05:12.59]

### Traveler Information Message (TIM) Interface

Refer to the [ODESwagger.yaml](https://github.com/usdot-jpo-ode/jpo-ode/blob/develop/docs/ODESwagger.yaml) for details of the TIM interface.

Table - Outbound TIM Messages and Alerts

|  |  |  |  |
| --- | --- | --- | --- |
| Message or Alert | HTTP Response Code | Success | Criteria |
| [Response List] | 200 | true | Traveler Information Message was submitted successfully to one more specified RSUs and SDW. See the response for details of which RSUs or SDW received or failed to receive the message. |
| Endpoint received null request | 400 | false | Request body is empty. |
| Invalid Request Body | 400 | false | Request body is not a valid JSON object or not a valid TIM schema |
| Invalid Traveler Information Message data value in the request | 500 | false | ODE was unable to build a valid TIM message from the request data. One or more data elements may be invalid or out of range. |
| Internal Error: OssTravelerMessageBuilder#encodeTravelerInformationToHex returned null |  |  | Internal error occurred. Check application logs to determine cause. |
| Failed to encode Traveler Information Message | 400 | false | Internal Error: Unable to encode a TIM PDU |
| [Response List] | 500 | false | Did not success to send data to all specified RSUs. See the response body for details. Possible error messages in the response list include:   * No response from RSU IP=<RSU IP address> * SNMP deposit failed, error code=<SNMP error status code> (<SNMP error message>) |
| Error depositing to SDW | 500 | false | RSU responded with error or failed to send data to RSUs and SDW |

### Probe Data Management Messages (PDM) Interface

Refer to the [ODESwagger.yaml](https://github.com/usdot-jpo-ode/jpo-ode/blob/develop/docs/ODESwagger.yaml) for details of the PDM interface.

Table - Probe Data Management Messages and Alerts

|  |  |  |  |
| --- | --- | --- | --- |
| Message or Alert | HTTP Response Code | Success | Criteria |
| [Response List] | 200 | true | Traveler Information Message was submitted successfully to one more specified RSUs and SDW. See the response for details of which RSUs or SDW received or failed to receive the message. |
| Endpoint received null request | 400 | false | Request body is empty. |
| Invalid Request Body | 400 | false | Request body is not a valid JSON object or not a valid PDM schema |
| [Response List] | 500 | false | Did not success to send data to all specified RSUs. See the response body for details. Possible error messages in the response list include:   * No response from RSU IP=<RSU IP address> * SNMP deposit failed, error code=<SNMP error status code> (<SNMP error message>) |



## ODE Streaming API

ODE client applications will be able to subscribe to data streams via two distinct but dependent interfaces.

1. Clients may interface directly or through proxies with Kafka brokers to subscribe to a well-known topics. See section 7.3.1 for details.
2. Clients may Interface directly with ODE through ODE provided WebSocket interface as defined by RFC 6455 (<http://tools.ietf.org/html/rfc6455> ). See section 7.3.2 for details.

### Direct Kafka Interface

To interface with Kafka directly, the client needs to know the list of available Kafka brokers and the name of the topic that will contain the data. The client application may use any of the following methods to access Kafka topics:

* Native Kafka API (C, Java, Python, etc.)
* Kafka API RESTful Proxy such as: <https://www.confluent.io/blog/a-comprehensive-open-source-rest-proxy-for-kafka/>
* Kafka API WebSocket Proxy such as: <https://github.com/b/kafka-websocket/blob/master/pom.xml>

A sample Java client will be available in the ODE source repository under jpo-ode-consumer-example project.

#### Kafka Publish/Subscribe Topics

The following table contains the name of the topics and the type of data in each topic.

|  |  |  |
| --- | --- | --- |
| Topic Name | Data Type | Format |
| topic.J2735Bsm | us.dot.its.jpo.ode.plugin.j2735.J2735Bsm | Serialized POJO |
| topic.J2735BsmRawJSON | us.dot.its.jpo.ode.plugin.j2735.J2735Bsm | JSON |

Table - Kafka Publish/Subscribe Topics

### ODE WebSocket Interface

To interface with the ODE through its WebSocket interface, the client needs to know use the following parameters:

host: <Host server IP address or DNS name>

basePath: /ode/api/ws

schemes:

- ws

- wss

#### ODE WebSocket Control Messages

The response to a streaming data requests may contain control messages such as CONNECT, START, STOP, etc. Here is a sample Control Message.

