Contract #: DTFH6116D00035

TOPR #: HOIT212116217

JPO
Operational Data Environment
User Guide

Submitted to:

U.S. Department of Transportation (USDOT)

Federal Highway Administration ITS JPO

January 17, 2017~~January 16, 2017~~~~January 14, 2017~~

Prepared by:

Booz Allen Hamilton
8283 Greensboro Drive
McLean, VA 22102

This page intentionally blank

Contents

Version History

Version #	Implemented By	Revision Date	What Changed?
1.0	Hamid Musavi		Initial draft

1 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.

~~All stakeholders are invited to provide input to this document. Stakeholders should direct all input on this document to the JPO Product Owner at DOT, FHWA, JPO.~~

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 ~~should~~ direct all input ~~on this document~~ to the JPO Product Owner at DOT, FHWA, JPO. To provide feedback, we recommend that you create an "issue" 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.

2 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.

3   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.***
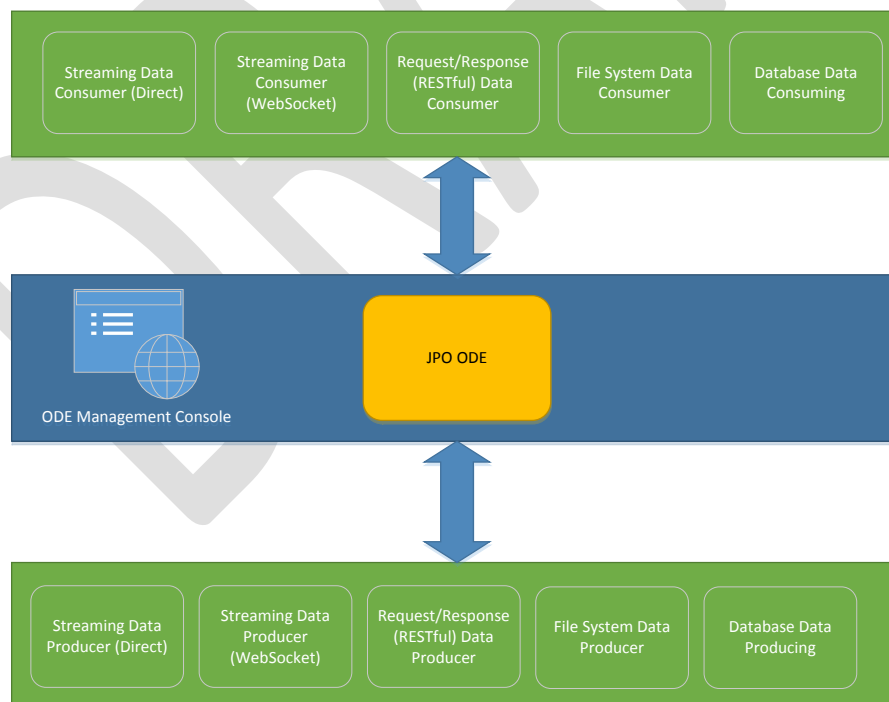

Figure 1 - ODE System Data Producers and Consumers

4 Audience

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

5 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

6 ODE DEVELOPMENT ENVIRONMENT

6.1 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

6.2 Java

Install Java Development Kit (JDK) 1.8

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

6.3 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.

6.4 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/

6.5 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.

6.6 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.

6.6.1 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.

6.6.2 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 ASN.1 compiler and runtime libraries.

6.6.2.1 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 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.

6.6.2.2 Build and Deploy Procedure

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

6.6.3 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*

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 instance parameter.

The following table describes all the ODE properties currently available.

Name	Default Value	Description
ode.kafkaBrokers	$DOCKER_HOST_IP:9092	List of kafka brokers and ports
ode.uploadLocation	uploads	Location of the shared directory where ODE monitors for files to ingest.

6.6.4 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.xml* 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.

7 ODE INTERFACES

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 0

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 0
- Streaming API: described in section 7.3.

All of the above interfaces will be secured using SSL encryption and role based authentication and authorization.

Note: Initial access to these interfaces is currently restricted to the ODE development team, but once sufficient testing is complete, any individual who successfully registers at the ODE Portal web site will be able to leverage the ODE interfaces. The word "individual" is intended to identify a person who will develop an application leveraging ODE data stream(s).

7.1 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 property ode.uploadLocation. ODE creates the specified directory if it does not already exist. The default location is a directory in the current working directory of the launched ODE application by the name "uploads".

