Sensor Data Ingestion Interface

Data Specification

Open Location Platform Version 3.3.1

Important Information
Notices

Topics:

This section contains document notices.

Legal Notices

Trademark Acknowledgements

HERE is trademark or registered trademark of HERE Global B.V.

Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

This content is provided "as-is" and without warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, satisfactory quality and non-infringement. HERE does not warrant that the content is error free and HERE does not warrant or make any representations regarding the quality, correctness, accuracy, or reliability of the content. You should therefore verify any information contained in the content before acting on it.

To the furthest extent permitted by law, under no circumstances, including without limitation the negligence of HERE, shall HERE be liable for any damages, including, without limitation, direct, special, indirect, punitive, consequential, exemplary and/ or incidental damages that result from the use or application of this content, even if HERE or an authorized representative has been advised of the possibility of such damages.

Document Information

Product	
Name:	Sensor Data Ingestion Interface
Version:	Open Location Platform Version 3.3.1
Document	
Name:	Sensor Data Ingestion Interface Data Specification
ID:	38390c6-1522935929-c3a3b542
Status:	FINAL
Date:	2018-Apr-05, 13:50 (GMT)

Contents

Chapter 1
Introduction

This document introduces the Sensor Data Ingestion Interface and:

- explains the main components of Sensor Data Ingestion Interface messages
- documents the structures in the protocol buffer (protobuf) based on the Sensor Ingestion Interface Specification.

Chapter 2
Message Components

Topics:

Sensor data packages are contained in `Messages` that may provide various type of content using *data elements*. The data elements may be grouped in different combinations.

A message may contain the following items:

- one (1) Envelope – Mandatory
- one (1) Path – Mandatory
- one or more Path Events – Optional
- one or more Path Media – Optional

When a change occurs, add the corresponding data element to the current Path as a Path Event. According to the type of content and the corresponding submission policy, either immediately submit the Path Events or collect them for later submission. For more information, see the relevant Developer's Guide.

Packages with multiple `Messages` can be combined and submitted as a single `Message` for better performance. These *multimessage* submissions consist of one or more `Message` elements. For more details about message submission, see the relevant Developer Guide.

For an example of a message, see *Sample Message* on page 14.

Envelope

The `Envelope` contains fundamental information about the individual sender (the vehicle). However, it does not contain enough information to identified the owner of a vehicle or to determine if different messages originate from the same vehicle.

For an example of an envelope structure, see *Sample Message* on page 14.

Path

A `Path` contains a sequential list of PositionEstimates of the same or different Position Types, ordered by timestamps.

Note: Paths must contain at least one Position Estimate.

Figure 1: Path with a number of Position Estimates

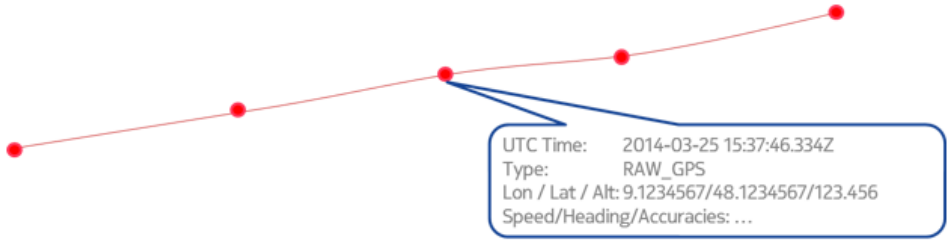
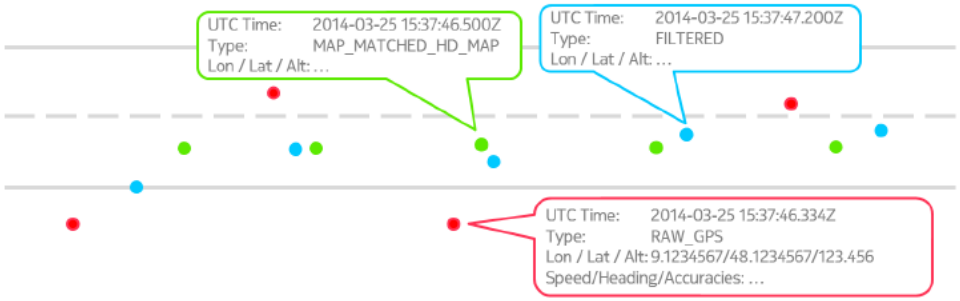


Figure 2: Path with different types of Position Estimates



A path can be very short or very long, depending on the *Path Events* that occur on the way:

- Near real time events are transmitted immediately after they occur and generate a very short path.

  Example: For a speed limit sign, a Sign Recognition Path Event is immediately created. The Path starts Position Estimates at a defined distance before the actual event and should include two Position Estimates after the event.
- If a drive lasts many hours, record the vehicle trace and events for later submission. This results in very long paths.

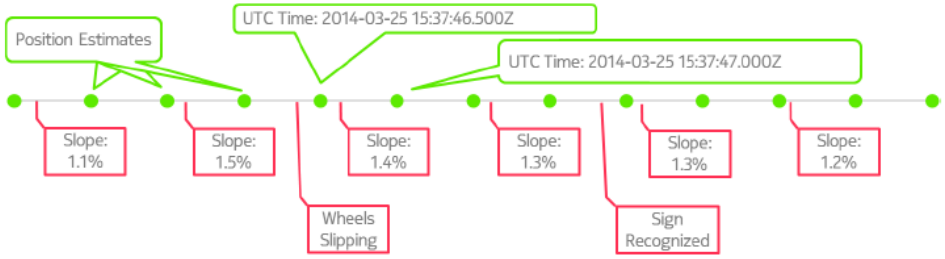
For an example of a path structure, see *Sample Message* on page 14.

Path Events

`Path Events` provide additional information along a path, as follows:

- Singular events - for example, a change of vehicle operation mode or information about an electronic stability program event.
- Continuously collected information - for example, velocity or curvature measurements.

Figure 3: Continuous Path Events and Singular Path Events



The `Path Events` list can contain Path Events of different types. Path Events are referencing the Path based on the timestamp of each Path Event. As this timestamp may not exactly match a timestamp in the Path, interpolation is required during processing of the information in the HERE Location Cloud.

Note: Path Events are not collected before the first Position Estimate in the Path and also not collected after the last Position Estimate in the Path.

`Path Events` are optional in the sensor data submission message. A message that contains only a Path has a meaning of its own without actual events. Nevertheless given that the HERE is supposed to collect rich sensor data, it is expected that a number of Path Events are typically included.

A `Path Event` is a complex data type, which may contain several mandatory and optional elements. You can configure the vehicles to report the status consistently using the `Path Event` elements.

Example:

A vehicle is reporting transmission mode and temperature as part of the `VehicleStatus` complex type. The vehicle must report this information at the very beginning of the path initially.

The following situations might occur:

- Transmission mode is changing - the vchicle reports another Path Event that indicates the new transmission mode in the `VehicleStatus` complex type.
- Temperature is changing - the vehicle reports another Path Event that indicates the temperature change in the `VehicleStatus` complex type.
- If the vehicle reports two Path Events at the exact same timestamp, then the information is merged into one combined Path Event, as long as both events do not have conflicting information.

For an example of a path event structure, see *Sample Message* on page 14.

Path Media

`PathMedia` provides additional media content along a `Path`. It may collect information that refers to certain `PathEvents`, but it may also include content requested at a certain location. Similar to `PathEvents`, `PathMedia` contains one or more instances of `MediaContainer`. Each `MediaContainer` can hold information for exactly one type of media, for example an image or a video clip of an image sensor.

Sample Message

The codeblock below illustrates a (one) sample Sensor Ingestion Interface Specification compliant message.


```
{
  "envelope": {
    "version": "1.0",
    "submitter": "HERE Test User",
    "vehicleMetaData": {
      "vehicleTypeGeneric": "PASSENGER_CAR",
      "vehicleSpecificMetaData": [
        { "key": "SpecificVehicleType", "value": "HERE Test Car" }
      ],
      "vehicleReferencePointDeltaAboveGround_m": 1.62,
      "curvatureAccuracy_1pm": 1e-005,
      "slopeAccuracy_percent": 0.01
    },
    "transientVehicleID": 12345
  },
  "path": {
    "positionEstimate": [
      {
        "timeStampUTC_ms": 1397764944000,
        "positionType": "RAW_GPS",
        "longitude_deg": 8.9743712,
        "latitude_deg": 49.4607534,
        "horizontalAccuracy_m": 1.3,
        "altitude_m": 123.7,
        "heading_deg": 76.09999999999999,
        "speed_mps": 13.8,
        "altitudeAccuracy_m": 24.5,
        "headingAccuracy_deg": 2.7,
        "speedAccuracy_mps": 0.7
      },
      {
        "timeStampUTC_ms": 1397764945000,
        "positionType": "RAW_GPS",
        "longitude_deg": 8.9748056,
        "latitude_deg": 49.4608278,
        "horizontalAccuracy_m": 1.4,
        "altitude_m": 123.6,
        "heading_deg": 72.515,
        "speed_mps": 13.4,
        "altitudeAccuracy_m": 25.7,
        "headingAccuracy_deg": 2.6,
        "speedAccuracy_mps": 0.6
      },
      {
        "timeStampUTC_ms": 1397764946000,
        "positionType": "RAW_GPS",
```

```
        "longitude_deg": 8.975258999999999,
        "latitude_deg": 49.4609312,
        "horizontalAccuracy_m": 1.7,
        "altitude_m": 123.5,
        "heading_deg": 69.405,
        "speed_mps": 13.1,
        "altitudeAccuracy_m": 33.3,
        "headingAccuracy_deg": 2.1,
        "speedAccuracy_mps": 0.8
      },
      {
        "timeStampUTC_ms": 1397764947000,
        "positionType": "RAW_GPS",
        "longitude_deg": 8.975643399999999,
        "latitude_deg": 49.4610356,
        "horizontalAccuracy_m": 3.8,
        "altitude_m": 130.522,
        "heading_deg": 67.526,
        "speed_mps": 13.2,
        "altitudeAccuracy_m": 50.0,
        "headingAccuracy_deg": 3.1,
        "speedAccuracy_mps": 1.7
      }
    ]
  },
  "pathEvents": {
    "vehicleDynamics": [
      {
        "timeStampUTC_ms": 1397764944000,
        "curvature_1pm": 0.002077,
        "slope_percent": -0.09900000000000001
      },
      {
        "timeStampUTC_ms": 1397764945000,
        "curvature_1pm": 0.001821,
        "slope_percent": -0.129
      },
      {
        "timeStampUTC_ms": 1397764946000,
        "curvature_1pm": 0.001298,
        "slope_percent": -0.145
      },
      {
        "timeStampUTC_ms": 1397764947000,
        "curvature_1pm": 0.00086,
        "slope_percent": -0.145
      }
    ],
    "signRecognition": [
      {
        "timeStampUTC_ms": 1397764945300,
        "roadSignType": "SPEED_LIMIT_START",
        "roadSignPermanency": "STATIC",
        "roadSignValue": "30",
        "roadSignDependencies": "SCHOOL"
      }
    ]
  }
}
```

Chapter 3
Data Elements

Topics:

All data elements have units according to the International System of Units (SI) unless otherwise explicitly noted.

The data types describe multiply used base and complex types.

- *VehicleMetaData.Vehicl...*
- *VehicleStatus*
- *VehicleStatus.DoorStat...*
- *VehicleStatus.Ventilati...*
- *Envelope.MapStandardEn...*
- *EnvironmentStatus.Light...*
- *EnvironmentStatus.Preci...*
- *EnvironmentStatus.Road...*
- *LaneBoundaryRecognition...*
- *LaneBoundaryRecognition...*
- *LaneBoundaryRecognition...*
- *LaneBoundaryRecognition...*
- *LaneBoundaryRecognition...*
- *LaneBoundaryRecognition...*
- *LocalizationInformation....*
- *LocalizationInformation....*
- *MediaContainer.MediaTy...*
- *ObjectDetection.Object...*
- *ObjectDetection.Object...*
- *ObjectDetection.Object...*
- *ObjectDetection.Object...*
- *ObjectDetection.Object...*
- *PassengerEnvironment.Me...*
- *PassengerEnvironment.Pa...*
- *PassengerEnvironment.Pa...*
- *PassengerEnvironment.Pa...*
- *PassengerEnvironment.Pa...*
- *PassengerEnvironment.Ro...*
- *PositionEstimate.Headin...*
- *PositionEstimate.LaneC...*
- *PositionEstimate.Positi...*
- *PositionEstimate.Speed...*
- *PositionOffset.Lateral...*
- *PositionOffset.Longitud...*
- *PositionOffset.Vertical...*
- *RoadAttributeRecognition...*
- *RoadAttributeRecognition...*
- *RoadBoundaryRecognition...*
- *RoadBoundaryRecognition...*
- *RoadBoundaryRecognition...*
- *RoadCondition.RoadRoug...*
- *RoadMarkingRecognition....*
- *SignRecognition.RoadSi...*
- *SignRecognition.RoadSi...*
- *SignRecognition.RoadSi...*
- *SignRecognition.RoadSi...*
- *SignRecognition.RoadSi...*
- *SignRecognition.RoadSi...*
- *SpecificObservedEvent....*
- *SpecificObservedEvent....*

ADServiceAndSensorState

Message Summary

message **ADServiceAndSensorState**

A complex datataype that holds information refering to Advanced Driving Systems describing their capabilities, current states and settings.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*int64*	required	Timestamp of the message
`adSpeedControl`	*bool*	optional	Contains the information if the vehicle is actively using a component that controls the speed of the vehicle.
`adBrakeControl`	*bool*	optional	Contains the information if the vehicle is actively using a component that controls the brakes of the vehicle.
`adSteeringControl`	*bool*	optional	Contains the information if the vehicle is actively using a component that controls the steering of the vehicle.
`adConnection Available`	*bool*	optional	Contains the information if the vehicle is able to use the an online connection or if the online connection is available.
`sensorObject Recognition`	*bool*	optional	Contains the information if the vehicle is able to detect object around the vehicle.
`sensorSign Recognition`	*bool*	optional	Contains the information if the vehicle is able to use the an online connection or if the online connection is available.
`sensorLane Recognition`	*bool*	optional	Contains the information if the vehicle is able to detect lanes.
`sensorRoadSurface`	*bool*	optional	Contains the information if the vehicle is able to detect the state of the road surface.
`sensorEnvironment`	*bool*	optional	Contains the information if the vehicle is able to detect environmental conditions as air temperature, precipitation, light.
`sensorParkingPilot Available`	*bool*	optional	Contains the information if the parking pilot sensor is available on the vehicle. A Parking Pilot is a device that allows the vehicle to park into a parking space automatically, or at least semi automatic by steering and/or accelerating and breaking, or giving guidance for steering actions. The Available flag is also disregarding the current state of the sensor (enabled or disabled)

Field	Type	Label	Description
sensorParkingSpot Available	*bool*	optional	Contains the information if the parking spot sensor is available on the vehicle. A Parking Pilot is a device that allows the vehicle to identify a free parking space with adequate measurement for the vehicle to park in. A parking Spot sensor is disregarding the functionality of automatically or semi automatically park in the vehicle. The Available flag is also disregarding the current state of the sensor (enabled or disabled)
sensorParkingSpot Enabled	*bool*	optional	Contains the information if the parking spot sensor is enabled on the vehicle. An enabled parking spot sensor is regularly checking the surrounding to provide information to the vehicle if a free parking spot is available. A free parking spot may be provided using the Road Boundary Attribution.
adParkingPilot Enabled	*bool*	optional	Contains the information if the vehicle is currently parking automated. An enabled Parking Pilot actually is parking the vehicle in. This is under the assumption, that the vehicle had found a fitting parking spot.
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

AntiLockBrakingSystemEvent

Message Summary

message **AntiLockBrakingSystemEvent**

DEPRECATED

Replaced by ElectronicStabilityControlEvent [deprecated = true]

Include: `sdii.v3.3.1.proto`

CrashDetectedEvent

Message Summary

message **CrashDetectedEvent**

This object within the ExceptionalVehicleState contains necessary attributes if a crash has been detected and stored in the object CrashDetectedEvent.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
vehicleIsDrivable	*bool*	optional	If the vehicles sensor decide that the vehicle is still drivable, the sensor data notifies on this event.
airbagsDeployed	*bool*	optional	This attribute contains the information if any of the airbags has been deployed.
eCallActivated	*bool*	optional	If the vehicle decides to activate an eCall, the sensor data notifies on this event.
vehicleIsObstacleOnRoad	*bool*	optional	If the vehicle detects that it is located on the road and a potential obstacle to other vehicles, this state is set.
maxAccelerationVector_mps2	*Vector3D*	optional	The maximum acceleration contains the 3 values of the 3D Vector. **Unit:** Meter per Second [m/s] in 3D | **Range:** -100..100m/s in 3D | **Resolution:** 0.1m/s

DynamicStabilityControlEvent

Message Summary

message **DynamicStabilityControlEvent**

DEPRECATED

Replaced by ElectronicStabilityControlEvent [deprecated = true]

Include: sdii.v3.3.1.proto

ElectronicStabilityControlEvent

Message Summary

message **ElectronicStabilityControlEvent**

This complex data type contains flags for any occurred event that is counter measured by the ESC.

Include: sdii.v3.3.1.proto

Properties

Field	Type	Label	Description
antiSlipActionEvent	*bool*	optional	Signaling if the wheels spin at acceleration and counter measures are active (e.g. throttling or de-clutching).
antiLockActionEvent	*bool*	optional	Signaling if the wheels block at braking and counter measures are active (e.g. ABS).