{"metadata":{"payloadType":"control","version":1},"payload":{"dataSourceBundleCount":20,"receivedRecordCount":203,"sentRecordCount":10,"tag":"STOP","dataType":"Control","version":1},"version":1}

For detailed schema of ODE Control messages see section 8.2.6.

STATUS: Control Messages have not been implemented yet.

#### BSM WebSocket Subscription API

STATUS: A prototype BSM WebSocket Subscription API has been implemented but requires modifications to comply with the above specifications.

|  |  |
| --- | --- |
| **Path** | /bsm |
| **Summary** | Subscription request for BSM data |
| **Consumes** | - application/json |
| **Produces** | - application/json |
| **Parameters** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Value** | **Description** | **Required** | **In** | | request | Subscription Request (see section 8.1.1) | The subscription request parameters | true | body | | token | string | A valid security token obtained from login service | true | query-param | |
| **Responses** | |  |  |  | | --- | --- | --- | | **Code** | **Description** | **Schema** | | 0 | SUCCESS | J2735Bsm | | 1 | FAILURE | OdeStatus | | 2 | SOURCE\_CONNECTION\_ERROR | OdeStatus | | 3 | INVALID\_REQUEST\_TYPE\_ERROR | OdeStatus | | 4 | INVALID\_DATA\_TYPE\_ERROR | OdeStatus | |

## ODE Request Schemas

The following sub-sections describe the structure and specification of ODE request messages.

### ODE Request Message Metadata

Every ODE Request message component will contain the following data element.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Required / Optional (R | O)** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** | **Default** |
| version | O | Integer | The version number for this schema. See section 8 for details. | N/A | 0 (latest) | MAX INT | 0 |

Table – ODE Request Message Metadata

## ODE Response Schemas

The following sub-sections describe the structure and specification of ODE messages returned as response to data requests.

### ODE Data Message

STATUS: ODE DATA Messages have not yet been implemented.

The top level structure of ODE Data Message is defined in the table below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| metadata | OdeMsgMetadata | See section 8.2.2 |  |  |  |
| payload | One of [OdeMsgPayload](#_ODE_Data_Message) types. | See section 8.2.7 and its subsections |  |  |  |
| version | Integer | The version number for this schema. See section 8 for details. | N/A | 1 | MAX INT |

Table - OdeDataMessage

### ODE Message Metadata

STATUS: ODE Message Metadata has not yet been implemented.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| payloadType | string | One of: "status", "control", "security", "bsm", "tim", "map", "spat", "other" | enum |  |  |
| latency | Integer | Amount of time it took to process this record from the time it was received (based on receivedAt payload field) | Milliseconds | 0 | 9223372036854775807 |
| receivedAt | iso-date-time | The date and time that the payload paired with this metadata was received by the ODE. | ISO standard format include time zone: **yyyy-MM-ddThh:mm:ss.sssZ** |  |  |
| violations | Array of violation objects | This is an array of [OdePayloadViolation](#_ODE_Payload_Violation) objects. See section 8.2.3 for details. |  |  |  |
| serialId | String | A unique serial number representing this record. | N/A | N/A | N/A |
| version | Integer | The version number for this schema. See section 8 for details. | N/A | 1 | MAX INT |

Table – OdeMsgMetadata

### ODE Payload Violation

STATUS: ODE Payload Violations schema have not yet been implemented.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| actualValue | Decimal | The actual value of the field | The Unit of measure of the fieldName | Min Decimal | Max Decimal |
| fieldName | string | The name of the payload field for which violation was detected. | N/A |  |  |
| validMax | Decimal | Maximum value considered to be valid. | The Unit of measure of the fieldName |  |  |
| validMin | Decimal | Minimum value considered to be valid | The Unit of measure of the fieldName |  |  |