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

7.2 ODE Admin API

ODE exposes a RESTful API for use by clients for security and other administrative functions 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
```

The following template will be used to specify the ODE API:

Path	Path relative to the root context path				
Verb	GET	POST	PUT	DELETE					
Summary	Summary description				
Consumes	Request data type such as application/json, application/text				
Produces	Response data type such as application/json, application/text				
Headers	**Name**	**Value**	**Description**		**Required (T | F)**
	Name of the header	Value of the header	Description of the header		True or False
Parameters	**Name**	**Value**	**Description**	**Required**	**In (Q|B)**
	Name of the parameter	Value of the parameter	Description of the parameter	True or False	Query or Body
Responses	**Code**	**Description**	**Schema**		
	HTTP Status code	Description of the response	Schema of the response body		
	HTTP Status code	Description of the response	Schema of the response body		

Table 1 - ODE API Specification Template

7.2.1 Get Security Token (Future)

STATUS: Get Security Token API has not yet been implemented

Path	/security/token			
Verb	GET			
Summary	Authenticates the user given username/password and returns a security token			
Consumes	- application/json			
Produces	- application/json			
Headers	Name	Value	Description	Required
	Authorization	<username>:<password>	The user name/password combination in the standard Basic HTTP Authentication format 'username:password'	True
Parameters	Name	Value	Description	Required	In (Q|B)
Responses	Code	Description	Schema	
	200 - OK	successful operation	Token	
	401 – File Not Found	Invalid username/password supplied	**See return response below.**	

Table 2 – Get Security Token API

15

If login fails due to invalid username/password, the following response will be returned to the client:

```
Status: error

Error Thrown: Unauthorized

Details: TBD
```

7.2.2 Revoke Security Token (Future)

STATUS: Revoke Security Token API has not yet been implemented.

Path	/security/token				
Verb	DELETE				
Summary	Revokes a security token				
Consumes	- application/json				
Produces	- application/json				
Headers	**Name**	**Value**	**Description**		**Required**
	Authorization	Bearer <token>	The authorization token to be revoked		True
Parameters	**Name**	**Value**	**Description**	**Required**	**In (Q|B)**
Responses	**Code**	**Description**	**Schema**		
	200 - OK	Successful Logout			
	401 - Unauthorized	Invalid token			

Table 3 – Revoke a Security Token API

7.2.3 Upload BSM File

Path	bsm/file				
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	bin | hex | base64	The type of data contained in the 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 4 - Upload BSM File API

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

7.3 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.

7.3.1 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.

7.3.2 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
```

7.3.2.1 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,"receivedR ecordCount":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.

7.3.2.2 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	

8 Schemas

The ODE API consists of data "request" and "responses". Section 8.1 specifies the structure of the data contained in the body of a "request". Section 8.2 specifies the structure of the data contained in the body of the "response" records.

All ODE request schemas will allow the inclusion of a *version* data element in the message. If new data elements are added or removed from the schema compared to a previous release of the ODE software, the version number will be incremented. Below is the specification of the *version* element within a request schema.

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

All ODE data response schemas will include a *version* data element in the message. If new data elements are added or removed from the schema compared to a previous release of the ODE software, the version number will be incremented. . Below is the specification of the *version* element within a data response schema.

Name	Required / Optional (R | O)	Type	Description	Units	Valid Min	Valid Max	Default
version	R	Integer	The version number for this schema.	N/A	1	MAX INT	N/A

STATUS: *version* element has not yet been implemented in the ODE API.

8.1 ODE Request Schemas

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

8.1.1 Subscription Data Request

STATUS: A prototype BSM WebSocket Subscription API does not currently require a request message. The client will be subscribed to BSM data automatically upon connection to the ODE WebSocket service. Additional modifications will be made at TBD time to provide a more flexible subscription service.

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

Table 5 – ODE BSM Subscription Request

8.2 ODE Response Schemas

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

8.2.1 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 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

22

Table 6 - OdeDataMessage

8.2.2 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 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 7 – OdeMsgMetadata

8.2.3 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		

8.2.4 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 8 - OdeAuthentication

8.2.5 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 9 - OdeStatus

8.2.6 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 10 - OdeControlData

8.2.7 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 11 – OdeMsgPayload

The following subsection describe the child schemas of OdeMsgPayload.

8.2.7.1 J2735 BSM Data

Name	Type	Description	Units	Valid Min	Valid Max
coreData	J2735BsmCoreData	See section 8.2.8.1	N/A	N/A	N/A
partII	Array of 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

8.2.8 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.