Field	Type	Label	Description
electronicStability ControlEvent	*bool*	optional	Signaling if the vehicle turns out of control and ESC-actions are active
requested AccelerationVector_ mps2	*Vector3D*	optional	Informs about the requested Acceleration Vector in contrast to the measured Acceleration Vector. **Unit:** Meter per Second [m/s] in 3D | **Range:** -100..100m/s in 3D | **Resolution:** 0.1m/s
requestedRotation RateVector_omega	*Vector3D*	optional	Informs about the requested Rotation Vector in contrast to the measured rotation Vector. The direction of positive values is describted in the object Vector3D. **Unit:** Angular Velocity rad/second [rad/s] | **Range:** -314..314rad/s| **Resolution:** 0.1rad/s

EmergencyBrakingEvent

Message Summary

message **EmergencyBrakingEvent**

When the vehicle is facing a braking event that goes beyond the comfortable behavior, this event is triggered. and transported

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
maxAcceleration Vector_mps2	*Vector3D*	optional	Contains a vector with the maximum acceleration in all three dimension referring to the vehicles lateral, longitudinal, and vertical axis. **Unit:** Meter per Second [m/s] in 3D | **Range:** -100..100m/s in 3D | **Resolution:** 0.1m/s

Envelope

Message Summary

message **Envelope**

The envelope includes fundamental information about the individual sender (the vehicle) but not to a level that owner of the vehicle can be identified or different messages can be identified that originate from a single vehicle.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`version`	*string*	required	A textual value describing the version of interface specification in use. For example: “3.3.2”
`submitter`	*string*	required	The name of the company that submits the data (e.g. an OEM name or a System Vendor name). It is not an individual vehicle identifier. The same company may use different values for the submitter text value for different purposes (e.g. production group, pre-development group). The submitter value is based on submitter-receiver agreement and may also be combined to enable distinguishment between different submitter environments (e.g. “OEM-HeadUnit”, “OEM-Cloud”, ...)
`vehicleMetaData`	*VehicleMetaData*	optional	Contains data referring to the vehicle that are not changing in usual driving conditions.
`transientVehicleID`	*int64*	optional	**DEPRECATED** The Transient Vehicle ID is a numeric and optional value. If the transient vehicle ID value is submitted then each submission during the drive cycle of a single vehicle shall receive the same transient vehicle ID value. This allows stitching together multiple smaller path submissions of a vehicle during a drive. A different drive of the same vehicle (e.g. on the next day) should receive a new transient vehicle ID. Within all submissions of a “Submitter” the ID space shall remain unique. This attribute is deprecated and replaced by transientVehicleUUID of data type string.
`vehicleProfileID`	*int64*	optional	**DEPRECATED** The Vehicle Profile ID is a numeric and optional value that is unique for a vehicle. If the vehicle Profile ID value is submitted then each submission including the profile ID is identifiable with the vehicle. This allows stitching together multiple single submissions of a vehicle during over a multitude of drives. The vehicle profile ID should be provided to events as e.g. fuel events where the fuel profile for a single vehicle should be analyzed and provided back to the single vehicle through a different interface. This attribute is deprecated and replaced by persistentVehicleUUID with the data type string.
`transientEventID`	*int64*	optional	Allows the vehicle to send in an optional identifier of the reported sensor data if it has it available. The Event ID may be used to a future referencing of the current Sensor Data Message.
`persistentDriverUUID`	*string*	optional	The Persistent Driver UUID is a string and an optional value that is unique for a driver in a vehicle. If the Persistent Driver UUID value is submitted then each submission including that ID identifies the driver who is assigned that ID. This allows stitching together multiple single submissions for a driver over a multitude of drives. The Persistent Driver UUID should be provided for events e.g. fuel events where the fuel profile for a single driver could be analyzed and provided back to the vehicle through a different interface.

Field	Type	Label	Description
persistentVehicleUUID	*string*	optional	The Persistent Vehicle UUID is a string and an optional value that is unique for a vehicle. If the Persistent Vehicle UUID value is submitted then each submission including that ID identifies the vehicle. This allows stitching together multiple single submissions for a vehicle over a multitude of drives. The Persistent Vehicle UUID should be provided for events e.g. fuel events where the fuel profile for a single vehicle could be analyzed and provided back to the vehicle through a different interface.
transientVehicleUUID	*string*	optional	The Transient Vehicle UUID is a string and an optional value. If the transient vehicle UUID value is submitted then each submission during the drive cycle of a single vehicle shall receive the same transient vehicle UUID value. This allows stitching together multiple smaller path submissions of a vehicle during a drive. A different drive of the same vehicle (e.g. on the next day) should receive a new transient vehicle UUID. Within all submissions of a "Submitter" the ID space shall remain unique.
submissionConfigurationUUIDArray	*string*	repeated	Using the Submission Configuration UUID, the system can group individual submissions of sensor data. One or more Submission Configurtion UUIDs can be sent along with the pertinent sensor data in a single message.
mapProvider	*string*	optional	The name of the provider of the map installed in the vehicle. This information is important in case, map matched information (linkID, map matched position). If provided, all three attributes Map Provider, Map Version, and Map Standard must be set.
mapVersion	*string*	optional	The version of the installed map used for providing information with map matched data.
mapStandard	*MapStandardEnum*	optional	The standard format of the map installed in the vehicle. If provided, all three attributes Map Provider, Map Version, and Map Standard must be set.
transientEventUUID	*string*	optional	The Transient Vehicle UUID is a string and an optional value. If the transient event UUID value is submitted, then each submission shall have a unique UUID value.
vehicleHeadUnitVersion	*string*	optional	The supplier based version of the vehicle Head Unit. It may contain a part for the hardware version, and a part for the software version.

EnvironmentStatus

Message Summary

message **EnvironmentStatus**

Container holidng information of the outside environment at a given time.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*int64*	required	Timestamp of the message
`lightConditions`	*LightConditionsEnum*	optional	Provides the current environmental light conditions according to environmental sensors.
`externalAir Temperature_DegC`	*double*	optional	Contains the degrees of the external air temperature **Unit:** Degree Centigrade [°C] | **Range:** -100..100°C | **Resolution:** 1°C
`externalAir TemperatureAccuracy _DegC`	*double*	optional	Contains the accuracy of the sensor measurement of the external Air Temperature. **Unit:** Degree Centigrade [°C] | **Range:** -100..100°C | **Resolution:** 1°C
`precipitation`	*PrecipitationEnum*	optional	Contains the identified or derived type of precipitation.
`visibleDistance_m`	*double*	optional	Contains the detected distance of visible light. **Unit:** Meter [m] | **Range:** 0..10000m | **Resolution:** 1m
`roadSurface Temperature_DegC`	*double*	optional	Contains the degrees of the road surface temperature. **Unit:** Degree Centigrade [°C] | **Range:** -100..100°C | **Resolution:** 1°C
`roadSurface TemperatureAccuracy _DegC`	*double*	optional	Contains the accuracy of the sensor measurement of the road surface temperature. **Unit:** Degree Centigrade [°C] | **Range:** -100..100°C | **Resolution:** 1°C
`roadSurfaceType`	*RoadSurfaceTypeEnum*	optional	Provides the identified road surface type below the vehicle
`airPressure_Pa`	*int32*	optional	The ambient air pressure arround the vehicle **Unit:** Pascal [Pa] | **Range:** 0..MAX Pa | **Resolution:** 1Pa
`airHumidity_percent`	*int32*	optional	The ambient air humidity arround the vehicle **Unit:** Percent [%] | **Range:** 0..100% | **Resolution:** 1%
`extensionContainer`	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

ExceptionalVehicleState

Message Summary

message **ExceptionalVehicleState**

The combination of information used to report exceptional vehicle states. These vehicle states are of rare nature and typically indicate a non-regular condition (e.g. tires slipping, crash detected, strong breaking).

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
timeStampUTC_ms	*int64*	required	Timestamp of the message
tireSlippage	*TireSlippageEvent*	optional	DEPRECATED: Handled within ElectronicStabilityControlEvent.
crashDetected	*CrashDetectedEvent*	optional	This Event contains relevant information about a detected crash.
emergencyBraking	*EmergencyBrakingEvent*	optional	This Event contains emergency braking information.
dynamicStability Control	*DynamicStabilityControl Event*	optional	**DEPRECATED** Handled within ElectronicStabilityControlEvent
antiLockBraking System	*AntiLockBrakingSystem Event*	optional	**DEPRECATED** Handled within ElectronicStabilityControlEvent
electronicStability Control	*ElectronicStabilityControl Event*	optional	Contains ESC events, as ABS, TC, ESP
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

ExtensionContainer

Message Summary

message **ExtensionContainer**

A container, that allows to provide additional content value in binary form identifying the content by a key of datatype string and a format defined by a specification denomination of datatype string.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
dataKey	*string*	optional	
dataSpecification	*string*	optional	
dataValue	*bytes*	optional	

KeyValuePairString

Message Summary

message **KeyValuePairString**

Parts of the interface are using generic key value pairs. Hence, a data type is defined that allows such key value pairs.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`key`	*string*	required	
`value`	*string*	required	

LaneBoundaryRecognition

Message Summary

message **LaneBoundaryRecognition**

Container holidng information of one recognized lane boundary (lane marking). The Position Offset data element provides positional information relative to the vehicle for the reported lane boundary. The Curvature data element is used to report lane boundary curvature in case this can be detected on board.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*int64*	required	Timestamp of the message
`positionOffset`	*PositionOffset*	required	Describes the relative position of the lane boundary in respect to the vehicle reference point and the vehicle reference axis.
`laneBoundaryType`	*LaneBoundaryTypeEnum*	optional	Describes the type of the recognized lane boundary
`laneBoundaryColor`	*LaneBoundaryColorEnum*	optional	Information about what color the lane marking has.
`curvature_1pm`	*double*	optional	The curvature as measured from the lane detection algorithm. Defining the curvature of the lane at the proximity of the vehicle. A positive value means a curvature to the right, a negative value means a curvature to the left. **Unit:** 1/Meter[1/m]| **Range:** -1..1/m| **Resolution:** 0.001/m
`laneMarkerWidth_mm`	*int32*	optional	The width of the detected lane marker at the proximity of the vehicle. **Unit:** Milimeter[mm]| **Range:** 0..1000mm| **Resolution:** 1mm
`laneMarkerWidth Accuracy_mm`	*int32*	optional	The accuracy of the lane Marker width depending on the quality of the sensor readings. **Unit:** Milimeter[mm]| **Range:** 0..1000mm| **Resolution:** 1mm

Field	Type	Label	Description
laneDeclination_deg	*double*	optional	The measured declination between the lane marker at the proximity of the vehicle and the vehicle's driving direction (it's view) in mathematical rotation direction (positive == the lane is rotated to the right in reference to the vehicle) **Unit:** Degree[°]| **Range:** -90..90° | **Resolution:** 0.1°
laneDeclination Accuracy_deg	*double*	optional	The accuracy of the measured lane declination as it can be biased by noise in the sensors (e.g. dirt on the road, bad weather ...) **Unit:** Degree[°]| **Range:** 0..90° | **Resolution:** 0.1°
laneBoundaryType Confidence_percent	*int32*	optional	An OEM internal confidence value providing the confidence that the recognized sign type is correct. **Unit:** Percent[%]| **Range:** 0..100%| **Resolution:** 1%
mapMatchedLaneID	*int64*	optional	The LaneID determined by the vehicle based on map matching. When mapMatchedLaneID is provided, it is required that the mapVersion, mapProvider and mapStandard are provided in the envelope.
laneBoundary RecognitionType	*LaneBoundary RecognitionTypeEnum*	optional	The lane boundary recognition type indicates whether a lane was detected or not. Not detecting a lane is meaningful information to report on whether lanes are no longer present.
laneBoundary RecognitionChange	*LaneBoundary RecognitionChangeEnum*	optional	The lane boundary recognition change indicates whether it was the start or the end of the lane boundary.
detectedObjectID	*int64*	optional	If the Sign Recognition is combined with an object Recognition, then here, the reference to the recognized object is given. The ID is unique inside of the SDII-message.
verticalCurvature_ 1pm	*double*	optional	The vertical curvature as measured (calculated) by the vehicle at the location of the vehicle of the slope of the road. A positive value means the slope is increasing, A negative value means the slope is decreasing. In alignment of the attribute slope, an increasing slope can be a change from steep downhill to flat terrain. **Unit:** per meter[1/m]| **Range:** -1..1/m| **Resolution:** 0.0001/m
verticalCurvature Accuracy_1pm	*double*	optional	The standard deviation of the vertical curvature measured by the vehicle. **Unit:** per meter[1/m]| **Range:** -1..1/m| **Resolution:** 0.0001/m
laneBoundary PositionReference	*LaneBoundaryPosition ReferenceEnum*	optional	The reference type for the recognized lane boundary.
laneBoundaryEgoLane Reference	*LaneBoundaryEgoLane ReferenceEnum*	optional	Description of identification of a lane boundary, if the lane marking is referenced to the ego-lane or other lanes.
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

LocalizationInformation

Message Summary

message **LocalizationInformation**

contains information about the positioning systems.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timestampUTC_ms`	*int64*	required	Timestamp of the message
`usedLocalization Systems`	*PositioningSystem*	repeated	Provides zero or more Description of positioning systems used for deriving a position.
`enhancedByDead Reconning`	*bool*	optional	In navigation, dead reckoning or dead-reckoning (also ded for deduced reckoning or DR) is the process of calculating one's current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course. This boolean field indicates if dead reckoning technique is applied for the current navigation system.
`enhancedByMap Localization`	*bool*	optional	To specify if map localization is enabled in the current navigation system.
`HDOP`	*double*	optional	The calculated horizontal dilution of precison as calculated by the GNSS device **Unit:** none [] | **Range:** 0..MAX | **Resolution:** 0.01
`VDOP`	*double*	optional	The calculated vertical dilution of precison as calculated by the GNSS device **Unit:** none [] | **Range:** 0..MAX | **Resolution:** 0.01
`extensionContainer`	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

LocalizationInformation.PositioningSystem

Message Summary

message **LocalizationInformation.PositioningSystem**

contains information about one specific positioning system.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
positioningSystem	*PositioningSystemTypeEnum*	optional	Provides the type of used system for the described positioning system object
gnssFix	*GNSSFixEnum*	optional	In a Global Navigation Satellite System (GNSS), A position fix (PF) is a position derived from measuring external reference points. A 2D fix gives only longitude and latitude. And A 3D fix gives full longitude latitude + altitude position.
numberOfSatellites	*int32*	optional	The number of satellites for the used GPS system.
enhancedByCellOrWifi	*bool*	optional	The Boolean value indicates that position of the GPS signal was enhanced using Celluar or Wifi Network data.
enhancedByDifferentialSignal	*bool*	optional	The Boolean value indicates that the position of the GPS signal was enhanced by usage of a differential signal (e.g. DGPS).

MediaContainer

Message Summary

message **MediaContainer**

Contains additional media data for cloud based analysis, if additional confidence evaluation is required.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
timeStampUTC_ms	*int64*	required	Timestamp of the message
mediaType	*MediaTypeEnum*	required	provides the transported type of media.
mediaFormat	*string*	required	Contains information about the Media Format in text form.
mediaContent	*bytes*	required	The data content
mediaID	*int64*	optional	Identifies the media throughout all media contents and allows to reference a media from other events as e.g. object recognition.
sensorOffset	*PositionOffset*	optional	Describes the relative position of the sensor providing the media information in respect to the vehicle and the vehicle reference axis.
sensorDirection	*Vector3D*	optional	A vector defining the direction of the sensor view in relation to the vehicle reference axis. **Unit:** Radians [rad] in 3D| **Range:** 0..2*PI in 3D | **Resolution:** 0.01 rad

Field	Type	Label	Description
duration_s	*int32*	optional	If the media content contains a temporal stream of information (e.g. video or audio)then the duration of the media record is contained in this attribute **Unit:** Second [s] | **Range:** 0..86400s | **Resolution:** 1ms
verticalViewingAngle_deg	*double*	optional	Vertical Viewing angle of the sensor **Unit:** Degree [°] | **Range:** 0..360° | **Resolution:** 0.1°
horizontalViewingAngle_deg	*double*	optional	Horizontal Viewing angle of the sensor **Unit:** Degree [°] | **Range:** 0..360° | **Resolution:** 0.1°

Message

Message Summary

message **Message**

The container containing one full message of Sensor Data Submission including mandatory Envelope, Path and optional PathEvents, and PathMedia

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
envelope	*Envelope*	required	
path	*Path*	required	
pathEvents	*PathEvents*	optional	
pathMedia	*PathMedia*	optional	

MessageList

Message Summary

message **MessageList**

Multiple sensor messages can be batched and submitted as a single SDII Message using the MessageList for better performance throughput. This type of submission is referred to as multi message submission.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
message	*Message*	repeated	

ObjectDetection

Message Summary

message **ObjectDetection**

Event container holding information about one detected object.

Include: sdii.v3.3.1.proto

Properties

Field	Type	Label	Description
timeStampUTC_ms	*int64*	required	Timestamp of the message
detectedObjectID	*int64*	optional	A vehicle unique identification of an object, if the vehicle is capable of following single object detection over time. “0” if no ID given.
positionOffset	*PositionOffset*	optional	: This complex object describes the position in reference to the vehicles reference point. For general description, see chapter 3.1.2.
movingVector_mps	*Vector3D*	optional	A moving 3D vector of the object referring to the vehicles reference axis lateral, longitudinal and vertical. **Unit:** Meter per second [m/s] in 3D| **Range:** -1000..1000m/s in 3D | **Resolution:** 1 m/s in 3D
objectType	*ObjectTypeEnum*	optional	Information about the recognized object type.
objectSizeVector_m	*Vector3D*	optional	A moving 3D vector of the object referring to the vehicles reference axis lateral, longitudinal and vertical. The size is a BBOX-value building a box from the Position Estimate point. By way of example, an object with the longitudinal offset of 100m and a length of 10m is located in the range of 100m to 110m. An object with the offset of 100m and a length of -10m is located in the range of 90m to 100m. **Unit:** Meter [m] in 3D| **Range:** -1000..1000m in 3D | **Resolution:** 0.01 m in 3D
objectSizeAccuracyVector_m	*Vector3D*	optional	When objects are not rectangular and detection can be imprecise, the accuracy of the detection is given here. This object is deprecated and replaced by the Accuracy in the complex structure of Size including accuracy. **Unit:** Meter [m] in 3D| **Range:** -1000..1000m in 3D | **Resolution:** 0.01 m

Field	Type	Label	Description
mediaID	*int64*	optional	When media content is provided with the object, a mediaID as reference can be provided with this object.
objectSurfaceType	*ObjectSurfaceTypeEnum*	optional	provides the surface type of the recognized object
objectSurface MaterialType	*ObjectSurfaceMaterial TypeEnum*	optional	provides the surface material of the recognized object
mapMatchedObjectID	*int64*	optional	The ObjectID determined by the vehicle based on map matching. When mapMatchedObjectID is provided, it is required that the mapVersion, mapProvider and mapStandard are provided in the envelope.
objectRecognition Type	*ObjectRecognitionType Enum*	optional	provides the type of recognition to the provided object message
objectCount	*int32*	optional	If a detection contains multiple objects with the same attributes (e.g. a group of people with same location == crowd), The total (estimated) number of objects of this recognized object can be provided aggregating the group of objects into one element. An estimate is sufficient.
objectRecognition MatchType	*ObjectRecognitionMatch TypeEnum*	optional	provides the matching type of the provided object
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

PassengerEnvironment

Message Summary

message **PassengerEnvironment**

Contains the information about driver and passenger in the vehicle.