### ODE GET TOKEN Response

STATUS: ODE Get Token Response Message has not yet been implemented

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| dataType | string | Authorization response type | Enum | "security" | "security" |
| version | Integer | The version number for this schema. See section 8 for details. | N/A | 1 | MAX INT |

Table - OdeAuthentication

### ODE Status Message

STATUS: ODE Status Message has not yet been implemented

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| status | String | Request status | Enum | "status" | "status" |
| version | Integer | The version number for this schema. See section 8 for details. | N/A | 1 | MAX INT |

Table - OdeStatus

### ODE Control Message

STATUS: ODE Control Message has not yet been implemented

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| dataSourceBundleCount | Integer | Number of records in a bundle reported by the data source | Count | 0 | 9223372036854775807 |
| receivedRecordCount | Integer | Number of messages received by the ODE | Count | 0 | 9223372036854775807 |
| sentRecordCount | Integer | Number of records sent by the ODE to the client | Count | 0 | 9223372036854775807 |
| tag | Enum | The control message event tag | Enum | One of CONNECTED, START, STOP, ERROR | One of CONNECTED, START, STOP, ERROR |
| version | Integer | The version number for this schema. See section 8 for details. | N/A | 1 | MAX INT |

Table - OdeControlData

### ODE Data Message Payload

STATUS: ODE Data Message Payload schema has not yet been implemented

OdeMsgPayload is the parent class for all payload data structures, each of which contain the following data elements.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| version | Integer | The version number for this schema. See section 8 for details. | N/A | 1 | MAX INT |

Table – OdeMsgPayload

The following subsection describe the child schemas of [OdeMsgPayload](#_ODE_Data_Message_1).

#### J2735 BSM Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| coreData | [J2735BsmCoreData](#_J2735BsmCoreData) | See section 8.2.8.1 | N/A | N/A | N/A |
| partII | Array of [J2735BsmPart2Content](#_J2735BsmPart2Content) | See section 8.2.8.2 | N/A | N/A | N/A |
| version | Integer | The version number for this schema. See section 8 for details. | N/A | 1 | MAX INT |

### ODE Data Message Supporting Data Structures

The data structures described in this section and subsections are referenced by top level ODE data message schemas.

STATUS: Supporting data structures implementations are in progress and have not yet been completed for all supported data types.

#### J2735BsmCoreData

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| msgCnt | Integer |  |  |  |  |
| id | String |  |  |  |  |
| secMark | Integer |  |  |  |  |
| position | [J2735Position3D](#_J2735Position3D) |  |  |  |  |
| accelSet | [J2735AccelerationSet4Way](#_J2735AccelerationSet4Way) |  |  |  |  |
| accuracy | [J2735PositionalAccuracy](#_J2735PositionalAccuracy) |  |  |  |  |
| transmission | enum | One of:   * ***neutral***, // Neutral, speed relative to the vehicle alignment * ***park***, // Park, speed relative the to vehicle alignment * ***forwardGears***, // Forward gears, speed relative the to vehicle alignment * ***reverseGears***, // Reverse gears, speed relative the to vehicle alignment * ***reserved1***, ***reserved2***, ***reserved3***, ***unavailable***; // not-equipped or unavailable   // value, | | | |
| speed | decimal | Vehicle speed | m/s | 0 | 163.8 |
| heading | decimal | Vehicle heading | Degrees | 0.000 | 360.000 |
| angle | decimal | Steering wheel angle | Degrees | 0.000 | 360.000 |
| brakes | [J2735BrakeSystemStatus](#_J2735BrakeSystemStatus) |  |  |  |  |
| size | [J2735VehicleSize](#_J2735VehicleSize) |  |  |  |  |

#### J2735BsmPart2Content

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| id | enum | One of:   * vehicleSafetyExt * specialVehicleExt * supplementalVehicleExt |  |  |  |
| value | J2735BsmPart2Extension | One of the following object types:   * [J2735VehicleSafetyExtensions](#_J2735VehicleSafetyExtensions) * [J2735SpecialVehicleExtensions](#_J2735SpecialVehicleExtensions) * [J2735SupplementalVehicleExtensions](#_J2735SupplementalVehicleExtensions) |  |  |  |