8.2.8.1 J2735BsmCoreData

Name	Type	Description	Units	Valid Min	Valid Max
msgCnt	Integer				
id	String				
secMark	Integer				
position	J2735Position3D				
accelSet	J2735AccelerationSet4Way				
accuracy	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				
size	J2735VehicleSize				

8.2.8.2 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 • J2735SpecialVehicleExtensions • J2735SupplementalVehicleExtensions			

8.2.8.3 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 12 - OdePosition3D

8.2.8.4 J2735AccelerationSet4Way

Name	Type	Description	Units	Valid Min	Valid Max

accelLat	decimal	Latitudinal acceleration, i.e. acceleration perpendicular to the direction of travel.	m/s^2	-20.00	+20.00
accelLong	decimal	Longitudinal acceleration, i.e. acceleration in the direction of travel.	m/s^2	-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

8.2.8.5 J2735PositionalAccuracy

Name	Type	Description	Units	Valid Min	Valid Max
semiMajor	decimal				
semiMinor	decimal				
orientation	decimal				

8.2.8.6 J2735BrakeSystemStatus

Name	Type	Description	Units	Valid Min	Valid Max
wheelBrakes	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			

8.2.8.7 J2735VehicleSize

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

8.2.8.8 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*.

8.2.8.9 J2735VehicleSafetyExtensions

Name	Type	Description	Units	Valid Min	Valid Max
events	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	
pathPrediction	J2735PathPrediction	
lights	J2735BitString	A Boolean dictionary of the following indicators: • lowBeamHeadlightsOn • highBeamHeadlightsOn • leftTurnSignalOn • rightTurnSignalOn • hazardSignalOn • automaticLightControlOn • daytimeRunningLightsOn • fogLightOn • parkingLightsOn

8.2.8.10   J2735SpecialVehicleExtensions

Name	Type	Description	Units	Valid Min	Valid Max
vehicleAlerts	J2735EmergencyDetails				
description	J2735EventDescription				
trailers	J2735TrailerData				

8.2.8.11   J2735SupplementalVehicleExtensions

33

Name	Type	Description	Units	Valid Min	Valid Max
classification	Integer				
classDetails	J2735VehicleClassification				
vehicleData	J2735VehicleData				
weatherReport	J2735WeatherReport				
weatherProbe	J2735WeatherProbe				
obstacle	J2735ObstacleDetection				
status	J2735DisabledVehicle				
speedProfile	J2735SpeedProfile				
theRTCM	J2735RTCMPackage				
regional	Array of J2735RegionalContent				

8.2.8.12   J2735PathHistory

Name	Type	Description	Units	Valid Min	Valid Max
initialPosition	J2735FullPositionVector				
currGNSSstatus	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`	

8.2.8.13   J2735PathPrediction

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

8.2.8.14   J2735EmergencyDetails

Name	Type	Description	Units	Valid Min	Valid Max
`sspRights`	`Integer`			0	31
`events`	`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
		    o peEmergencyLightsActive
		    o peEmergencySoundActive
		    o peNonEmergencyLightsActive
		    o 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*

8.2.8.15   J2735EventDescription

Name	Type	Description	Units	Valid Min	Valid Max
description	Array of integers	1 to 8 ITIS.ITIScodes			
extent	J2735Extent				
heading	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				

8.2.8.16   J2735TrailerData

Name	Type	Description	Units	Valid Min	Valid Max
sspRights	Integer			0	31
connection	J2735PivotPointDescription				
units	Array of J2735TrailerUnitDescription				

8.2.8.17   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*			

8.2.8.18   J2735VehicleData

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.19   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

8.2.8.20   J2735WeatherProbe

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.21   J2735ObstacleDetection

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.22   J2735DisabledVehicle

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.23   J2735SpeedProfile

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.24   J2735RTCMPackage

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.25   J2735RegionalContent

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.26   J2735FullPositionVector

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.27   J2735PathHistoryPoint

Name	Type	Description	Units	Valid Min	Valid Max

42

8.2.8.28   J2735PrivilegedEvents

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.29   J2735LightbarInUse

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.30   J2735MultiVehicleResponse

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.31   J2735ResponseType

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.32   J2735SirenInUse

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.33   J2735Extent

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.34   J2735PivotPointDescription

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.35   J2735TrailerUnitDescription

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.36   J2735BumperHeights

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.37   J2735TrailerHistoryPoint

Name	Type	Description	Units	Valid Min	Valid Max

8.2.8.38   J2735Node_XY

Name	Type	Description	Units	Valid Min	Valid Max

Table 13 – BSM Data

8.2.8.39   OdeDateTime

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

Table 14 - OdeDateTime