Include: sdii.v3.3.1.proto

Properties

Field	Type	Label	Description
timeStampUTC_ms	*int64*	required	Timestamp of the message
generalPassenger Count	*int32*	optional	The passger number on the vehicle, discregarding the actual seating position.
passengerSeatUsage	*PassengerSeatUsage*	repeated	Provides information of zero or more seats and their current usage.
currentMediaSource	*MediaSourceEnum*	optional	provides the current used type of media source
radioFrequency_MHz	*double*	optional	Radio frequency if Media Source is RADIO (2). **Unit:** Megahertz [MHz] | **Range:** 0..2000MHz | **Resolution:** 0.1MHz

Field	Type	Label	Description
navigationSearch Destination	*string*	optional	If the infotainment map is in navigation mode and the vehicle is routing towards a specific location, the coordinates may be provided within the Environmental Container. The destination may be provided as text.
navigation Destination Longitude_deg	*double*	optional	If the infotainment map is in navigation mode and the vehicle is routing towards a specific location, this field provides the longitude of the destination. **Unit:** Degree [°] | **Range:** -180..180° | **Resolution:** 0.000001°
navigation DestinationLatitude _deg	*double*	optional	If the infotainment map is in navigation mode and the vehicle is routing towards a specific location, this field provides the latitude of the destination. **Unit:** Degree [°] | **Range:** -90..90° | **Resolution:** 0.000001°
navigationOrigin Longitude_deg	*double*	optional	If the infotainment map is in navigation mode and the vehicle is routing towards a specific location, the field may provide the origin position longitude coordinate attribute. **Unit:** Degree [°] | **Range:** -180..180° | **Resolution:** 0.000001°
navigationOrigin Latitude_deg	*double*	optional	If the infotainment map is in navigation mode and the vehicle is routing towards a specific location, the coordinates may provide the origin position latitude coordinate attribute. **Unit:** Degree [°] | **Range:** -90..90° | **Resolution:** 0.000001°
navigationAnd Routing_ETA_ms	*double*	optional	In Navigation mode, the estimated time of arrival at the destination location. The time is defined as timestamp (UTC) in miliseconds. For more detailas see common attribute type for timestampUTC_ms
navigationAnd Routing_ETT_ms	*int64*	optional	If the infotainment map is in navigation mode, the estimated travel time can be published to give indication of the duration of the travel. **Unit:** Milisecond [ms] | **Range:** 0..360000000ms | **Resolution:** 1ms
routingSelection PathType	*RoutingSelectionPath TypeEnum*	optional	provides the used routing type for the current route or drive
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.
timeDriving_min	*int32*	optional	provides the driving time since the last parking/stopping. **Unit:** Minute [minute] | **Range:** 0..MAXmin | **Resolution:** 0.1min
mostRecentParking Time_min	*int32*	optional	If the vehicle is in motion, and submitting sensor data messages, the most recent parking time (departure time – arrival time) can be provided in mostRecentParkingTime_min **Unit:** Minute [minute] | **Range:** 0..MAXmin | **Resolution:** 0.1min

PassengerEnvironment.PassengerSeatUsage

Message Summary

message **PassengerEnvironment.PassengerSeatUsage**

Contains information about the usage of one seat.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`simpleSeatRow`	*PassengerSeatRowSimpleEnum*	optional	Provides simple information on the seat row. With two rows, the rows front and rear, shall be used and with one row of passenger seats, the row front shall be used.
`simpleSeatColumn`	*PassengerSeatColumnSimpleEnum*	optional	Simple passenger seat column type. Having 2 columns per row, the columns left and right shall be used. With just 1 seat per row, the seat column middle shall be used.
`seatRowNumber`	*int32*	optional	The number of passenger seat row. If seatRowNumber is given, the simple row enumeration can be ignored. Row 1 is most front row.
`seatColumnNumber`	*int32*	optional	The number of passenger seat column. If seatColumnNumber is given, the simple column enumeration can be ignored. Column 1 is most left columns.
`seatUsage`	*PassengerSeatUsageEnum*	optional	provides the type of usage for the designated seat.
`seatOccupancyType`	*PassengerOccupancyTypeEnum*	optional	Provides the type of occupancy for the designated seat. If no distinguishment between light weight or heavy weight can be done, heavy weight shall be used as default.

Path

Message Summary

message **Path**

Path is the mandatory logical data type containing one or more position estimates of the vehicle ordered by timestamp. The position estimates are ordered starting with the oldest position estimate towards the newest position estimate. Two consecutive positions can be linearly interpolated, if the second point has not the attribute "firstPointAfterFixLoss" set to true. A path can be very short. For example, for near real time events that are transmitted immediately after they occur. A path can also be very long. For example, an entire drive over many hours that records the vehicle trace and events for later submission. A Path can have a mixture

of Position Estimates of different Position Types. For example, raw GPS positions can be included while map matched positions might only be included when available.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`positionEstimate`	*PositionEstimate*	repeated	The list of position estimates that make up the path

PathEvents

Message Summary

message **PathEvents**

Path Events provide additional information along a path. This may be single events, such as a change of vehicle operation mode or information about an electronic stability program event. It can also be continuously collected information such as velocity or curvature measurements. Path Events are of different types and can be mixed in the Path Events list. Path Events reference the Path based on the timestamp of each Path Event. This timestamp may not exactly match a timestamp in the Path, which means interpolation is required during processing of the information in the HERE cloud. Note that Path Events should not be collected before the first Position Estimate in the Path and also not collected after the last Position Estimate in the Path. Path Events are optional in the sensor data submission message. Onnly a Path is provided, which may have a meaning of its own without actual events. Nevertheless given that the HERE Sensor Data Ingestion interface is supposed to collection rich sensor data it is expected that a number of Path Events are typically included. Path Events are included in the sensor message as a list of Path Events. The Path Events are ordered starting with the oldest Path Event towards the newest Path Event. This means they are not grouped by type. A Path Event is typically a complex data type, which has a number of mandatory and optional components. Vehicles must be configured to report the components consistently. For example, if the vehicle reports transmission mode and temperature as part of the VehicleStatus complex type then it must report these two informations at the very beginning of the path initially. Whenever the transmission mode changes, another Path Event must be reported that indicates the transmission mode in the VehicleStatus complex type. Whenever temperature changes, another Path Event must be reported that indicates the temperature change in the VehicleStatus complex type. If by chance two Path Events are reported at the exact same timestamp, then all Information can be merged into one combined Path Events, as long as both events do not have conflicting information. The order within each list of Path Event Types is by timestamp ascending.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`vehicleStatus`	*VehicleStatus*	repeated	The list of zero or more vehicle status messages.

Field	Type	Label	Description
vehicleDynamics	*VehicleDynamics*	repeated	The list of zero or more vehicleDynamics messages.
signRecognition	*SignRecognition*	repeated	The list of zero or more sign recognition messages.
laneBoundary Recognition	*LaneBoundary Recognition*	repeated	The list of zero or more LaneBoundaryRecognition messages.
exceptionalVehicle State	*ExceptionalVehicleState*	repeated	The list of zero or more ExceptionalVehicleState messages
proprietaryInfo	*ProprietaryInfo*	repeated	The list of zero or more ProprietaryInfo messages.
environmentStatus	*EnvironmentStatus*	repeated	The list of zero or more Environmental Status messages.
objectDetection	*ObjectDetection*	repeated	The list of zero or more ObjectDetection messages.
adServiceAndSensor State	*ADServiceAndSensorState*	repeated	The list of zero or more ADServiceAndSensorState messages.
specificObserved Event	*SpecificObservedEvent*	repeated	The list of zero or more SpecificObservedEvent messages.
roadCondition	*RoadCondition*	repeated	The list of zero or more RoadCondition messages
roadBoundary Recognition	*RoadBoundary Recognition*	repeated	An sensor event that has specific Information about the road boundary recognition
vehicleManeuver	*VehicleManeuverEvent*	repeated	The information about the vehicle maneuver or movement.
localization Information	*LocalizationInformation*	repeated	An sensor event that has specific Information about the localization system of the vehicle
road Attributerecognition	*RoadAttributeRecognition*	repeated	An sensor event that has specific Information about road attributes
passenger Environment	*PassengerEnvironment*	repeated	The information covering the passenger environment status in the vehicle.
roadMarking Recognition	*RoadMarkingRecognition*	repeated	An event on recognized Markings on the road surface (lane)
trafficSignalHead Detection	*TrafficSignalHead Recognition*	repeated	The information of recognized Traffic Signal Heads. A traffic signal head is the group of lights (green, yellow, red).
proprietaryData Container	*ProprietaryDataContainer*	repeated	A message object, for proprietary usage.

PathMedia

Message Summary

message **PathMedia**

Path Media provide additional media content along a path. It can be collected information referring to certain Path Events, but also content requested at a certain location by a Cloud. Similar to Path Events, Path

Media does contain a repeated amount of Media Container. Each Container can hold exactly one media as e.g. an Image or a Video-Clip of an image sensor.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`mediaContainer`	*MediaContainer*	repeated	

PathSegment

Message Summary

message **PathSegment**

DEPRECATED

This message has no references.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`positionEstimate`	*PositionEstimate*	repeated	

PositionEstimate

Message Summary

message **PositionEstimate**

The position estimate data type combines information related to the position of a vehicle and its derived attributes. A position estimate contains at least the coordinates of the estimated position and optionally additional attributes.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*int64*	required	Timestamp of the message
`positionType`	*PositionTypeEnum*	required	The type of the position of this message.

Field	Type	Label	Description
interpolatedPoint	*bool*	optional	The interpolated point flag is used to flag points in the path that were created artificially by interpolating “real” measurements.
longitude_deg	*double*	required	Longitude is part of an absolute position. Longitude is used for different position types, such as raw GPS position, fused position based on absolute and inertial measurements (dead reckoning), and map matched position. Reference system is WGS84. **Unit:** Degrees [°] | **Range:** -180..180° | **Resolution:** 0.00001°
latitude_deg	*double*	required	Latitude is part of an absolute position. Longitude is used for different position types, such as raw GPS position, fused position based on absolute and inertial measurements (dead reckoning), and map matched position. Reference system is WGS84. **Unit:** Degrees [°] | **Range:** -90..90° | **Resolution:** 0.00001°
horizontalAccuracy_m	*double*	required	The calculated standard deviation for the horizontal position (longitude and latitude combined). **Unit:** Meter [m] | **Range:** 0..10000m | **Resolution:** 0.01m
altitude_m	*double*	optional	Altitude is the distances of the position of the WGS84 reference ellipsoid and not based on a mean sea level. The altitude is expected to be the altitude at street level, not the altitude of the position of the GPS antenna. Alternatively a different altitude can be provided, e.g. altitude at the roof of the vehicle but then an altitude offset compared to the street level needs to be provided in the vehicle meta data. **Unit:** Meter [m] | **Range:** -1000..9000m | **Resolution:** 0.01m
heading_deg	*double*	optional	The heading of the vehicle. Clockwise measured from North (0 deg), East (90 deg). **Unit:** Degree [°] | **Range:** 0..360° | **Resolution:** 1°
speed_mps	*double*	optional	The speed of the vehicle. Speed for raw GPS position has no sign. Other positions are required to be signed where positive means forward and negative means backwards. **Unit:** Meter per second [m/s] | **Range:** -150..150m/s | **Resolution:** 0.1 m/s
altitudeAccuracy_m	*double*	optional	The calculated standard deviation for the altitude **Unit:** Meter | **Range:** 0..10000m | **Resolution:** 0.01m
headingAccuracy_deg	*double*	optional	The calculated standard deviation for the heading. **Unit:** Degree [°] | **Range:** 0..360° | **Resolution:** 1°
speedAccuracy_mps	*double*	optional	The calculated standard deviation for the vehicle speed. **Unit:** Meter per second [m/s] | **Range:** 0..150m/s | **Resolution:** 0.1m/s
speedDetectionType	*SpeedDetectionEnum*	optional	The type of speed detection technology used.
headingDetectionType	*HeadingDetectionEnum*	optional	The type of heading detection technology used.

Field	Type	Label	Description
vehicleReferencedOrientationVector_rad	*Vector3D*	optional	Information of the vehicles orientation from the reference horizontal north position as yaw/roll/pitch-rotation value. **Example:** A vector of {lateral=-PI/6,longitudinal=PI/4, vertical=PI} corresponds to a rotation of 90° to the right (EAST), a roll of 45° along the longitudinal axis (leaning left) and a pitch of -30° along the lateral axis (pointing upwards). **Unit:** Radians | **Range:** 0..2*PI for each dimensions | **Resolution:** 0.01 rad
currentLaneEstimate	*int32*	optional	The currnet lane number as estimation according to the vehicle internal sensors If no Lane Count Direction is given, then the default counting direction is starting with lane 0 at the most inner lane (left on right hand traffic and right on left hand traffic. **Unit: #** | **Range:** 0..20 | **Resolution:** 1
mapMatchedLinkID	*int64*	optional	The linkID on the map that is installed in the vehicle to which the position matched to.
mapMatchedLinkIDConfidence_percent	*int32*	optional	The confidence value determining the quality and the belief in the correctness of the provided map matched linkID. **Unit:** Percent(%) | **Range:** 0..100% | **Resolution:** 1
mapMatchedLinkIDOffset_m	*double*	optional	The offset on the map matched link where the position lies. **Unit:** Meter | **Range:** -100..100m | **Resolution:** 0.01m
laneCountDirection	*LaneCountDirectionEnum*	optional	The direction the lane was counted in.
currentLaneEstimateConfidence_percent	*int32*	optional	The Confidence Value determining the quality and the belief in the correctness of the current lane estimation. **Unit**: Percent | **Range:** 0..100% | * **Resolution:** 1
firstPointAfterFixLoss	*bool*	optional	The indicator indicates that the position estimate is the first after a signal loss. The path between the preceeding and current point may not be interpolated linearly. This can occure e.g. between tunnel entry and tunnel exit, or in other occasions when no position estimate can be done.
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

PositionOffset

Message Summary

message **PositionOffset**

The position offset describes a position delta from the vehicle reference point as defined in the vehicle metadata.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`lateralOffset_m`	*double*	optional	The lateral offset value is used to describe a distance to the side of the vehicle from the vehicle reference point (which is the absolute position of the vehicle). A positive value is to the right of the vehicle in driving direction. **Unit:** Meter [m] | **Range:** -1000..1000m| **Resolution:** 0.01 m
`lateralOffsetSimple`	*LateralOffsetSimpleEnum*	optional	
`longitudinalOffset_m`	*double*	optional	The longitudinal offset value is used to describe a distance in front (positive) or to the back of the vehicle from the vehicle reference point. **Unit:** Meter [m] | **Range:** -1000..1000m| **Resolution:** 0.01 m
`longitudinalOffsetSimple`	*LongitudinalOffsetSimpleEnum*	optional	
`verticalOffset_m`	*double*	optional	The vertical offset value is used to describe a distance above or below the vehicle from the vehicle reference point. **Unit:** Meter [m] | **Range:** -1000..1000m| **Resolution:** 0.01 m
`verticalOffsetSimple`	*VerticalOffsetSimpleEnum*	optional	Enum to describe the vertical offset in simple categories (above, at level, below)
`lateralOffsetAccuracy_m`	*double*	optional	The lateral offset accuracy value is used to describe the confidence of the lateral offset. **Unit:** Meter [m] | **Range:** -1000..1000m| **Resolution:** 0.01 m
`longitudinalOffsetAccuracy_m`	*double*	optional	The longitudinal offset accuracy value is used to describe the confidence of the longitudinal offset **Unit:** Meter [m] | **Range:** -1000..1000m| **Resolution:** 0.01 m
`verticalOffsetAccuracy_m`	*double*	optional	The vertical offset accuracy value is used to describe the confidence of the vertical offset **Unit:** Meter [m] | **Range:** -1000..1000m| **Resolution:** 0.01 m

ProprietaryDataContainer

Message Summary

message **ProprietaryDataContainer**

The proprietary Data Container message is an empty message, that is reserved for individual proprietary specification by the using party in branched and individually updated versions of the specification. reserved for use of proprietary specified data

Include: `sdii.v3.3.1.proto`

ProprietaryInfo

Message Summary

message **ProprietaryInfo**

The combination of information to report OEM proprietary information. The proprietary data is submitted as key value pairs in string form. Should binary data need to be transmitted it may be converted using Base64 or other technologies in order to utilize the string key value pairs.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*int64*	required	Timestamp of the message
`keyValuePairs`	*KeyValuePairString*	repeated	
`extensionContainer`	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

RoadAttributeRecognition

Message Summary

message **RoadAttributeRecognition**

Contains the information about the recognition results of road attributes. A vehicle can identify the attribution of a road without the specific collection of individual sensor information. This attribution can be provided as aggregated environmental attribute.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timestampUTC_ms`	*int64*	required	Timestamp of the message
`roadAttribute`	*RoadAttributeTypeEnum*	optional	
`roadLaneCount_count`	*int32*	optional	The number of the lanes on the road
`roadLaneCount Confidence_ percentage`	*int32*	optional	This percentage value indicates the confidence of the road lane count **Unit:** Percent [%] | **Range:** 0..100% | **Resolution:** 1%

Field	Type	Label	Description
roadLaneDetectionCount_count	*int32*	optional	The number of the lanes on the road, as detected. It might be less than the real road lane count. Number of detected lanes(not actual number of lanes)
roadAttributeReference	*ReferenceTypeEnum*	optional	The reference type of the recognition of the current road attribute.
referencedValue_m	*int32*	optional	The reference value describes the distance in meter at which the provided attribute has began or ended, if the Attribute object is provided at the time when the start or end of the attribute is been defined as confident. If the value of road attribute reference type is START_BEFORE (2) or ENDED_BEFORE (3), the reference value should be specified. **Unit:** Meter [m] | **Range:** 1..MAXm | **Resolution:** 1m if ReferenceTypeEnum is 2 or 3;
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

RoadBoundaryRecognition

Message Summary

message **RoadBoundaryRecognition**

Container holidng information of one recognized road boundary. The Position Offset data element provides positional information relative to the vehicle for the reported road boundary. The Curvature data element is used to report road boundary curvature in case this can be detected on board.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
timeStampUTC_ms	*int64*	required	Timestamp of the message
positionOffset	*PositionOffset*	repeated	Describes the zero or more relative positions of the road boundary in respect to the vehicle and the vehicle reference axis.
roadBoundaryType	*RoadBoundaryTypeEnum*	optional	Providing the type of recognized road boundary
roadBoundaryRecognitionType	*RoadBoundaryRecognitionTypeEnum*	optional	provides the recognition type of the road boundary object
roadBoundaryChangeType	*RoadBoundaryChangeTypeEnum*	optional	
detectedObjectID	*int64*	optional	If the Sign Recognition is combined with an object Recognition, then here, the reference to the recognized object is given.
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

RoadCondition

Message Summary

message **RoadCondition**

Road Conditions contains all information regarding the road infrastructure that is not defined in more detail in other Containers (e.g. Sign Recognition or Lane Recognition). Typically, road condition information is measured by the vehicle using on board sensors at high frequency compared to positions (e.g. 5Hz or 10Hz). Depending on the provided attribute road condition events can be provided on a “on change”-bases valid for a long road segment until next event is provided or on a “occurrence”-base only valid for the particular location of the provided event.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*int64*	required	Timestamp of the message
`roadRoughness SegmentLevel`	*int32*	optional	This value provides the smoothness / roughness of the road along a segment. Road Roughness is provided in levels from 1 to 7. Sensor Values are highly subjective (suspension configuration, Tire pressure, vehicle type,...) and therefore more than the raw value is needed, here. A more detailed definition of roughness levels to be defined in a later version of the specification. The segment is defined by a starting point “current timestamp of the road condition event – Road Roughness Segment Duration” and an ending point “current timestamp of the Road Condition Event”. The Level is defined by the International Roughness Index
`roadRoughness SegmentDuration_ms`	*int64*	optional	Defines the duration since when the Road Roughness Segment Level is prevailing. **Unit:** Milisecond [ms] | **Range:** 1..MAXms | **Resolution:** 1ms Defines the length of the segment of the road where the roughness is present **Unit:** Meter [m] | **Range:** 0..MAXm | **Resolution:** 1m
`roadRoughness SegmentLength_m`	*int32*	optional	
`roadRoughnessLocal Event`	*bool*	optional	Defines a change in the Road Roughness that is only limited on the geometric extend and can be caused by a speed bump or a pothole. By setting this field to True, a local road roughness level temporarily overrides a prior sent “Global” road roughness segment level for the duration of setting the field.
`roadRoughness LateralPosition`	*RoadRoughnessLateral PositionEnum*	optional	Specifies to location of road roughness referenced to the vehicle (left and right side)

Field	Type	Label	Description
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

RoadMarkingRecognition

Message Summary

message **RoadMarkingRecognition**

A message containing information on a painted road marking on the road.