#### J2735Position3D

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| elevation | decimal | Elevation | meters | -409.5 | 6143.9 |
| latitude | decimal | Position latitude | Degrees | -90.0000000 | +90.0000000 |
| longitude | decimal | Position longitude | Degrees | -180.0000000 | \*180.0000000 |

Table - OdePosition3D

#### J2735AccelerationSet4Way

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| accelLat | decimal | Latitudinal acceleration, i.e. acceleration perpendicular to the direction of travel. | m/s2 | -20.00 | +20.00 |
| accelLong | decimal | Longitudinal acceleration, i.e. acceleration in the direction of travel. | m/s2 | -20.00 | +20.00 |
| accelVert | decimal | Vertical acceleration, i.e. acceleration along the z-axis | G | -3.4 | +1.54 |
| accelYaw | decimal | Rotational acceleration around the z axis | Degrees/sec | -327.67 | +327.67 |

#### J2735PositionalAccuracy

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| semiMajor | decimal |  |  |  |  |
| semiMinor | decimal |  |  |  |  |
| orientation | decimal |  |  |  |  |

#### J2735BrakeSystemStatus

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| wheelBrakes | [J2735BitString](#_J2735BitString) | A Boolean dictionary of the following indicators:   * unavailable -- When set, the brake applied status is unavailable * leftFront -- Left Front Active * leftRear -- Left Rear Active * rightFront -- Right Front Active * rightRear -- Right Rear Active | | | |
| traction | enum | One of:   * unavailable -- B'00 Not Equipped with traction control or traction control status is unavailable * off -- B'01 traction control is Off * on -- B'10 traction control is On (but not Engaged) * engaged -- B'11 traction control is Engaged | | | |
| abs | enum | One of:   * unavailable -- B'00 Vehicle Not Equipped with ABS Brakes or ABS Brakes status is unavailable * off -- B'01 Vehicle's ABS are Off * on -- B'10 Vehicle's ABS are On ( but not Engaged ) * engaged -- B'11 Vehicle's ABS control is Engaged on any wheel | | | |
| scs | enum | One of:   * unavailable, -- B'00 Not Equipped with SC or SC status is unavailable * off, -- B'01 Off * on, -- B'10 On or active (but not engaged) * engaged -- B'11 stability control is Engaged | | | |
| brakeBoost | enum | One of:   * unavailable, -- Vehicle not equipped with brake boost or brake boost data is unavailable * off, -- Vehicle's brake boost is off * on -- Vehicle's brake boost is on (applied) | | | |
| auxBrakes | enum | One of:   * unavailable -- B'00 Vehicle Not Equipped with Aux Brakes or Aux Brakes status is unavailable * Off -- B'01 Vehicle's Aux Brakes are Off * on -- B'10 Vehicle's Aux Brakes are On ( Engaged ) * reserved -- B'11 | | | |

#### J2735VehicleSize

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| length | integer | Vehicle length | Centimeter | 0 | 16383 |
| width | integer | Vehicle Width | Centimeter | 0 | 1023 |

#### J2735BitString

A J2735BitString is represented by a Boolean dictionary, a collection of name-value pairs where the name identifies a Boolean indicator and the value represents the state of the indicator as *true* or *false*.

#### J2735VehicleSafetyExtensions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| events | [J2735BitString](#_J2735BitString) | A Boolean dictionary of the following indicators:   * eventHazardLights * eventStopLineViolation -- Intersection Violation * eventABSactivated * eventTractionControlLoss * eventStabilityControlactivated * eventHazardousMaterials * eventReserved1 * eventHardBraking * eventLightsChanged * eventWipersChanged * eventFlatTire * eventDisabledVehicle -- The DisabledVehicle DF may also be sent * eventAirBagDeployment | | | |
| pathHistory | [J2735PathHistory](#_J2735PathHistory) |  |  |  |  |
| pathPrediction | [J2735PathPrediction](#_J2735PathPrediction) |  |  |  |  |
| lights | [J2735BitString](#_J2735BitString) | A Boolean dictionary of the following indicators:   * lowBeamHeadlightsOn * highBeamHeadlightsOn * leftTurnSignalOn * rightTurnSignalOn * hazardSignalOn * automaticLightControlOn * daytimeRunningLightsOn * fogLightOn * parkingLightsOn | | | |