Include: sdii.v3.3.1.proto

Properties

Field	Type	Label	Description
timeStampUTC_ms	*double*	required	Timestamp of the message
roadMarkingType	*RoadMarkingTypeEnum*	optional	Contains the type of the identified roadmarking
positionOffset	*PositionOffset*	optional	Describes the relative position of the road marking in respect to the vehicle and the vehicle reference axis.
laneMarkingSize	*Vector3D*	optional	Describes the size of the road marking in 3D vectors according to the vehicles reference axis
laneReferenceID	*int32*	optional	Describes a local laneID the road marking has been identified on.
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

SignRecognition

Message Summary

message **SignRecognition**

Container holidng information of one recognized traffic sign. The Position Offset data element provides positional information relative to the vehicle for the reported traffic sign.

Include: sdii.v3.3.1.proto

Properties

Field	Type	Label	Description
timeStampUTC_ms	*int64*	required	Timestamp of the message

Field	Type	Label	Description
positionOffset	*PositionOffset*	optional	Describes the relative position of the traffic sign in respect to the vehicle and the vehicle reference axis.
roadSignType	*RoadSignTypeEnum*	required	describes the type of road sign.
roadSignPermanency	*RoadSignPermanency Enum*	optional	Enumeration used to indicate the permanency of a sign (static sign versus a variable (electronic) message sign).
roadSignValue	*string*	optional	The value of a sign. For example a speed limit sign has a speed limit value. The sign value does not have any unit but does reflect the text, that is written on the sign. **Example:** Speed limit shows '30', value="30" Speed limit shows '45 mph', value="45 mph" Maximum height '12"', value = "12"" Maximum width '2,2m', value = "2,2m" Maximum weight '12 to', value = "12 to"
roadSign Dependencies	*RoadSignDependencies Enum*	optional	
roadSignValidity	*RoadSignValidityEnum*	optional	
roadSignValidity Value	*double*	optional	The optional value of a validity sign. Provided with the sign itself.
roadSignRecognition Type	*RoadSignRecognition TypeEnum*	optional	The sign event type indicates whether a sign was detected or a sign was not detected. Not detecting a sign is a meaningful information to report signs that are no longer present.
detectedObjectID	*int64*	optional	If the Sign Recognition is combined with an object Recognition, then here, the reference to the recognized object is given. The Object ID is for message internal references between objects of same or different type of Path Events.
mediaID	*int64*	optional	If the Sign Recognition is provided with a media content then here, the reference to the media is given. The Media ID is for message internal references between the object and an optional provided media content.
signRecognition Confidence_percent	*int32*	optional	The Confidence Value determining the detection quality and the belief in the correctness of the provided Sign Recognition **Unit:** Percent(%) | **Range:** 0..100% | **Resolution:** 1
mapMatchedSignID	*int64*	optional	The SignID determined by the vehicle based on map matching. When mapMatchedSignID is provided, it is required that the mapVersion, mapProvider and mapStandard are provided in the envelope.
unspecifiedSignType	*string*	optional	In the case, where the list above for road Sign Type does not capture the type of the sign detected, the provider can use this field to send that information.
size3D	*Vector3D*	optional	The size of the sign measured in length, breadth and width. **Unit:** Meter [m] in 3D| **Range:** 0..MAX m in 3D| **Resolution:** 0.01m in 3D
signShape	*RoadSignRecognition ShapeEnum*	optional	Describes the shape Shape of the recognized sign

Field	Type	Label	Description
rotation	*Vector3D*	optional	The rotation of the sign in degrees measured in counterclockwise in x, y and z planes. The value 0,0,0 means, that the sign plane is perpendicular (90°) to the view and no distortion to the sign is experienced. If viewing the sign from in an angle of 45° from the right, a value of 45° along the vertical Axis is provided. The sign is rotated by 45° counterclockwise along the horizontal (bottom-up) axis according to ENU-definition. **Unit:** Degree [°] in 3D| **Range:** 0..360° in 3D| **Resolution:** 1° in 3D
roadSignDependencyText	*string*	optional	The dependency text indicated on the sign or a supplemental plaque. This type is used if an appropriate dependency cannot be mapped to the Specification due to a variety of content of supplemental dependencies. Specified values are: **Time dependency**: A time based dependency where the sign is valid between two given times of the day is formatted as "HHMM-HHMM" in 24h format where one or more spaces may be before and behind the dash "-" but no spaces or additional characters within the 4 digit time block.
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

SpecificObservedEvent

Message Summary

message **SpecificObservedEvent**

Container holidng information about one observed event derived from sensor data.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
timeStampUTC_ms	*int64*	required	Timestamp of the message
cause	*SpecificObservedEventCauseEnum*	optional	
subcause	*SpecificObservedEventSubCauseEnum*	optional	
relevanceTrafficDirection	*RelevanceTrafficDirectionEnum*	optional	
relevanceEventReference	*RelevanceEventReferenceEnum*	optional	
relevanceDistance	*RelevanceDistance*	optional	

Field	Type	Label	Description
eventTimeToLive	*int64*	optional	Event time to live **Unit:** Milisecond [ms] | **Range:** 1..MAXms | **Resolution:** 1ms
specificObservedEventConfidence_percent	*int32*	optional	The Confidence Value determining the detection quality and the belief in the correctness of the provided specific observed event **Unit:** Percent [%] | **Range:** 0..100% | **Resolution:** 1%
negotiationFlag	*NegotiationTypeEnum*	optional	Provides the actual type of negotiation of the event object
eventID	*string*	optional	The Event UUID is a string and an optional value that is unique for a detected Event and shall stay persistent within the vehicle. This ID is used to identify and resubmit information on a specific event (e.g. cancelation flag). An event shall be uniquely identified over the fleet with detection UUID and a session, or vehicle ID
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

SpecificObservedEventSubCauseEnum

Message Summary

message **SpecificObservedEventSubCauseEnum**

Sensor Data providing client application may use this event type to send information about events recorded in the vehicular environment. This event may not always be originating from the sensors built into the vehicle (e.g., events detected by cameras traveling with the vehicle or any other environmental test equipment, as well as events entered by the user using a communications device (“crowd-sourced information”)). Events are described by assignment of event cause and subcause values, which must align with ETSI ITS Specification [1].

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
subcause	**One Of**		
trafficConditionSubCause	*TrafficConditionSubCauseEnum*	optional	
accidentSubCause	*AccidentSubCauseEnum*	optional	
roadworksSubCause	*RoadworksSubCauseEnum*	optional	

Field		Type	Label	Description
	adverseWeatherCondition_AdhesionSubCause	*AdverseWeatherCondition_AdhesionSubCauseEnum*	optional	
	hazardousLocation_SurfaceConditionSubCause	*HazardousLocation_SurfaceConditionSubCauseCode*	optional	
	hazardousLocation_ObstacleOnTheRoadSubCause	*HazardousLocation_ObstacleOnTheRoadSubCauseCode*	optional	
	hazardousLocation_AnimalOnTheRoadSubCause	*HazardousLocation_AnimalOnTheRoadSubCauseCode*	optional	
	humanPresenceOnTheRoadSubCause	*HumanPresenceOnTheRoadSubCauseCode*	optional	
	wrongWayDrivingSubCause	*WrongWayDrivingSubCauseCode*	optional	
	rescueAndRecoveryWorkInProgressSubCause	*RescueAndRecoveryWorkInProgressSubCauseCode*	optional	
	adverseWeatherCondition_ExtremeWeatherConditionSubCause	*AdverseWeatherCondition_ExtremeWeatherConditionSubCauseCode*	optional	
	adverseWeatherCondition_VisibilitySubCause	*AdverseWeatherCondition_VisibilitySubCauseCode*	optional	
	adverseWeatherCondition_PrecipitationSubCause	*AdverseWeatherCondition_PrecipitationSubCauseCode*	optional	
	slowVehicleSubCause	*SlowVehicleSubCauseCode*	optional	
	dangerousEndOfQueueSubCause	*DangerousEndOfQueueSubCauseCode*	optional	

Field		Type	Label	Description
	vehicleBreakdownSubCause	*VehicleBreakdownSubCauseCode*	optional	
	postCrashSubCause	*PostCrashSubCauseCode*	optional	
	humanProblemSubCause	*HumanProblemSubCauseCode*	optional	
	stationaryVehicleSubCause	*StationaryVehicleSubCauseCode*	optional	
	emergencyVehicleApproachingSubCause	*EmergencyVehicleApproachingSubCauseCode*	optional	
	hazardousLocation_DangerousCurveSubCause	*HazardousLocation_DangerousCurveSubCauseCode*	optional	
	collisionRiskSubCause	*CollisionRiskSubCauseCode*	optional	
	signalViolationSubCauseCodeSubCause	*SignalViolationSubCauseCode*	optional	
	dangerousSituationSubCause	*DangerousSituationSubCauseCode*	optional	

TireSlippageEvent

Message Summary

message **TireSlippageEvent**

DEPRECATED: Handled by ElectronicStabilityControlEvent [deprecated = true]

Include: `sdii.v3.3.1.proto`

TrafficSignalHeadRecognition

Message Summary

message **TrafficSignalHeadRecognition**

The container provided with information regarding the detection of a traffic signal head and its lights / colors.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*double*	required	Timestamp of the message
`positionOffset`	*PositionOffset*	optional	Describes the relative position of the traffic signal in respect to the vehicle and the vehicle reference axis. Datatype changed from Vector to PositionOffset through v3.3.0 v2 fix
`trafficSignalDesignType`	*TrafficSignalHeadDesignTypeEnum*	optional	The main design type if identifyable.
`trafficSignalHeadOrientationType`	*TrafficSignalHeadOrientationTypeEnum*	optional	The orientation of the traffic signal head.
`trafficSignalLightColorBitfield`	*int32*	optional	A bitfield that references the colorization and the state of the individual traffic lights. The bitfield is grouped by 5bits per light starting with the most upper or left light. Every 5 bit pair is divided in 2 bits for the state and 3 bits for the color. State-Bits (0,1): 0=off, 1=on, 2 = blinking Color-Bits (2,3,4): 0=unknown; 1=red; 2=yellow, 3=green;4=white;5=reserved;6=reserved; Example: 001.00 010.00 011.01 1=r off 3=g on 2=y off Example: 001.00 010.10 011.00 1=r off 3=g off 2=y blinking
`laneReferenceID`	*int32*	optional	If lane recognition is activated and the reference between traffic signal and lane is known, the reference to a reference ID can be provided. The reference ID is a local ID within a message and not persisted over multiple messages.
`objectReferenceID`	*int32*	optional	If the Traffic Signal is provided with an object Type (e.g. a pole), then an object Reference can be used to identify a traffic Signal and that object to be correlated. The object ID is local within a message and not persisted over multiple messages.
`extensionContainer`	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

Vector3D

Message Summary

message **Vector3D**

A 3D-vector containing double values. Vector3D is used for anything requiring a three dimensional vector with optional accuracy, e.g. an acceleration-vector. According to the ENU (East-North-Up) System, a value is positive in right, up, front direction and negative in left, down, rear direction. When representing rotations then the value represents the rotation along the named axis in angular speed with the rotation in mathematical positive direction facing the positive end of the axis is represented by a positive value.

From the vehicles reference point, increasing any dimension of a rotation vector results in “yaw to the right” (heading), “roll to the left” (cross-slope) and “pitching to the front” (slope). A longitudinal value always refers to the rear-front-axis in longitudinal direction, a transversal value refers to the left-right axis and a vertical value refers to the down-up axis.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`longitudinalValue`	*double*	required	
`lateralValue`	*double*	required	
`verticalValue`	*double*	required	
`longitudinal Accuracy`	*double*	optional	
`lateralAccuary`	*double*	optional	
`verticalAccuracy`	*double*	optional	

VehicleDynamics

Message Summary

message **VehicleDynamics**

Vehicle Dynamics are measurements beyond the position of a vehicle. Typically vehicle dynamics information is measured by the vehicle using on board sensors at high frequency compared to positions (e.g. 5Hz or 10Hz). Depending of the set of sensors in the vehicle different values could be provided. In order to keep the complexity at a manageable level these raw measurements must be converted into meaningful values and hence are a result of calculations either in the vehicle or in the OEM or System Vendor backend.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*int64*	required	Timestamp of the message
`curvature_1pm`	*double*	optional	The curvature as measured (calculated) by the vehicle at the location of the vehicle. A positive value means a curve to the right in driving direction. **Unit:** 1/Meter [1/m] | **Range:** -1..1/m| **Resolution:** 0.001/m

Field	Type	Label	Description
slope_percent	*double*	optional	The slope as measured (calculated) by the vehicle at the location of the vehicle. A positive value means uphill slope in driving direction, a negative value means downhill slope in driving direction. **Unit:** Percent [%] | **Range:** -100..100% | **Resolution:** 0.1%
curvatureAccuracy_1pm	*double*	optional	The standard deviation for the curvature. **Unit:** 1/Meter [1/m] | **Range:** -1..1/m| **Resolution:** 0.0001/m
slopeAccuracy_percent	*double*	optional	The standard deviation for the curvature. **Unit:** 1/Meter [1/m] | **Range:** -1..1/m| **Resolution:** 0.0001/m
averageSuspensionTravel_mm	*int32*	optional	Over a duration of 1 seconds the accumulated travel of the suspensions (averaged over all 4 wheels) is provided in this attribute. **Unit:** Milimeter [mm] | **Range:** 0..10000mm| **Resolution:** 1mm
averageAccelerationVector_mps2	*Vector3D*	optional	TODO **Unit:** Meter per Square Second [mps2] in 3D| **Range:** -100..100mps2 in 3D | **Resolution:** 0.5mps2 in 3D
averageRotationRateVector_omega	*Vector3D*	optional	Over a timerange of 1 second the average rotation rate vector is provided in this complex data type as a yaw,pitch,roll rotation rate. A value of 2*PI ~ 6.28 equals a rotation rate of one full rotation per second. The direction of positive values is describted in the object Vector3D. **Unit:** Angular Velocity [rad/s] | **Range:** -314..314rad/s| **Resolution:** 0.1rad/s
averageSuspensionTravelAccuracy_mm	*int32*	optional	the accuracy of the provided suspension travel value **Unit:** Milimeter [mm] | **Range:** 0..10000mm | **Resolution:** 1mm
bankAngle_percent	*double*	optional	The bank angle is the angle at which the vehicle is inclined about its longitudinal axis with respect to the horizontal. A negative value describes an inclination to the left, a positive value describes an inclination to the right. **Unit:** Degree [°] | **Range:** -200°..200°| **Resolution:** 0.1°
steeringWheelAngle_deg	*int32*	optional	The steering wheel angle. Where 0 is the angle of the steering wheel where the vehicle is driving straight. A positive value describes a rotated steering wheel to the left, a negative value describes a rotated steering wheel to the right. **Unit:** Degree [°] | **Range:** -1080..1080°| **Resolution:** 1°
steeringWheelAngleAccuracy_deg	*int32*	optional	Standard deviation of the steering wheel angle accuracy. **Unit:** Degree [°] | **Range:** -180..180°| **Resolution:** 1°

Field	Type	Label	Description
steeringWheelAngleChange_degPerSec	*int32*	optional	The rate of the steering wheel angle change in unit of degree per second. A positive value describes in a steering wheel change to the left (increase of steering wheel angle), a negative value describes the steering wheel change to the right (decrease of steering wheel angle). **Unit:** Degree per Second [°/sec] | **Range:** -360..360°/sec| **Resolution:** 1°/sec
steeringWheelAngleChangeAccuracy_degPerSec	*int32*	optional	Standard deviation of the steering wheel angle change accuracy in unit of degree per second. **Unit:** Degree per Second [°/sec] | **Range:** -180..180°/sec| **Resolution:** 1°/sec
accelerationPedalPosition_percent	*int32*	optional	Acceleration pedal position – percentage in relate to the original position. The value 0 describes no pedal pressing where the value 100% describes a full pressed down pedal position. **Unit:** Percent [%] | **Range:** 0..100%| **Resolution:** 1%
brakePedalPosition_percent	*int32*	optional	Bake pedal position – percentage in relate to the original position. The value 0 describes no pedal pressing where the value 100% describes a full pressed down pedal position. **Unit:** Percent [%] | **Range:** 0..100%| **Resolution:** 1%
engineSpeed_rpm	*int32*	optional	Engine speed in revolutions per minute. **Unit:** Revolutions per Minute [rpm] | **Range:** 0..MAX_INT | **Resolution:** 1rpm
engineTorque_nm	*int32*	optional	Engine torque. **Unit:** Newtonmeter [nm] | **Range:** 0..MAX_INT| **Resolution:** 1nm
brakeTorque_nm	*int32*	optional	Brake Torque **Unit:** Newtonmeter [nm] | **Range:** 0..MAX_INT| **Resolution:** 1nm
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

VehicleManeuverEvent

Message Summary

message **VehicleManeuverEvent**

Sensor Data providing information on one vehicle maneuver event derived by a not defined set of in vehicle sensor data.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timestampUTC_ms`	*int64*	required	Timestamp of the message
`vehicleManeuverType`	*VehicleManeuverTypeEvent*	optional	The type of provided vehicle maneuver
`extensionContainer`	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

VehicleMetaData

Message Summary

message **VehicleMetaData**

The vehicle meta data provides information about the vehicle that is valid for the entire path. This includes vehicle type information, the vehicle reference point. All absolute positions (longitude / latitude) that are reported to the Sensor Data Ingestion Interface are expected to be at the center of the vehicle. All offsets that are reported are expected to be offsets from this center point of the vehicle. Altitude that is reported to the Interface are expected to be altitude on the ground (not the altitude of the location of the GPS antenna). Instead of providing the altitude on the ground it is possible to report a different altitude with a constant offset. This offset from the ground must be provided through the vehicle meta data.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`vehicleTypeGeneric`	*VehicleTypeGenericEnum*	optional	The generic vehicle type is an enumeration used to provide information what type of vehicle the sensor data produced.
`vehicleSpecificMetaData`	*KeyValuePairString*	repeated	Some of the data submitted may be very OEM or vehicle specific. For this purpose a generic mechanism of string key value pairs is included that allows submission of such generic information. Based on information provided through such key value pairs in conjunction with individual OEM agreements different processing assumptions may be made.

Field	Type	Label	Description
vehicleReferencePointDeltaAboveGround_m	*double*	optional	The vehicle reference point delta above ground provides the difference between the altitude values and the street level. Altitude values should always be provided at street level and hence this value here should be zero. However in case the altitude is provided at a different level (e.g. at the roof of the vehicle) then this delta value shall provide the distance between the altitude points and the street level. This attribute will be deprecated in a future version and replaced by the addition of a 3D Vector for Reference offset (**GNSSPositionReferencePointOffset**, and **RelativePositionReferencePointOffset**) **Unit:** Meter [m] | **Range:** 0..20m | **Resolution:** 0.01m
curvatureAccuracy_1pm	*double*	optional	**Deprecated** The standard deviation for the curvature. Curvature Accuracy is deprecated in Vehicle Meta Data and will be provided with the Curvature Value within Vehicle Dynamics. **Unit:** 1/meter [1/m] | **Range:** -1..1/m | **Resolution:** 0.00001/m
slopeAccuracy_percent	*double*	optional	**Deprecated** The standard deviation for the slope. Slope Accuracy is deprecated in Vehicle Meta Data and will be provided with the Slope Value within Vehicle Dynamics. **Unit:** Percent [%] | **Range:** -100..100% | **Resolution:** 0.01%
vehicleLength_m	*double*	optional	The length of the vehicle from most front part to the most rear part (including e.g. bumpers and spoilers). **Unit:** Meter [m] | **Range:** 0..100m | **Resolution:** 0.001m
vehicleWidth_m	*double*	optional	The registered width of the vehicle from the most left part to the most right part (including e.g. mirrors). **Unit:** Meter [m] | **Range:** 0..100m | **Resolution:** 0.001m
vehicleHeight_m	*double*	optional	The total height of the vehicle from lowest part (surface) to the highest part (e.g. antenna). **Unit:** Meter [m] | **Range:** 0..100 | **Resolution:** 0.001m
primaryFuelTankVolume	*double*	optional	Fuel tank quantity of the primary tank. The unit is arbitrary depending on the Fuel Type described in attribute primaryFuelType. **Unit:** arbitrary upon Fuel Type | **Range:** 0..1000 | **Resolution:** 0.01
primaryFuelType	*FuelTypeEnum*	optional	The type of the primary fuel source. Fuel tank quantity of the secondary tank. The unit is arbitrary depending on the Fuel Type described in attribute secondaryFuelType. **Unit:** arbitrary upon Fuel Type | **Range:** 0..1000 | **Resolution:** 0.01
secondaryFuelTankVolume	*double*	optional	
secondaryFuelType	*FuelTypeEnum*	optional	The type of the secondary fuel source.
vehicleHeightDetail	*VehicleHeightDetail*	repeated	The list of zero or more hight values of the vehicle

Field	Type	Label	Description
GNSSPosition ReferencePoint Offset	*Vector3D*	optional	The position of the GNSS reference point (latitude, longitude, altitude) as a relative offset of the vehicles reference point. The vehicle reference point may vary or be defined in a future version. The sum of both vectors GNSSPositionReferencePointOffset and RelativePositionReferencePointOffset describe the offset betwenn the absolute position coordinates and the local sensor coordinates in local vehicle orientation. **Unit:** Milimeters [mm] in 3D | **Range:** -5000..5000mm in 3D | **Resolution:** 1mm
RelativePosition ReferencePoint Offset	*Vector3D*	optional	The position of the relative offset reference point from which relative distances are returned within the SDII messages. The position is localized from the vehicle reference point. The vehicle reference point may vary or be defined in a future version. The sum of both vectors GNSSPositionReferencePointOffset and RelativePositionReferencePointOffset describe the offset betwenn the absolute position coordinates and the local sensor coordinates in local vehicle orientation. **Unit:** Milimeters [mm] in 3D | **Range:** -5000..5000mm in 3D | **Resolution:** 1mm

VehicleMetaData.VehicleHeightDetail

Message Summary

message **VehicleMetaData.VehicleHeightDetail**

In Case a more complex height information shall be provided than just the hight in meters, Vehicel Height Details hold the hight plus additionally the reference points of the hight measurement.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
longitudinalVehicle HeightReference Point	*LongitudinalVehicle HeightReferencePoint Enum*	optional	A designation of the location (front, middle, rear) of the provided height value.
lateralVehicle HeightReference Point	*LateralVehicleHeight ReferencePointEnum*	optional	A designation of the location (left, middle, right) of the provided height value.
vehicleHeightValue _mm	*int64*	optional	The height of the vehicle at the designated position from road surface to the highest point. **Unit:** Milimeter [mm] | **Range:** 0..5000mm | **Resolution:** 1mm

VehicleStatus

Message Summary

message **VehicleStatus**

Contains the description of multiple sensor based attributes describing the vehicle status at one timestamp.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`timeStampUTC_ms`	*int64*	required	Timestamp of the message
`transmissionMode`	*TransmissionModeEnum*	optional	The actual transmission mode of the vehicle
`lightStateBitfield`	*int64*	optional	The light state is a bitfield that encodes which lights of the vehicle are turned on and off. For the encoding following bitmask description is used: LOWBEAMS = 0x0001; /// bit 0 HIGHBEAMS = 0x0002; /// bit 1 FOGLAMP_FRONT = 0x0004; /// bit 2 FOGLAMP_REAR = 0x0008; /// bit 3 HAZARD = 0x0010; /// bit 4 LEFT_TURN = 0x0020; /// bit 5 RIGHT_TURN = 0x0040; /// bit 6 Bitwise OR of LightStateBitfield values
`wiperState`	*WiperStateEnum*	optional	
`temperatureExternal_cel`	*double*	optional	
`wiperSpeed_wpm`	*int32*	optional	
`driveWheelReference`	*int64*	optional	
`chassisClearance_m`	*double*	optional	
`mileage_km`	*double*	optional	
`primaryFuelState`	*double*	optional	
`primaryFuelStateAccuracy`	*double*	optional	
`estimatedPrimaryRange_km`	*double*	optional	
`secondaryFuelState`	*double*	optional	
`secondaryFuelStateAccuracy`	*double*	optional	
`estimatedSecondaryRange_km`	*double*	optional	

Field	Type	Label	Description
maintenanceLightState	*MaintenanceLightStateBitfield*	optional	The maintenance light state is a bitfield that encodes which maintenance lights or warnings are active – not during control cycle upon ignition startup. the following bits are used: WARNING_ENGINE_CONTROL = 0x0001; WARNING_OIL_PRESSURE = 0x0002; WARNING_COOLANT_TEMP = 0x0004; WARNING_VEHICLE_SERVICE = 0x0008; WARNING_BATTERY_CHARGING = 0x0010; WARNING_TIRE_PRESSURE = 0x0020; WARNING_LAMP_OUT = 0x0040; WARNING_OTHER_HIGH_PRIO = 0x0080; WARNING_OTHER_LOW_PRIO = 0x0100;
currentGear	*uint32*	optional	
ignitionState	*IgnitionStateEnum*	optional	
engineState	*EngineStateEnum*	optional	
doorStates	*DoorStateMessage*	repeated	List of zero or more messages each describing a door state
ventilationState	*VentilationStateMessage*	optional	This section contains the description of all available information about the air ventilation system.
extensionContainer	*ExtensionContainer*	repeated	Contains the description and byte value of an dynamic extension content, that is undefined in this Specification.

VehicleStatus.DoorStateMessage

Message Summary

message **VehicleStatus.DoorStateMessage**

Contains information on the state of one door.

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
door	*DoorIdentificationEnum*	required	
state	*DoorStateEnum*	optional	
lock	*DoorLockStateEnum*	optional	

VehicleStatus.VentilationStateMessage

Message Summary

message **VehicleStatus.VentilationStateMessage**

Contains information on the state of the in vehicle ventilation

Include: `sdii.v3.3.1.proto`

Properties

Field	Type	Label	Description
`mainVentilationState`	*VentilationStateEnum*	optional	The state of the ventilation state
`mainVentilationSetting`	*VentilationSettingEnum*	optional	The settings of the AC and Ventilation
`circulationSetting`	*CirculationSettingEnum*	optional	The settings of the AC and Ventilation

Envelope.MapStandardEnum

TODO

Enumeration Details

- 0 : `MAP_STANDARD_UNDEFINED`
- 1 : `OTHER`
- 2 : `NDS` The format of the map installed in the vehicle is based on NDS standard.
- 3 : `GDF` The format of the map installed in the vehicle is based on GDF standard.
- 4 : `RDF` The format of the map installed in the vehicle is based on RDF standard.
- 5 : `HERE_HDMAP` The map installed in the vehicle is HERE_HDMAP.

EnvironmentStatus.LightConditionsEnum

Describes possible light conditions around the vehicle

Enumeration Details

- 0 : `LIGHT_CONDITIONS_UNDEFINED` No Lightconditions provided
- 1 : `GENERAL_DAYLIGHT` Describing light environment
- 2 : `GENERAL_DARK` Describing dark environment (usually lights are turned on)
- 3 : `DAYLIGHT_SUN` Describing daylight with sun shining conditions
- 4 : `DAYLIGHT_CLOUD` Describing daylight with diffuse light conditions
- 5 : `DUSK_OR_DAWN` Describing a halfway light and dark situation with intermediate visibility
- 6 : `NIGHT_LUMINATED` Describing a night situation with illumination (tunnel or urban areas)
- 7 : `NIGHT_DARK` Describing a night situation with no illumination of the road and surrounding

EnvironmentStatus.PrecipitationEnum

Describes possible types of precipitation

Enumeration Details

- 0 : `PRECIPITATION_TYPE_UNDEFINED` No Precipitation Type provided
- 1 : `NONE` No precipitation detected
- 2 : `RAIN` Refers to rain
- 3 : `HAIL` Refers to hail
- 4 : `SNOW` Refers to snow

EnvironmentStatus.RoadSurfaceTypeEnum

Describes possible types of road surface

Enumeration Details

- 0 : `ROAD_SURFACE_TYPE_UNDEFINED`
- 1 : `ASPHALT`
- 2 : `CONCRETE`
- 3 : `PAVED`
- 4 : `GRAVEL`
- 5 : `ICEORSNOW`
- 6 : `UNKNOWN`

LaneBoundaryRecognition.LaneBoundaryColorEnum

Describes the possible recognizable colors of lane marking.

Enumeration Details

- 0 : `LANE_BOUNDARY_COLOR_UNDEFINED` Undefined Lane marking color
- 1 : `WHITE` White lane marking color
- 2 : `YELLOW` Yellow lane marking color
- 3 : `BLUE` Blue lane marking color
- 4 : `RED` Red lane marking color Additional colors to be defined later

LaneBoundaryRecognition.LaneBoundaryEgoLaneReferenceEnum

Enumeration Details

- 0 : `LANE_BOUNDARY_EGO_LANE_REFERENCE_UNDEFINED` No Description Type provided
- 1 : `LANE_BOUNDARY_EGO_LANE_REFERENCE_TRUE` The Lane marking belongs to the EGO-Lane
- 2 : `LANE_BOUNDARY_EGO_LANE_REFERENCE_TRUE_LEFT` The lane marking is the left marking of the ego lane
- 3 : `LANE_BOUNDARY_EGO_LANE_REFERENCE_TRUE_RIGHT` The lane marking is the right marking of the ego lane
- 4 : `LANE_BOUNDARY_EGO_LANE_REFERENCE_FALSE` The lane marking does not belong to the ego lane.

LaneBoundaryRecognition.LaneBoundaryPositionReferenceEnum

Enumeration Details

- 0 : `LANE_BOUNDARY_POSITION_REFERENCE_UNDEFINED` No Recognition Type provided
- 1 : `LANE_BOUNDARY_POSITION_REFERENCE_INNER` The position is measured from the inne (near) end of lane boundary marking
- 2 : `LANE_BOUNDARY_POSITION_REFERENCE_MIDDLE` The position is measured from the middle of the lane boundary marking
- 3 : `LANE_BOUNDARY_POSITION_REFERENCE_OUTER` The position is measured from the outer (far) end of the boundary marking

LaneBoundaryRecognition.LaneBoundaryRecognitionChangeEnum

Describing possible recognition changes

Enumeration Details

- 0 : `LANE_BOUNDARY_RECOGNITION_CHANGE_UNDEFINED` No Recognition Change provided
- 1 : `LANE_BOUNDARY_START` The start of the lane was detected
- 2 : `LANE_BOUNDARY_END` The end of the lane was detected

LaneBoundaryRecognition.LaneBoundaryRecognitionTypeEnum

Describes possible recognition types of a lane boundary

Enumeration Details

- 0 : `LANE_BOUNDARY_RECOGNITION_TYPE_UNDEFINED` No Recognition Type provided
- 1 : `LANE_DETECTED` A lane was detected
- 2 : `LANE_NOT_DETECTED` No lane was detected

LaneBoundaryRecognition.LaneBoundaryTypeEnum

Describes possible Lane Boundary types

Enumeration Details

- 0 : `LANE_BOUNDARY_TYPE_UNDEFINED` No Lane Boundary Type defined
- 1 : `SINGLE_SOLID_PAINT` A single solid painted line
- 2 : `DOUBLE_SOLID_PAINT` A double solid painted line
- 3 : `LONG_DASHED_PAINT` A line consisting of painted long dashes
- 4 : `SHORT_DASHED_PAINT` A line consisting of painted short dashes
- 5 : `SHADED_AREA_PAINT` The Lane Boudnary is a shared area of multiple diagonal painted lines
- 6 : `DASHED_BLOCKS` The line consists of physical blocks fixed on the road
- 7 : `DOUBLE_LINE_DASHED_SOLID_PAINT` The line is a double line where dashed (left) and solid (right) line markings are combined. Left and Right are related to driving direction.
- 8 : `DOUBLE_LINE_SOLID_DASHED_PAINT` The line is a double line where solid (left) and dashed (right) line markings are combined. Left and Right are related to driving direction.
- 9 : `PHYSICAL_DIVIDER` The line marking is a physical divider.
- 10 : `DOUBLE_DASHED_LINES` The line is a double dashed line.

LocalizationInformation.PositioningSystem.GNSSFixEnum

Describes possible states of position fix.

Enumeration Details

- 0 : `GNSS_FIX_UNDEFINED`
- 1 : `GNSS_FIX_NO`
- 2 : `GNSS_FIX_2D`
- 3 : `GNSS_FIX_3D`

LocalizationInformation.PositioningSystem.PositioningSystemTypeEnum

Describes possible systems for positioning.

Enumeration Details

- 0 : `POSITIONING_SYSTEM_TYPE_UNDEFINED`
- 1 : `POSITIONING_SYSTEM_TYPE_OTHER`
- 2 : `POSITIONING_SYSTEM_TYPE_GPS`
- 3 : `POSITIONING_SYSTEM_TYPE_GLONASS`
- 4 : `POSITIONING_SYSTEM_TYPE_GALLILEO`
- 5 : `POSITIONING_SYSTEM_TYPE_IRNSS`
- 6 : `POSITIONING_SYSTEM_TYPE_BeiDou`

MediaContainer.MediaTypeEnum

Describes possible types of media that can be provided

Enumeration Details

- 0 : `MEDIA_TYPE_UNDEFINED`
- 1 : `OTHER`
- 2 : `IMAGE`
- 3 : `VIDEO`
- 4 : `AUDIO`

ObjectDetection.ObjectRecognitionMatchTypeEnum

Describes possible matchin types to a provided object

Enumeration Details

- 0 : `OBJECT_RECOGNITION_MATCH_TYPE_UNDEFINED` No object recognition match type provided
- 1 : `OBJECT_MATCHED` The Object was matching to expectations
- 2 : `OBJECT_NOT_MATCHED` The object was not matching to expectations

ObjectDetection.ObjectRecognitionTypeEnum

Describes possible types of recognition (type of the Object Recognition Message)

Enumeration Details

- 0 : `OBJECT_RECOGNITION_TYPE_UNDEFINED` No object recognition type provided
- 1 : `OBJECT_DETECTED` An object was detected
- 2 : `OBJECT_NOT_DETECTED` No object was detected where an object was expected. E.g. derived from local hazard feed content

ObjectDetection.ObjectSurfaceMaterialTypeEnum

Describing possible surface material of an object

Enumeration Details

- 0 : `OBJECT_SURFACE_MATERIAL_UNDEFINED` No material type provided
- 1 : `UNKNOWN_MATERIAL` No material type identifyable
- 2 : `STONE`
- 3 : `ASPHALT`
- 4 : `ORGANIC` e.g. Grass
- 5 : `METALLIC`
- 6 : `PLASTIC`
- 7 : `WOOD`

ObjectDetection.ObjectSurfaceTypeEnum

Describing possible types of object surface

Enumeration Details

- 0 : `OBJECT_SURFACE_UNDEFINED`
- 1 : `UNKNOWN_SURFACE`
- 2 : `FLAT`
- 3 : `ROUGH`

ObjectDetection.ObjectTypeEnum

Describing possibly types of recognized object

Enumeration Details

- 0 : `OBJECT_TYPE_UNDEFINED` An undefined type
- 1 : `MOVING_GENERAL` A general movable object that is not further specified
- 2 : `STATIC_GENERAL` A general static object that is not further specified
- 3 : `STATIC_GENERAL_VERTICAL` A static object, with a prevailing vertical extent
- 4 : `STATIC_GENERAL_TRANSVERSAL` A static object, with a prevailing transversal extent

- 5 : STATIC_GENERAL_LATERAL A static object with a prevailing lateral extent
- 6 : MOVING_VEHICLE A movable (standing or moving) object identified as vehicle
- 7 : MOVING_TRUCK A movable (standing or moving) object identified as Truck
- 8 : MOVING_BIKE A movable object identified as bike (bicycle or motorbike)
- 9 : MOVING_PERSON A movable object identified as person
- 10 : STATIC_BRIDGE A static object identified as bridge over the road
- 11 : STATIC_TUNNEL A static object identified as the entrance of a tunnel, or the walls of a tunnel
- 12 : STATIC_POLE A pole
- 13 : STATIC_BAR A bar, crossing the road
- 14 : STATIC_TREE A tree at the side of a road
- 15 : STATIC_WALL A wall along the road
- 16 : STATIC_BOLLARD A bollard on the road
- 17 : STATIC_GUIDERAIL A guiderail along the raod
- 18 : STATIC_TRAFFICISLAND A trafficisland on the road
- 19 : STATIC_SIGN A traffic Sign object
- 20 : STATIC_TRAFFIC_LIGHT A traffic Light object

PassengerEnvironment.MediaSourceEnum

Describes possible types of Media Sources.

Enumeration Details

- 0 : MEDIA_SOURCE_UNDEFINED
- 1 : MEDIA_SOURCE_OFF
- 2 : MEDIA_SOURCE_RADIO
- 3 : MEDIA_SOURCE_EXTERNAL_SOURCE

PassengerEnvironment.PassengerSeatUsage.PassengerOccupancyTypeEnum

Describes possible types of occupancies for one passenger seat

Enumeration Details

- 0 : PASSENGER_OCCUPANCY_TYPE_UNDEFINED
- 1 : PASSENGER_OCCUPANCY_LIGHT_WEIGHT The seat is occupied by a light weight (e.g. child)
- 2 : PASSENGER_OCCUPANCY_HEAVY_WEIGHT The seat is occupied by a common weight (e.g. adult)
- 3 : PASSENGER_OCCUPANCY_CHILD_SAEFTY_SEAT The seat is equipped with a child seat
- 4 : PASSENGER_OCCUPANCY_BABY_CARRIER The seat is equipped with a baby carrier

PassengerEnvironment.PassengerSeatUsage.PassengerSeatColumnSimpleEnum

Describes possible seat columns description in simple enumeration.