#### J2735SpecialVehicleExtensions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| vehicleAlerts | [J2735EmergencyDetails](#_J2735EmergencyDetails) |  |  |  |  |
| description | [J2735EventDescription](#_J2735EventDescription) |  |  |  |  |
| trailers | [J2735TrailerData](#_J2735TrailerData) |  |  |  |  |

#### J2735SupplementalVehicleExtensions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| classification | Integer |  |  |  |  |
| classDetails | [J2735VehicleClassification](#_J2735VehicleClassification) |  |  |  |  |
| vehicleData | [J2735VehicleData](#_J2735VehicleData) |  |  |  |  |
| weatherReport | [J2735WeatherReport](#_J2735WeatherReport) |  |  |  |  |
| weatherProbe | [J2735WeatherProbe](#_J2735WeatherProbe) |  |  |  |  |
| obstacle | [J2735ObstacleDetection](#_J2735ObstacleDetection) |  |  |  |  |
| status | [J2735DisabledVehicle](#_J2735DisabledVehicle) |  |  |  |  |
| speedProfile | [J2735SpeedProfile](#_J2735SpeedProfile) |  |  |  |  |
| theRTCM | [J2735RTCMPackage](#_J2735RTCMPackage) |  |  |  |  |
| regional | Array of [J2735RegionalContent](#_J2735RegionalContent) |  |  |  |  |

#### J2735PathHistory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| initialPosition | [J2735FullPositionVector](#_J2735FullPositionVector) |  |  |  |  |
| currGNSSstatus | [J2735BitString](#_J2735BitString) | A Boolean dictionary of the following indicators:   * unavailable -- Not Equipped or unavailable * isHealthy * isMonitored * baseStationType -- Set to zero if a moving base station, or if a rover device (an OBU), set to one if it is a fixed base station * aPDOPofUnder5 -- A dilution of precision greater than 5 * inViewOfUnder5 -- Less than 5 satellites in view * localCorrectionsPresent -- DGPS type corrections used * networkCorrectionsPresent -- RTK type corrections used | | | |
| crumbData | Array of [J2735PathHistoryPoint](#_J2735PathHistoryPoint) |  | | | |

#### J2735PathPrediction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| confidence | decimal | Confidence | percent | 0 | 100 |
| radiusOfCurve | decimal | Radius of curve | centimeters | 0 | TBD |

#### J2735EmergencyDetails

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| sspRights | Integer |  |  | 0 | 31 |
| events | [J2735BitString](#_J2735BitString) | A Boolean dictionary of the following indicators:   * peUnavailable -- Not Equipped or unavailable * peEmergencyResponse -- The vehicle is a properly authorized public safety vehicle, is engaged in a service call, and is currently moving or is within the roadway. Note that lights and sirens may not be evident during any given response call * Emergency and Non Emergency Lights related   + peEmergencyLightsActive   + peEmergencySoundActive   + peNonEmergencyLightsActive   + peNonEmergencySoundActive | | | |
| lightsUse | Enum | One of:   * ***unavailable***, * ***notInUse***, * ***inUse***, * ***yellowCautionLights***, * ***schooldBusLights***, * ***arrowSignsActive***, * ***slowMovingVehicle***, * ***freqStops*** | | | |
| multi | Enum | One of:   * ***unavailable***, * ***singleVehicle***, * ***multiVehicle***, * ***reserved*** | | | |
| responseType | Enum | One of:   * ***notInUseOrNotEquipped***, * ***emergency***, * ***nonEmergency***, * ***pursuit***, * ***stationary***, * ***slowMoving***, * ***stopAndGoMovement*** | | | |
| sirenUse | Enum | One of:   * ***unavailable***, * ***notInUse***, * ***inUse***, * ***reserved*** | | | |