Enumeration Details

- 0 : PASSENGER_SEAT_COLUMN_UNDEFINED
- 1 : PASSENGER_SEAT_COLUMN_LEFT
- 2 : PASSENGER_SEAT_COLUMN_MIDDLE
- 3 : PASSENGER_SEAT_COLUMN_RIGHT

PassengerEnvironment.PassengerSeatUsage.PassengerSeatRowSimpleEnum

Describes possible seat row description in simple enumeration.

Enumeration Details

- 0 : PASSENGER_SEAT_ROW_UNDEFINED
- 1 : PASSENGER_SEAT_ROW_FRONT
- 2 : PASSENGER_SEAT_ROW_MIDDLE
- 3 : PASSENGER_SEAT_ROW_REAR

PassengerEnvironment.PassengerSeatUsage.PassengerSeatUsageEnum

Describes possible types of usage for one passenger seat

Enumeration Details

- 0 : PASSENGER_SEAT_USAGE_UNDEFINED
- 1 : PASSENGER_SEAT_USAGE_FREE
- 2 : PASSENGER_SEAT_USAGE_OCCUPIED

PassengerEnvironment.RoutingSelectionPathTypeEnum

Describes possible types of routing

Enumeration Details

- 0 : ROUTING_SELECTED_PATH_TYPE_UNDEFINED
- 1 : ROUTING_SELECTED_PATH_TYPE_SHORTEST
- 2 : ROUTING_SELECTED_PATH_TYPE_FASTEST
- 3 : ROUTING_SELECTED_PATH_TYPE_ECONOMIC

PositionEstimate.HeadingDetectionEnum

Describes different technologies for heading detection.

Enumeration Details

- 0 : HEADING_DETECTION_UNDEFINED No speed detection defined
- 1 : HEADING_RAW_GPS Accuracy derived from two consecutive location detection.
- 2 : HEADING_MAGNETIC_SENSOR Heading fusioned by GPS additionally with magnetic sensors.
- 3 : HEADING_MULTI_SENSOR_FUSION Heading fusioned by multiple inputs including driving distance, steering angle, etc.
- 4 : HEADING_BY_MAP The heading is derived from the map link on which a vehicle was mapped.

PositionEstimate.LaneCountDirectionEnum

describing the possible directions of lane counting.

Enumeration Details

- 0 : LANE_COUNT_DIRECTION_UNDEFINED No lane count direction defined. Default to be used
- 1 : LANE_COUNT_DIRECTION_FROM_LEFT Counted from the left
- 2 : LANE_COUNT_DIRECTION_FROM_RIGHT Counted from the right

PositionEstimate.PositionTypeEnum

The position type is an enumeration used to differentiate between different types of absolute positions.

Enumeration Details

- 0 : POSITION_TYPE_UNDEFINED Description:
- 1 : RAW_GPS Raw GPS position without usage of other sensors and no map matching applied.
- 2 : FILTERED GPS data filtered or fused with inertial data but not map matched
- 3 : MAP_MATCHED_REGULAR_MAP Position based on sensors such as GPS and inertial fused and matched to a normal infotainment map database
- 4 : MAP_MATCHED_HD_MAP Position based on sensors such as GPS and inertial fused and matched to a the HERE HD Map on centerline level
- 5 : MAP_MATCHED_HD_MAP_LANE Position based on sensors such as GPS and inertial fused and matched to a the HERE HD Map on lane level

PositionEstimate.SpeedDetectionEnum

Describes different technologies for speed detection.

Enumeration Details

- 0 : `SPEED_DETECTION_UNDEFINED` The type of speed detection technology used.
- 1 : `SPEED_RAW_GPS` Accuracy derived from two consecutive location detection.
- 2 : `SPEED_WHEEL_TICKS` Speed derived by the rotation speed of the vehicles wheels.
- 3 : `SPEED_RADAR_SONAR` Speed derived by detection of returned signals from high frequency sensors.
- 4 : `SPEED_FILTERED` Speed derived by filtered information from a multitude of sensors.

PositionOffset.LateralOffsetSimpleEnum

Describing the lateral offset on simple directions (left, middle, right)

Enumeration Details

- 0 : `LATERAL_OFFSET_SIMPLE_UNDEFINED` No offset is provided
- 1 : `LATERAL_OFFSET_SIMPLE_LEFT` Towards the left
- 2 : `LATERAL_OFFSET_SIMPLE_MIDDLE` Approximately in the middle
- 3 : `LATERAL_OFFSET_SIMPLE_RIGHT` Towards the right

PositionOffset.LongitudinalOffsetSimpleEnum

The simple lateral offset is an enumeration to indicate positions compared to the vehicle reference point.

Enumeration Details

- 0 : `LONGITUDINAL_OFFSET_SIMPLE_UNDEFINED` No offset is provided
- 1 : `LONGITUDINAL_OFFSET_SIMPLE_FRONT` In front of the vehicle
- 2 : `LONGITUDINAL_OFFSET_SIMPLE_CENTER` Approximately at the vehicle
- 3 : `LONGITUDINAL_OFFSET_SIMPLE_BACK` Behind the vehicle

PositionOffset.VerticalOffsetSimpleEnum

Enum to describe the vertical offset in simple categories (above, at level, below)

Enumeration Details

- 0 : `VERTICAL_OFFSET_SIMPLE_UNDEFINED` No offset is provided

- 1 : VERTICAL_OFFSET_SIMPLE_ABOVE Above the vehicle
- 2 : VERTICAL_OFFSET_SIMPLE_AT_LEVEL Approximately at the level of the vehicle
- 3 : VERTICAL_OFFSET_SIMPLE_BELOW Below the vehicle

RoadAttributeRecognition.ReferenceTypeEnum

Describes possible types of reference of the attribute to the road.

Enumeration Details

- 0 : REFERENCE_TYPE_UNDEFINED
- 1 : SINGLE_RECOGNITION The attribute has been identified at the current position (referenced by timestamp)
- 2 : STARTED_BEFORE The attribute has been identified as started a given distance before this point
- 3 : ENDED_BEFORE The attribute has been identified as ended a given distance before this point

RoadAttributeRecognition.RoadAttributeTypeEnum

Describes possible types of road attributes that can be described in the object

Enumeration Details

- 0 : ROAD_ATTRIBUTE_UNDEFINED
- 1 : ROAD_ATTRIBUTE_MOTORWAY The current road is a motorway
- 2 : ROAD_ATTRIBUTE_ONE_WAY_TRAFFIC The current road is a one way road
- 3 : ROAD_ATTRIBUTE_TWO_WAY_TRAFFIC The current road is a two way raod
- 20 : ROAD_ATTRIBUTE_TUNNEL The current road segment is within a tunnel
- 21 : ROAD_ATTRIBUTE_BRIDGE The current road segment is a bridge
- 22 : ROAD_ATTRIBUTE_ARTIFICIAL_ILLUMINATION The current road is artificial illuminated
- 23 : ROAD_ATTRIBUTE_ROAD_WORKS The vehicle is driving within a construction area
- 24 : ROAD_ATTRIBUTE_ROAD_WORKS_NARROW_LANES Similar to 23. Additionally the lanes are narrowed or shifted
- 50 : ROAD_ATTRIBUTE_LANE_START_RIGHT_SIDE One additional lane start has been detected on the right side of the road
- 51 : ROAD_ATTRIBUTE_LANE_START_LEFT_SIDE One additional lane start has been detected on the left side of the road
- 52 : ROAD_ATTRIBUTE_LANE_END_RIGHT_SIDE The end of one lane has been detected on the right side of the road
- 53 : ROAD_ATTRIBUTE_LANE_END_LEFT_SIDE The end of one lane has been detected on the left side of the road
- 54 : ROAD_ATTRIBUTE_LANE_SPLIT_MIDDLE A lane in the middle of the road has been split (not a right side or left side lane)
- 55 : ROAD_ATTRIBUTE_LANE_MERGE_MIDDLE Two lanes have merged together with no identifyable ending and prevailing lane.
- 56 : ROAD_ATTRIBUTE_CROSS_WALK The vehicle has passed a cross walk. (no specification of crossing regulation)

RoadBoundaryRecognition.RoadBoundaryChangeTypeEnum

Describes possible types of road boundary change of the road boundary objectCount

Enumeration Details

- 0 : ROAD_BOUNDARY_CHANGE_UNDEFINED
- 1 : ROAD_BOUNDARY_START A “Start”-Flag is giving information that the provided boundary is starting at the proximity of the vehicle (changing from not being there to being there)
- 2 : ROAD_BOUNDARY_END A “end”-Flag is giving information that the provided boundary is ending at the proximity of the vehicle (changing from being there to not being there)

RoadBoundaryRecognition.RoadBoundaryRecognitionTypeEnum

Describes possible types of recognition for Road Boundary Objects

Enumeration Details

- 0 : ROAD_BOUNDARY_UNDEFINED
- 1 : ROAD_BOUNDARY_DETECTED A road boundary has been detected
- 2 : ROAD_BOUNDARY_NOT_DETECTED Derived road boundary information (e.g. map) has not been detected

RoadBoundaryRecognition.RoadBoundaryTypeEnum

Describes possible types of road boundary.

Enumeration Details

- 0 : ROAD_BOUNDARY_TYPE_UNDEFINED No Road Boundary Type provided
- 1 : GENERAL_UNPASSABLE The boundary is unpassable without any additional specified information
- 2 : GENERAL_PASSABLE The boundary is passable. A vehicle could drive over it without any damage on the vehicle
- 3 : PASSABLE_FREESPACE
- 4 : PASSABLE_GREENFIELD A greenfield begins behind the road boundary
- 5 : PASSABLE_PAVEMENT Drivable pavement is connected to the road boundary
- 6 : UNPASSABLE_WALL The road side ends with a wall
- 7 : UNPASSABLE_GUARDRAIL The road side end with a guardrail
- 8 : UNPASSABLE_DIVIDER The road side ends with a physical divider
- 9 : UNPASSABLE_CURBSTONE The road side ends with a curbstone (with or without a sidewalk)

- 10 : UNPASSABLE_PARKINGVEHICLE Similar to free parking space, at the road side parking vehicles are detected
- 11 : UNPASSABLE_PARKINGVEHICLE_DIAGONAL Similar to free parking space, at the road side parking vehicles are detected parking in diagonal orientation
- 12 : PASSABLE_FREEPARKINGSPACE A free parking space is directly at the road boundry

RoadCondition.RoadRoughnessLateralPositionEnum

Describes possible lateral positions of occurence for a road roughness description

Enumeration Details

- 0 : bothSidesOfVehicle Both sides of the vehicle
- 1 : leftSideOfVehicle Left side of the vehicle
- 2 : rightSideOfVehicle Right side of the vehicle

RoadMarkingRecognition.RoadMarkingTypeEnum

Describes possible types of road marking

Enumeration Details

- 0 : ROAD_MARKING_TYPE_UNDEFINED Not recognized Road Marking Type)
- 1 : ROAD_MARKING_TYPE_ARROW_LEFT Indicates a left arrow on the road surface, describing a left only turn on the lane (German Sign Type: STVO297)
- 2 : ROAD_RKING_TYPE_ARROW_STRAIGHT_LEFT Indicating straight and left arrow on the road surface, describing a straight or left only turn on the lane(German Sign Type: STVO297)
- 3 : ROAD_MARKING_TYPE_ARROW_STRAIGHT Indicating straight arrow on the road surface, describing a straight direction driving (German Sign Type:STVO297)
- 4 : ROAD_MARKING_TYPE_ARROW_STRAIGHT_RIGHT Indicating straight and left arrow on the road surface, describing a straight or right only turn on the lane(German Sign Type:STVO297)
- 5 : ROAD_MARKING_TYPE_ARROW_RIGHT Indicates a right arrow on the road surface, describing a right only turn on the lane(German Sign Type:STVO297)
- 6 : ROAD_MARKING_TYPE_ARROW_UNDEFINED Indicating an identified road marking without specifying any direction – if the direction could not be identified.(German Sign Type:STVO297)

SignRecognition.RoadSignDependenciesEnum

Describes possible dependency regulation of the sign.

Enumeration Details

- 0 : ROAD_SIGN_DEPENDENCY_UNDEFINED
- 1 : RAIN
- 2 : SNOW

- 3 : TIME
- 4 : SEASON
- 5 : FOG
- 6 : SCHOOL
- 7 : TRUCKS
- 8 : TRAILER
- 9 : PASSENGER_CARS_WITH_TRAILER
- 10 : TRUCKS_WITH_TRAILER
- 11 : EXCEPT_TRACTOR
- 12 : BUS
- 100 : OTHER_DEPENDENCY_AS_TEXT

SignRecognition.RoadSignPermanencyEnum

Enumeration Details

- 0 : ROAD_SIGN_PERMANENCY_UNDEFINED No Permanency provided
- 1 : STATIC Sign is a fixed sign. Sign may also be a temporary sign set up on the road sign if not identifyable from fixed installations.
- 2 : VARIABLE Variable sign, for example an electronic sign, or a gantry with electronic signs
- 3 : VARIABLE_DEACTIVATED Variable sign that is deactivated and not displaying any value.

SignRecognition.RoadSignRecognitionShapeEnum

Describes possible shapes of a sign

Enumeration Details

- 0 : ROAD_SIGN_RECOGNITION_SHAPE_UNDEFINED No Shape provided
- 1 : RECTANGLE Based on best estimate including distortion of the viewing angle, the sign is assumed to be a rectangular shape
- 2 : SQUARE Based on best estimate including distortion of the viewing angle, the sign is assumed to be a square shape
- 3 : TRIANGLE_UP Based on best estimate including distortion of the viewing angle, the sign is assumed to be a triangle shape with a flat base and a spike top.
- 4 : TRIANGLE_DOWN Based on best estimate including distortion of the viewing angle, the sign is assumed to be a triangle shape with a flat top and a spike base
- 5 : DIAMOND Based on best estimate including distortion of the viewing angle, the sign is assumed to be a diamond shape (square rotated by 45°)
- 6 : HEXAGON Based on best estimate including distortion of the viewing angle, the sign is assumed to be a hexagon shape (STOP-sign)
- 7 : CIRCLE Based on best estimate including distortion of the viewing angle, the sign is assumed to be a circle shape

SignRecognition.RoadSignRecognitionTypeEnum

Describes the possible types of recognition (type of the Sign Recognition Message)

Enumeration Details

- 0 : ROAD_SIGN_RECOGNITION_TYPE_UNDEFINED No road sign recognition type provided
- 1 : SIGN_DETECTED A sign was detected
- 2 : SIGN_NOT_DETECTED No sign was detected where a sign was expected. E.g. derived from map content
- 3 : SIGN_TEMPORARY_INVALIDATED Temporary invalidated sign e.g by a Tape
- 4 : SIGN_MISMATCHED Sign detected but mismatched in a not further specified detail (location, supplemental sign, rotation, ...)

SignRecognition.RoadSignTypeEnum

Enumeration of various types of road signs may be detected by a camera setup on the vehicle. TODO all Sign Types Image or other defintion clarification

Enumeration Details

- 0 : ROAD_SIGN_TYPE_UNDEFINED No sign type specified
- 1 : SPEED_LIMIT_START **DE-StVO**:274 |**US-MUTCD**:R2
- 2 : SPEED_LIMIT_END **DE-StVO**: 278 |**US-MUTCD**:
- 3 : NO_OVERTAKING_PASSENGER_CARS_START **DE-StVO**: 276 |**US-MUTCD**:
- 4 : NO_OVERTAKING_PASSENGER_CARS_END **DE-StVO**: 280 |**US-MUTCD**:
- 5 : NO_OVERTAKING_TRUCKS_START **DE-StVO**: 277 |**US-MUTCD**:
- 6 : NO_OVERTAKING_TRUCKS_END **DE-StVO**: 281 |**US-MUTCD**:
- 7 : ALL_RESTRICTIONS_END **DE-StVO**: 282 |**US-MUTCD**:
- 8 : CITY_START **DE-StVO**: 310 |**US-MUTCD**:
- 9 : CITY_END **DE-StVO**: 311|**US-MUTCD**:
- 10 : MOTORWAY_START **DE-StVO**: 330.2|**US-MUTCD**:
- 11 : MOTORWAY_END **DE-StVO**: 330.1|**US-MUTCD**:
- 12 : CONSTRUCTION_START **DE-StVO**: 123|**US-MUTCD**: W21-1a
- 13 : CONSTRUCTION_END **DE-StVO**: |**US-MUTCD**: G20-2a
- 14 : PROTECTED_OVERTAKING_EXTRALANE A protected extralane for overtaking without specification if the lane is on the right or left side on the carriageway.
- 15 : PROTECTED_OVERTAKING_EXTRALANE_RIGHTSIDE
- 16 : PROTECTED_OVERTAKING_EXTRALANE_LEFTSIDE
- 17 : LANE_MERGE_RIGHT **DE-StVO**: 482|**US-MUTCD**: W9-2L or W9-1R or W4-2R
- 18 : LANE_MERGE_LEFT **DE-StVO**: 482|**US-MUTCD**: W9-2R or W9-1L or W4-2L
- 19 : LANE_MERGE_CENTER **DE-StVO**: 482|**US-MUTCD**:
- 20 : RAILWAY_CROSSING_PROTECTED **DE-StVO**: 150|**US-MUTCD**:
- 21 : RAILWAY_CROSSING_UNPROTECTED **DE-StVO**: 151|**US-MUTCD**:

- 22 : ROAD_NARROWS **DE-StVO**: 120 |**US-MUTCD**: W5-1a
- 23 : SHARP_CURVE

  Sharp curve warning sign withouth specification of the direction of the curve.

  DE-StVO: |**US-MUTCD**: W1-2
- 24 : SHARP_CURVE_LEFT **DE-StVO**: 103-10 |**US-MUTCD**: W1-2L or W1-11L
- 25 : SHARP_CURVE_RIGHT **DE-StVO**: 103-20 |**US-MUTCD**: W1-2R or W111R
- 26 : WINDING_ROAD_STARTING_LEFT **DE-StVO**: 105-10 |**US-MUTCD**: W1-3
- 27 : WINDING_ROAD_STARTING_RIGHT **DE-StVO**: 105-20 |**US-MUTCD**: W1-3
- 28 : STEEP_HILL

  Steep hill warning sign without specification of the direction (up- or downhill)

  DE-StVO: |**US-MUTCD**:W7-1
- 29 : STEEP_HILL_UPWARDS **DE-StVO**: 108 |**US-MUTCD**:
- 30 : STEEP_HILL_DOWNWARDS **DE-StVO**: 110 |**US-MUTCD**: W7-1
- 31 : STOP_SIGN **DE-StVO**: 206 |**US-MUTCD**:R1-1
- 32 : LATERAL_WIND **DE-StVO**: 117 |**US-MUTCD**:
- 33 : GENERAL_WARNING **DE-StVO**: 101 |**US-MUTCD**:
- 34 : RISK_OF_GROUNDING **DE-StVO**: |**US-MUTCD**: W10-5
- 35 : ANIMAL_CROSSING **DE-StVO**: 142 |**US-MUTCD**: W11-3
- 36 : ICY_CONDITIONS **DE-StVO**: 113 |**US-MUTCD**:
- 37 : SLIPPERY_ROAD **DE-StVO**: 114 |**US-MUTCD**: W8-5
- 38 : FALLING_ROCKS **DE-StVO**: 115 |**US-MUTCD**:
- 39 : SCHOOL_ZONE **DE-StVO**: |**US-MUTCD**: S1-1
- 40 : TRAMWAY_CROSSING
- 41 : CONGESTION_HAZARD **DE-StVO**: 124 |**US-MUTCD**:
- 42 : ACCIDENT_HAZARD **DE-StVO**: 101 with sup 1006-36 |**US-MUTCD**:
- 43 : PRIORITY_OVER_ONCOMING_TRAFFIC **DE-StVO**: 308 |**US-MUTCD**:
- 44 : YIELD_TO_ONCOMING_TRAFFIC **DE-StVO**: 208|**US-MUTCD**:
- 45 : PREFERENCE_ROAD_START **DE-StVO**: 306|**US-MUTCD**:
- 46 : PREFERENCE_ROAD_END **DE-StVO**: 307|**US-MUTCD**:
- 47 : TRAFFIC_CALMING_START **DE-StVO**: 325|**US-MUTCD**:
- 48 : TRAFFIC_CALMING_END **DE-StVO**: 326|**US-MUTCD**:
- 49 : ENVIRONMENTAL_AREA_START **DE-StVO**: 270.1|**US-MUTCD**:
- 50 : ENVIRONMENTAL_AREA_END **DE-StVO**: 270.2|**US-MUTCD**:
- 51 : GIVE_WAY **DE-StVO**: 205|**US-MUTCD**:R1-2
- 52 : ROUNDABOUT_INTERSECTION **DE-StVO**: 215|**US-MUTCD**: W2-6
- 53 : MANDATORY_TURN_RIGHT_ONLY **DE-StVO**: 209-20|**US-MUTCD**: R3-5R
- 54 : MANDATORY_TURN_LEFT_ONLY **DE-StVO**: 209-10|**US-MUTCD**: R3-5L
- 55 : MANDATORY_TURN_STRAIGHT_ONLY **DE-StVO**: 209-30|**US-MUTCD**: R3-5a
- 56 : NO_ENTRY **DE-StVO**: 267|**US-MUTCD**: R5-1
- 57 : ADVISORY_SPEED **DE-StVO**: |**US-MUTCD**: W13
- 58 : HIGH_OCCUPANCY_VEHICLE_LANE **DE-StVO**: |**US-MUTCD**: R3-12 or W16-11
- 59 : SHOULDER_OPEN_FOR_TRAFFIC **DE-StVO**: 223.1 |**US-MUTCD**:
- 60 : SHOULDER_CLOSE_FOR_TRAFFIC **DE-StVO**: 223.2|**US-MUTCD**:
- 61 : LANE_CLOSED dynamic or static sign providing information on a current closure of a specific lane.

- 62 : PEDESTRIAN_CROSSING **DE-StVO**: 350|**US-MUTCD**:W11-2 directly at the sign, not the "warning of crossing ahead"
- 63 : RAILWAY_CROSSING_GENERAL Railway crossing without indication if protected or unprotected

SignRecognition.RoadSignValidityEnum

Enumeration Details

- 0 : ROAD_SIGN_VALIDITY_UNDEFINED
- 1 : STARTING_IN
- 2 : VALID_FOR
- 3 : IN_RIGHT_DIRECTION
- 4 : IN_LEFT_DIRECTION
- 5 : ZONE
- 6 : BEGIN_OF_VALIDITY
- 7 : END_OF_VALIDITY

SpecificObservedEvent.NegotiationTypeEnum

Describes possible types of negotiation for an event object

Enumeration Details

- 0 : NEGOTIATION_TYPE_UNDEFINED
- 1 : NEGOTIATION_TYPE_SUPPORT An event has been received as a warning that was additionally identified by a vehicle. This support-Flag notifies the receiver that this event already is known and still existing.
- 2 : NEGOTIATION_TYPE_CONTRADICT An event has been expected (e.g. by receiving a warning from a service provider) it was additionally identified by a vehicle to be wrong. This contradict-Flag notifies the receiver that the provided event may be wrong or already outdated and has not been identified as submitted.
- 3 : NEGOTIATION_TYPE_CANCEL An event has been send by the vehicle. The cancel-flag notifies the receiver that this event is being canceled by the “creator” of the event.

SpecificObservedEvent.RelevanceDistance

Enumeration Details

- 0 : lessThan50M
- 1 : lessThan100M
- 2 : lessThan200M
- 3 : lessThan500M
- 4 : lessThan1000M
- 5 : lessThan5KM

- 6 : lessThan10KM
- 7 : over10KM

SpecificObservedEvent.RelevanceEventReferenceEnum

Enumeration Details

- 0 : allStreamsTraffic
- 1 : upStreamTraffic
- 2 : downStreamTraffic

SpecificObservedEvent.RelevanceTrafficDirectionEnum

Enumeration Details

- 0 : allTrafficDirections
- 1 : sameTraffic
- 2 : oppositeTraffic

SpecificObservedEvent.SpecificObservedEventCause Enum

Enumeration Details

- 0 : reserved
- 1 : trafficCondition
- 2 : accident
- 3 : roadWorks
- 6 : adverseWeatherCondition_Adhesion
- 9 : hazardousLocation_SurfaceCondition
- 10 : hazardousLocation_ObstacleOnTheRoad
- 11 : hazardousLocation_AnimalOnTheRoad
- 12 : humanPresenceOnTheRoad
- 14 : wrongWayDriving
- 15 : rescueAndRecoveryWorkInProgress
- 17 : adverseWeatherCondition_ExtremeWeatherCondition
- 18 : adverseWeatherCondition_Visibility
- 19 : adverseWeatherCondition_Precipitation
- 26 : slowVehicle
- 27 : dangerousEndOfQueue
- 91 : vehicleBreakdown
- 92 : postCrash

- 93 : humanProblem
- 94 : stationaryVehicle
- 95 : emergencyVehicleApproaching
- 96 : hazardousLocation_DangerousCurve
- 97 : collisionRisk
- 98 : signalViolation
- 99 : dangerousSituation

SpecificObservedEventSubCauseEnum.AccidentSubCauseEnum

Enumeration Details

- 0 : unavailableAccidentSubCause
- 1 : multiVehicleAccident
- 2 : heavyAccident
- 3 : accidentInvolvingLorry
- 4 : accidentInvolvingBus
- 5 : accidentInvolvingHazardousMaterials
- 6 : accidentOnOppositeLane
- 7 : unsecuredAccident
- 8 : assistanceRequested

SpecificObservedEventSubCauseEnum.AdverseWeatherCondition_AdhesionSubCauseEnum

Enumeration Details

- 0 : unavailableAdverseWeatherCondition_AdhesionSubCause
- 1 : heavyFrostOnRoad
- 2 : fuelOnRoad
- 3 : mudOnRoad
- 4 : snowOnRoad
- 5 : iceOnRoad
- 6 : blackIceOnRoad
- 7 : oilOnRoad
- 8 : looseChippings
- 9 : instantBlackIce
- 10 : roadsSalted

SpecificObservedEventSubCauseEnum.AdverseWeatherCondition_ExtremeWeatherConditionSubCauseCode

Enumeration Details

- 0 : unavailableAdverseWeatherCondition_ExtremeWeatherConditionSubCause
- 1 : strongWinds
- 2 : damagingHail
- 3 : hurricane
- 4 : thunderstorm
- 5 : tornado
- 6 : blizzard

SpecificObservedEventSubCauseEnum.AdverseWeatherCondition_PrecipitationSubCauseCode

Enumeration Details

- 0 : unavailableAdverseWeatherCondition_PrecipitationSubCause
- 1 : heavyRainPrecipitation
- 2 : heavySnowFallPrecipitation
- 3 : softHail

SpecificObservedEventSubCauseEnum.AdverseWeatherCondition_VisibilitySubCauseCode

Enumeration Details

- 0 : unavailableAdverseWeatherCondition_VisibilitySubCause
- 1 : fog
- 2 : smoke
- 3 : heavySnowFallVisibility
- 4 : heavyRainVisibility
- 5 : heavyHail
- 6 : lowSunGlare
- 7 : sandstorms
- 8 : swarmsOfInsects

SpecificObservedEventSubCauseEnum.CollisionRiskSubCauseCode

Enumeration Details

- 0 : unavailableCollisionRiskSubCause
- 1 : longitudinalCollisionRisk
- 2 : crossingCollisionRisk
- 3 : lateralCollisionRisk
- 4 : vulnerableRoadUser

SpecificObservedEventSubCauseEnum.DangerousEndOfQueueSubCauseCode

Enumeration Details

- 0 : unavailableDangerousEndOfQueueSubCause
- 1 : suddenEndOfQueue
- 2 : queueOverHill
- 3 : queueAroundBend
- 4 : queueInTunnel

SpecificObservedEventSubCauseEnum.DangerousSituationSubCauseCode

Enumeration Details

- 0 : unavailableDangerousSituationSubCause
- 1 : emergencyElectronicBrakeEngaged
- 2 : preCrashSystemEngaged
- 3 : espEngaged
- 4 : absEngaged
- 5 : aebEngaged
- 6 : brakeWarningEngaged
- 7 : collisionRiskWarningEngaged

SpecificObservedEventSubCauseEnum.EmergencyVehicleApproachingSubCauseCode

Enumeration Details

- 0 : `unavailableEmergencyVehicleApproachingSubCause`
- 1 : `emergencyVehicleApproaching`
- 2 : `prioritizedVehicleApproaching`

SpecificObservedEventSubCauseEnum.HazardousLocation_AnimalOnTheRoadSubCauseCode

Enumeration Details

- 0 : `unavailableHazardousLocation_AnimalOnTheRoadSubCause`
- 1 : `wildAnimals`
- 2 : `herdOfAnimals`
- 3 : `smallAnimals`
- 4 : `largeAnimals`

SpecificObservedEventSubCauseEnum.HazardousLocation_DangerousCurveSubCauseCode

Enumeration Details

- 0 : `unavailableHazardousLocation_DangerousCurveSubCause`
- 1 : `dangerousLeftTurnCurve`
- 2 : `dangerousRightTurnCurve`
- 3 : `multipleCurvesStartingWithUnknownTurningDirection`
- 4 : `multipleCurvesStartingWithLeftTurn`
- 5 : `multipleCurvesStartingWithRightTurn`

SpecificObservedEventSubCauseEnum.HazardousLocation_ObstacleOnTheRoadSubCauseCode

Enumeration Details

- 0 : `unavailableHazardousLocation_ObstacleOnTheRoadSubCause`
- 1 : `shedload`
- 2 : `partsOfVehicles`
- 3 : `partsOfTyres`
- 4 : `bigObjects`
- 5 : `fallenTrees`
- 6 : `hubCaps`
- 7 : `waitingVehicles`

SpecificObservedEventSubCauseEnum.HazardousLocation_SurfaceConditionSubCauseCode

Enumeration Details

- 0 : `unavailableHazardousLocation_SurfaceConditionSubCause`
- 1 : `rockfalls`
- 2 : `earthquakeDamage`
- 3 : `sewerCollapse`
- 4 : `subsidence`
- 5 : `snowDrifts`
- 6 : `stormDamage`
- 7 : `burstPipe`
- 8 : `volcanoEruption`
- 9 : `fallingIce`

SpecificObservedEventSubCauseEnum.HumanPresenceOnTheRoadSubCauseCode

Enumeration Details

- 0 : `unavailableHumanPresenceOnTheRoadSubCause`
- 1 : `childrenOnRoadway`
- 2 : `cyclistOnRoadway`
- 3 : `motorcyclistOnRoadway`

SpecificObservedEventSubCauseEnum.HumanProblemSubCauseCode

Enumeration Details

- 0 : unavailableHumanProblemSubCause
- 1 : glycemiaProblem
- 2 : heartProblem

SpecificObservedEventSubCauseEnum.PostCrashSubCauseCode

Enumeration Details

- 0 : unavailablePostCrashSubCause
- 1 : accidentWithoutECallTriggered
- 2 : accidentWithECallManuallyTriggered
- 3 : accidentWithECallAutomaticallyTriggered
- 4 : accidentWithECallTriggeredWithoutAccessToCellularNetwork

SpecificObservedEventSubCauseEnum.RescueAndRecoveryWorkInProgressSubCauseCode

Enumeration Details

- 0 : unavailableRescueAndRecoveryWorkInProgressSubCause
- 1 : emergencyVehicles
- 2 : rescueHelicopterLanding
- 3 : policeActivityOngoing
- 4 : medicalEmergencyOngoing
- 5 : childAbductionInProgress

SpecificObservedEventSubCauseEnum.RoadworksSubCauseEnum

Enumeration Details

- 0 : unavailableRoadworksSubCause
- 1 : majorRoadWorks
- 2 : roadMarkingWork
- 3 : slowMovingRoadMaintenance
- 4 : shortTermStationaryRoadworks
- 5 : streetCleaning
- 6 : winterService

SpecificObservedEventSubCauseEnum.SignalViolationSubCauseCode

Enumeration Details

- 0 : unavailableSignalViolationSubCauseCodeSubCause
- 1 : stopSignViolation
- 2 : trafficLightViolation
- 3 : turningRegulationViolation

SpecificObservedEventSubCauseEnum.SlowVehicleSubCauseCode

Enumeration Details

- 0 : unavailableSlowVehicleSubCause
- 1 : maintenanceVehicle
- 2 : vehiclesSlowingToLookAtAccident
- 3 : abnormalLoad
- 4 : abnormalWideLoad
- 5 : convoy
- 6 : snowplough
- 7 : deicing
- 8 : saltingVehicles

SpecificObservedEventSubCauseEnum.StationaryVehicleSubCauseCode

Enumeration Details

- 0 : unavailableStationaryVehicleSubCause
- 1 : humanProblem
- 2 : vehicleBreakdown
- 3 : postCrash
- 4 : publicTransportStop
- 5 : carryingDangerousGoods

SpecificObservedEventSubCauseEnum.TrafficConditionSubCauseEnum

Enumeration Details

- 0 : unavailableTrafficConditionSubCause
- 1 : increasedVolumeOfTraffic
- 2 : trafficJamSlowlyIncreasing
- 3 : trafficJamIncreasing
- 4 : trafficJamStronglyIncreasing
- 5 : trafficStationary
- 6 : trafficJamSlightlyDecreasing
- 7 : trafficJamDecreasing
- 8 : trafficJamStronglyDecreasing

SpecificObservedEventSubCauseEnum.VehicleBreakdownSubCauseCode

Enumeration Details

- 0 : unavailableVehicleBreakdownSubCause
- 1 : lackOfFuel
- 2 : lackOfBatteryPower
- 3 : engineProblem
- 4 : transmissionProblem
- 5 : engineCoolingProblem
- 6 : brakingSystemProblem

- 7 : steeringProblem
- 8 : tyrePuncture

SpecificObservedEventSubCauseEnum.WrongWayDrivingSubCauseCode

Enumeration Details

- 0 : unavailableWrongWayDrivingSubCause
- 1 : wrongLane
- 2 : wrongDirection

TrafficSignalHeadRecognition.TrafficSignalHeadDesignTypeEnum

Enumeration Details

- 0 : TRAFFIC_SIGNAL_HEAD_TYPE_UNDEFINED No Traffic Signal Head Type is provided
- 1 : TRAFFIC_SIGNAL_HEAD_TYPE_UNKNOWN Traffic Signal Head Type could not be identified
- 2 : TRAFFIC_SIGNAL_HEAD_TYPE_REDYELLOWGREEN Standard Traffic Signal Head with Red, Yellow, Green
- 3 : TRAFFIC_SIGNAL_HEAD_TYPE_REDYELLOW Standard Traffic Signal Head with Red, Yellow – No green
- 4 : TRAFFIC_SIGNAL_HEAD_TYPE_3LIGHTS Standard Traffic Signal Head with 3 Lights without Color identification.
- 5 : TRAFFIC_SIGNAL_HEAD_TYPE_2LIGHTS Standard Traffic Signal Head with 2 Lights without Color identification
- 6 : TRAFFIC_SIGNAL_HEAD_TYPE_2YELLOWLIGHTS Traffic Signal with 2 yellow lights only

TrafficSignalHeadRecognition.TrafficSignalHeadOrientationTypeEnum

Enumeration Details

- 0 : TRAFFIC_SIGNAL_HEAD_ORIENTATION_TYPE_UNDEFINED
- 1 : TRAFFIC_SIGNAL_HEAD_ORIENTATION_TYPE_VERTICAL
- 2 : TRAFFIC_SIGNAL_HEAD_ORIENTATION_TYPE_HORIZONTAL

VehicleManeuverEvent.VehicleManeuverTypeEvent

Describes possible types of vehicle maneuvers

Enumeration Details

- 0 : `VEHICLE_MANEUVER_TYPE_UNDEFINED`
- 1 : `LANE_CHANGE` A lane change has been executed in an undefined direction
- 2 : `LANE_CHANGE_RIGHT` A lane change has been executed to the right.
- 3 : `LANE_CHANGE_LEFT` A lane change has been executed to the left.
- 11 : `PARKED_IN` The vehicle parked. (engine off, doors locked, ...)
- 12 : `PARKED_IN_LATERAL` The vehicle parked in into a perpendicular parking space.
- 13 : `PARKED_IN_DIAGONAL` The vehicle parked into a diagonal parking space
- 14 : `PARKED_IN_LONGITUDINAL` The vehicle parked in to a parallel parking space
- 15 : `PARKED_OUT` The vehicle parked out and is driving.