#### J2735EventDescription

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| description | Array of integers | 1 to 8 ITIS.ITIScodes |  |  |  |
| extent | [J2735Extent](#_J2735Extent) |  |  |  |  |
| heading | [J2735BitString](#_J2735BitString) | A Boolean dictionary of the following indicators:   * from000-0to022-5degrees * from022-5to045-0degrees * from045-0to067-5degrees * from067-5to090-0degrees * from090-0to112-5degrees * from112-5to135-0degrees * from135-0to157-5degrees * from157-5to180-0degrees * from180-0to202-5degrees * from202-5to225-0degrees * from225-0to247-5degrees * from247-5to270-0degrees * from270-0to292-5degrees * from292-5to315-0degrees * from315-0to337-5degrees * from337-5to360-0degrees | | | |
| priority | Integer | The urgency of the message | N/A | 0 | 7 |
| typeEvent | Integer | A code from the list of ITIS.ITIScodes | N/A | 0 | 65535 |
| regional | Array of [J2735RegionalContent](#_J2735RegionalContent) |  |  |  |  |

#### J2735TrailerData

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| sspRights | Integer |  |  | 0 | 31 |
| connection | [J2735PivotPointDescription](#_J2735PivotPointDescription) |  |  |  |  |
| units | Array of [J2735TrailerUnitDescription](#_J2735TrailerUnitDescription) |  |  |  |  |

#### J2735VehicleClassification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| fuelType | Enum | One of:   * ***unknownFuel***, // FuelType::= 0 -- Gasoline Powered * ***gasoline***, // FuelType::= 1 * ***ethanol***, // FuelType::= 2 -- Including blends * ***diesel***, // FuelType::= 3 -- All types * ***electric***, // FuelType::= 4 * ***hybrid***, // FuelType::= 5 -- All types * ***hydrogen***, // FuelType::= 6 * ***natGasLiquid***, // FuelType::= 7 -- Liquefied * ***natGasComp***, // FuelType::= 8 -- Compressed * ***propane*** // FuelType::= 9 | | | |
| hpmsType | Enum | One of:   * ***none***, * ***unknown***, * ***special***, * ***moto***, * ***car***, * ***carOther***, * ***bus***, * ***axleCnt2***, * ***axleCnt3***, * ***axleCnt4***, * ***axleCnt4Trailer***, * ***axleCnt5Trailer***, * ***axleCnt6Trailer***, * ***axleCnt5MultiTrailer***, * ***axleCnt6MultiTrailer***, * ***axleCnt7MultiTrailer*** | | | |
| iso3883 | Integer | Iso3833VehicleType | N/A | 0 | 100 |
| keyType | Integer | BasicVehicleClass | N/A | 0 | 255 |
| responderType | Enum | One of:   * ***emergency\_vehicle\_units***, * ***federal\_law\_enforcement\_units***, * ***state\_police\_units***, * ***county\_police\_units***, * ***local\_police\_units***, * ***ambulance\_units***, * ***rescue\_units***, * ***fire\_units***, * ***hAZMAT\_units***, * ***light\_tow\_unit***, * ***heavy\_tow\_unit***, * ***freeway\_service\_patrols***, * ***transportation\_response\_units***, * ***private\_contractor\_response\_units*** | | | |

#### J2735VehicleData

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735WeatherReport

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| weatherAirPres | decimal |  | hectopascals (hPa) | 580.0 | 1090.0 |
| weatherAirTemp | decimal |  | Centigrade (Celcius) | -40.0 | 151.0 |

#### J2735WeatherProbe

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735ObstacleDetection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735DisabledVehicle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735SpeedProfile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735RTCMPackage

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735RegionalContent

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735FullPositionVector

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735PathHistoryPoint

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735PrivilegedEvents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735LightbarInUse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735MultiVehicleResponse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735ResponseType

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735SirenInUse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735Extent

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735PivotPointDescription

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735TrailerUnitDescription

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735BumperHeights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735TrailerHistoryPoint

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### J2735Node\_XY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table – BSM Data

#### OdeDateTime

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Units** | **Valid Min** | **Valid Max** |
| day | Integer |  |  |  |  |
| hour | Integer |  |  |  |  |
| minute | Integer |  |  |  |  |
| month | Integer |  |  |  |  |
| second | Decimal |  |  |  |  |
| year | Integer |  |  |  |  |

Table - OdeDateTime