VehicleMetaData.FuelTypeEnum

Enumeration Details

- 0 : `FUEL_TYPE_UNDEFINED`
- 1 : `FUEL_TYPE_OTHER` Refers to any other non-specified Fuel Type. (UNIT: percentage)
- 2 : `FUEL_TYPE_GASOLINE_L` Refers to Fuel used for petrol engins with igntion systems. (UNIT: liter)
- 3 : `FUEL_TYPE_DIESEL_L` Refers to Fuel used in engines with spontaneous combustion (UNIT: liter)
- 4 : `FUEL_TYPE_AUTOGAS_KG` Refers to Liquid petrol gas (UNIT: kilograms)
- 5 : `FUEL_TYPE_BATTERY_AH` Refers to an energy accumulator (UNIT: ampere hours)
- 6 : `FUEL_TYPE_HYDROGEN_KG` Refers to hydrogen used in fuel cell vehicles (UNIT: kilograms)

VehicleMetaData.VehicleHeightDetail.LateralVehicleHeightReferencePointEnum

Enumeration Details

- 0 : `LATERAL_VEHICLE_HEIGHT_DETAIL_TYPE_UNDEFINED`
- 1 : `LATERAL_VEHICLE_HEIGHT_DETAIL_TYPE_LEFT`
- 2 : `LATERAL_VEHICLE_HEIGHT_DETAIL_TYPE_MIDDLE`
- 3 : `LATERAL_VEHICLE_HEIGHT_DETAIL_TYPE_RIGHT`

VehicleMetaData.VehicleHeightDetail.LongitudinalVehicleHeightReferencePointEnum

Enumeration Details

- 0 : `LONGITUDINAL_VEHICLE_HEIGHT_DETAIL_TYPE_UNDEFINED`
- 1 : `LONGITUDINAL_VEHICLE_HEIGHT_DETAIL_TYPE_FRONT`
- 2 : `LONGITUDINAL_VEHICLE_HEIGHT_DETAIL_TYPE_MIDDLE`
- 3 : `LONGITUDINAL_VEHICLE_HEIGHT_DETAIL_TYPE_REAR`

VehicleMetaData.VehicleTypeGenericEnum

Enumeration Details

- 0 : `VEHICLE_TYPE_UNDEFINED` No value provided
- 1 : `BUS` The vehicle is a bus
- 2 : `DELIVERY_TRUCK` The vehicle is a (smaller) delivery truck)
- 3 : `EMERGENCY_VEHICLE` The vehicle serves as an emergency vehicle
- 4 : `MOTORCYCLE` The vehicle is a motorcycle
- 5 : `PASSENGER_CAR` The vehicle is a passenger car
- 6 : `TAXI` The vehicle is a taxi
- 7 : `TRANSPORT_TRUCK` The vehicle is a (larger) transport truck
- 8 : `MOPED` The vehicle is a moped
- 9 : `TRAILER` The vehicle is a trailer
- 10 : `TRAM` The vehicle is a tram
- 11 : `UNKNOWN_VEHICLE` The vehicle type is unknown

VehicleStatus.DoorStateMessage.DoorIdentificationEnum

The identification of a door is done by this Enumeration. In case of only one door per side, the Front-door pair is used. In case of multiple doors per side, the first door is the front door, the last door is the rear door and all intermediate doors are other doors. The Engine_Door does not imply front or rear as well as the TRUNK.

Enumeration Details

- 0 : `DOOR_IDENTIFICATION_UNDEFINED` Undefined door
- 1 : `DOOR_IDENTIFICATION_FRONT_LEFT` References to front left door
- 2 : `DOOR_IDENTIFICATION_FRONT_RIGHT` References to front right door

- 3 : DOOR_IDENTIFICATION_REAR_LEFT References to rear left door
- 4 : DOOR_IDENTIFICATION_REAR_RIGHT References to rear right door
- 5 : DOOR_IDENTIFICATION_OTHER_LEFT References to other door on the left side
- 6 : DOOR_IDENTIFICATION_OTHER_RIGHT References to other door on the right side
- 10 : DOOR_IDENTIFICATION_TRUNK References to the trunk door of the vehicle
- 20 : DOOR_IDENTIFICATION_ENGINE_DOOR References to the engine hood of the vehicle
- 21 : DOOR_IDENTIFICATION_FUEL_ENTRY References to the fuel hatch
- 22 : DOOR_IDENTIFICATION_OTHER_MAINTENANCE References to any maintenance hatch

VehicleStatus.DoorStateMessage.DoorLockStateEnum

The door lock state provides information about the specified door.

Enumeration Details

- 0 : DOOR_LOCK_STATE_UNDEFINED Lock state is undefined
- 1 : DOOR_LOCK_STATE_UNLOCKED Door is unlocked
- 2 : DOOR_LOCK_STATE_LOCKED Door is locked
- 3 : DOOR_LOCK_STATE_NO_LOCK Door not equiped with lock
- 4 : DOOR_LOCK_STATE_AUTO_LOCKED The door is auto-locked programmatically e.g. by starting of movement

VehicleStatus.DoorStateMessage.DoorStateEnum

The doorState describes the current state of the specified door by open or closed. By way of example, a door may be in state open when the vehicle would provide a warning of an open door while moving.

Enumeration Details

- 0 : DOOR_STATE_UNDEFINED Door state is undefined
- 1 : DOOR_STATE_OPEN Door is open
- 2 : DOOR_STATE_CLOSE Door is closed

VehicleStatus.EngineStateEnum

The description of possible engine states of a vehicle

Enumeration Details

- 0 : ENGINE_STATE_UNDEFINED Engine state is undefined
- 1 : ENGINE_OFF Engine is turned off
- 2 : ENGINE_ECO_OFF The engine is automatically turned off, due to e.g. a stop at traffic lights to safe fuel.

- 3 : `ENGINE_SET` For e.g. EV engines, that do not run while standing, where the engine and vehicle can start whenever the accelerator pedal is pressed. e.g. EV where the vehicle can start driving whenever the accelerator pedal is pressed.
- 4 : `ENGINE_RUN` Engine is currently in running set (power chain is rotating)

VehicleStatus.IgnitionStateEnum

The description of possible ignition states of a vehicle

Enumeration Details

- 0 : `IGNITION_STATE_UNDEFINED` Ignition state is not defined
- 1 : `IGNITION_OFF` Ignition is turned off
- 2 : `IGNITION_ON` Ignition is turned on

VehicleStatus.LightStateBitfield

No actual use. This is a description of the Bitfield for Light state

Enumeration Details

- 1 : `LOWBEAMS`
- 2 : `HIGHBEAMS`
- 4 : `FOGLAMP_FRONT`
- 8 : `FOGLAMP_REAR`
- 16 : `HAZARD`
- 32 : `LEFT_TURN`
- 64 : `RIGHT_TURN`

VehicleStatus.MaintenanceLightStateBitfield

Enumeration Details

- 1 : `WARNING_ENGINE_CONTROL`
- 2 : `WARNING_OIL_PRESSURE`
- 4 : `WARNING_COOLANT_TEMP`
- 8 : `WARNING_VEHICLE_SERVICE`
- 16 : `WARNING_BATTERY_CHARGING`
- 32 : `WARNING_TIRE_PRESSURE`
- 64 : `WARNING_LAMP_OUT`
- 128 : `WARNING_OTHER_HIGH_PRIO`
- 256 : `WARNING_OTHER_LOW_PRIO`

VehicleStatus.TransmissionModeEnum

a list of possible enumeration transmission modes of the vehicle.

Enumeration Details

- 0 : `TRANSMISSION_MODE_UNDEFINED`
- 1 : `PARK` The vehicle is in park. This also applies if the engine is permanently switched off.
- 2 : `COASTING` The vehicle is in coasting mode (neutral) and engine is not permanently switched off.
- 3 : `DRIVE` The vehicle is in forward drive mode and engine is not permanently switched off.
- 4 : `REVERSE` The vehicle is in reverse drive mode and engine is not permanently switched off.

VehicleStatus.VentilationStateMessage.CirculationSettingEnum

The description of the circulation setting. If the circulation is on, external air is cut off from inflow into the passenger room.

Enumeration Details

- 0 : `CIRCULATION_SETTING_UNDEFINED` Undefined ventilation state
- 1 : `CIRCULATION_SETTING_OFF` Circulation is disabled, inflow of external air
- 2 : `CIRCULATION_SETTING_ON` Circulation is enabled, outside air is cut off.

VehicleStatus.VentilationStateMessage.VentilationSettingEnum

The description of the setting of the ventilation unit.

Enumeration Details

- 0 : `VENTILATION_SETTING_UNDEFINED` Undefined ventilation setting
- 1 : `VENTILATION_SETTING_AUTOMATIC` Ventilation setting to automatic
- 2 : `VENTILATION_SETTING_MANUAL` Ventilation setting to manual

VehicleStatus.VentilationStateMessage.VentilationStateEnum

The description of the ventilation state within the vehicle. This information does not provide the actual speed of the fan.

Enumeration Details

- 0 : `VENTILATION_STATE_UNDEFINED` Undefined ventilation state
- 1 : `VENTILATION_STATE_ON` Ventilation turned on
- 2 : `VENTILATION_STATE_OFF` Ventilation turned off

VehicleStatus.WiperStateEnum

Enumeration Details

- 0 : `WIPER_STATE_UNDEFINED`
- 1 : `WIPING_OFF`
- 2 : `WIPING_SLOW`
- 3 : `WIPING_MEDIUM`
- 4 : `WIPING_FAST`
- 5 : `WIPING_INTERVALL`

WheelReferenceBitfield

Enumeration Details

- 1 : `frontAxleLeft`
- 2 : `frontAxleRight`
- 4 : `rearAxleLeft`
- 8 : `rearAxleRight`

Appendix
Appendix

Topics:

This appendix contains general descriptions, futher information not captured in the Data Elements section, and also topics other than immediately regarding changes and issues in the release such as release overview, technical specifications, dependencies, 3rd party copyright notices, release quality and related documents.

General Information

```
3.1.1 Timestamp
PROTOBUF DEFINITION: timeStampUTC_ms
DATA TYPE: int64
UNIT: milliseconds ( = seconds/1000)
RANGE: [ 0,  ? ]
RESOULTION: 1milliseconds
DESCRIPTION: A timestamp is used to indicate an absolute time value. All timestamps used in any
 of the sensor ingestion API are based on UTC and measured in milliseconds since January 1, 1970.
 Please note that in the context of sensor data collection a GPS time may be used during data
 collection. GPS time is different from UTC. At the time of this writing, the GPS time is 16 seconds
 ahead of UTC. It is the responsibility of the data submitter to convert any time stamp into UTC
 before data submission. The timestamp does not count in leap seconds. Any necessary transformation
 is within the responsibility of the data submitter.
It is required that the timestamp is consistent at all times throughout all event messages. Hereby,
 the timestamp represents the point of time at which the sensor data has been captured and not the
 time at which the sensor data is available. By way of example, a position estimate of at time t=0
 is made available at t=1 and a map matching algorithm provides a matched position at t=5, then the
 map matched position estimate is provided as sensor data at t=6 with the data content #t=0#.



3.24 Specific Observed Event
3.24.1 Timestamp
PROTOBUF DEFINITION: timeStampUTC_ms
DATA TYPE: int64
MANDATORY/OPTIONAL: Mandatory
UNIT: milliseconds in UTC
DESCRIPTION: For general description of the timestamp see Timestamp
3.24.2 Cause
PROTOBUF DEFINITION: cause
DATA TYPE: SpecificObservedEventCauseEnum
MANDATORY/OPTIONAL: Optional
DESCRIPTION: Cause value of specific observed event sourced from ETSI spec [1] below.
MESSAGE
EnumPosition
PositionType
Description
0
reserved
Ref [1], Section A.10, page 21, bullet 1
1
trafficCondition
Ref [1], Section A.10, page 21, bullet 2
2
accident
Ref [1], Section A.10, page 21, bullet 3
3
roadWorks
Ref [1], Section A.10, page 21, bullet 4
6
adverseWeatherCondition_Adhesion
Ref [1], Section A.10, page 21, bullet 5
9
hazardousLocation_SurfaceCondition
Ref [1], Section A.10, page 21, bullet 6
10
hazardousLocation_ObstacleOnTheRoad
Ref [1], Section A.10, page 21, bullet 7
11
```

```
hazardousLocation_AnimalOnTheRoad
Ref [1], Section A.10, page 21, bullet 8
12
humanPresenceOnTheRoad
Ref [1], Section A.10, page 21, bullet 9
14
wrongWayDriving
Ref [1], Section A.10, page 21, bullet 10
15
rescueAndRecoveryWorkInProgress
Ref [1], Section A.10, page 21, bullet 11
17
adverseWeatherCondition_ExtremeWeatherCondition
Ref [1], Section A.10, page 21, bullet 12
18
adverseWeatherCondition_Visibility
Ref [1], Section A.10, page 21, bullet 13
19
adverseWeatherCondition_Precipitation
Ref [1], Section A.10, page 21, bullet 14
26
slowVehicle
Ref [1], Section A.10, page 21, bullet 15
27
dangerousEndOfQueue
Ref [1], Section A.10, page 21, bullet 16
91
vehicleBreakdown
Ref [1], Section A.10, page 21, bullet 17
92
postCrash
Ref [1], Section A.10, page 21, bullet 18
93
humanProblem
Ref [1], Section A.10, page 21, bullet 19
94
stationaryVehicle
Ref [1], Section A.10, page 21, bullet 20
95
emergencyVehicleApproaching
Ref [1], Section A.10, page 21, bullet 21
96
hazardousLocation_DangerousCurve
Ref [1], Section A.10, page 21, bullet 22
97
collisionRisk
Ref [1], Section A.10, page 21, bullet 23
98
signalViolation
Ref [1], Section A.10, page 21, bullet 24
99
dangerousSituation
Ref [1], Section A.10, page 21, bullet 25

3.24.3 Sub cause
PROTOBUF DEFINITION: subcause
DATA TYPE: SpecificObservedEventSubCauseEnum
MANDATORY/OPTIONAL: Optional
DESCRIPTION: This is used to provide more detailed information of the event related to the
 causeCode . The value of the sub cause code is based on the TPEG TEC specification as defined
 in ETSI [1]. The subCauseCode  shall be set to 0 (unavailable) if no specific information of the
 subCauseCode is available. The appropriate subcause for the assocated with the event above will be
 used.
Following are the list of SubCauseEnum supported:
3.24.3.1 Traffic Condition Sub Cause
PROTOBUF DEFINITION: trafficConditionSubCause
DATA TYPE: TrafficConditionSubCauseEnum
```

```
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.84, page 59, bullet 1
1
increasedVolumeOfTraffic
Ref [1], Section A.84, page 59, bullet 2
2
trafficJamSlowlyIncreasing
Ref [1], Section A.84, page 59, bullet 3
3
trafficJamIncreasing
Ref [1], Section A.84, page 59, bullet 4
4
trafficJamStronglyIncreasing
Ref [1], Section A.84, page 59, bullet 5
5
trafficStationary
Ref [1], Section A.84, page 59, bullet 6
6
trafficJamSlightlyDecreasing
Ref [1], Section A.84, page 59, bullet 7
7
trafficJamDecreasing
Ref [1], Section A.84, page 59, bullet
8
trafficJamStronglyDecreasing
Ref [1], Section A.84, page 59, bullet 9
3.24.3.2 Accident Sub Cause
PROTOBUF DEFINITION: accidentSubCause
DATA TYPE: AccidentSubCauseEnum
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.3, page 15, bullet 1
1
multiVehicleAccident
Ref [1], Section A.3, page 15, bullet 2
2
heavyAccident
Ref [1], Section A.3, page 15, bullet 3
3
accidentInvolvingLorry
Ref [1], Section A.3, page 15, bullet 4
4
accidentInvolvingBus
Ref [1], Section A.3, page 15, bullet 5
5
accidentInvolvingHazardousMaterials
Ref [1], Section A.3, page 15, bullet 6
6
accidentOnOppositeLane
Ref [1], Section A.3, page 15, bullet 7
7
unsecuredAccident
Ref [1], Section A.3, page 15, bullet 8
8
assistanceRequested
Ref [1], Section A.3, page 15, bullet 9
```

```
3.24.3.3 Roadworks Sub Cause
PROTOBUF DEFINITION: roadworksSubCause
DATA TYPE: RoadworksSubCauseEnum
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.66, page 50, bullet 1
1
majorRoadWorks
Ref [1], Section A.66, page 50, bullet 2
2
roadMarkingwork
Ref [1], Section A.66, page 50, bullet 3
3
slowMovingRoadMaintenance
Ref [1], Section A.66, page 50, bullet 4
4
shortTermStationaryRoadworks
Ref [1], Section A.66, page 50, bullet 5
5
streetCleaning
Ref [1], Section A.66, page 50, bullet 6
6
winterService
Ref [1], Section A.66, page 50, bullet 7
3.24.3.4 Adverse Weather Condition Adhesion Sub Cause
PROTOBUF DEFINITION: adverseWeatherCondition_AdhesionSubCause
DATA TYPE: AdverseWeatherCondition_AdhesionSubCauseEnum
MANDATORY/OPTIONAL:  Optional

MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.4, page 16, bullet 1
1
heavyFrostOnRoad
Ref [1], Section A.4, page 16, bullet 2
2
fuelOnRoad
Ref [1], Section A.4, page 16, bullet 3
3
mudOnRoad
Ref [1], Section A.4, page 16, bullet 4
4
snowOnRoad
Ref [1], Section A.4, page 16, bullet 5
5
iceOnRoad
Ref [1], Section A.4, page 16, bullet 6
6
blackIceOnRoad
Ref [1], Section A.4, page 16, bullet 7
7
oilOnRoad
Ref [1], Section A.4, page 16, bullet 8
8
looseChippings
Ref [1], Section A.4, page 16, bullet 9
9
instantBlackIce
```

```
Ref [1], Section A.4, page 16, bullet 10
10
roadsSalted
Ref [1], Section A.4, page 16, bullet 11
3.24.3.5 Hazardous Location Surface Condition Sub Cause
PROTOBUF DEFINITION: hazardousLocation_SurfaceConditionSubCause
DATA TYPE: HazardousLocation_SurfaceConditionSubCauseCode
MANDATORY/OPTIONAL:  Optional


MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.33, page 34, bullet 1
1
rockfalls
Ref [1], Section A.33, page 34, bullet 2
2
earthquakeDamage
Ref [1], Section A.33, page 34, bullet 3
3
sewerCollapse
Ref [1], Section A.33, page 34, bullet 4
4
subsidence
Ref [1], Section A.33, page 34, bullet 5
5
snowDrifts
Ref [1], Section A.33, page 34, bullet 6
6
stormDamage
Ref [1], Section A.33, page 34, bullet 7
7
burstPipe
Ref [1], Section A.33, page 34, bullet 8
8
volcanoEruption
Ref [1], Section A.33, page 34, bullet 9
9
fallingIce
Ref [1], Section A.33, page 34, bullet 10
3.24.3.6 Hazardous Location Obstacle On The Road Sub Cause
PROTOBUF DEFINITION: hazardousLocation_ObstacleOnTheRoadSubCause
DATA TYPE: HazardousLocation_ObstacleOnTheRoadSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.32, page 33, bullet 1
1
shedload
Ref [1], Section A.32, page 33, bullet 2
2
partsOfVehicles
Ref [1], Section A.32, page 33, bullet 3
3
partsOfTyres
Ref [1], Section A.32, page 33, bullet 4
4
bigObjects
Ref [1], Section A.32, page 33, bullet 5
```

```
5
fallenTrees
Ref [1], Section A.32, page 33, bullet 6
6
hubCaps
Ref [1], Section A.32, page 33, bullet 7
7
waitingVehicles
Ref [1], Section A.32, page 33, bullet 8
3.24.3.7 Hazardous Location Animal On The Road Sub Cause
PROTOBUF DEFINITION: hazardousLocation_AnimalOnTheRoadSubCause
DATA TYPE: HazardousLocation_AnimalOnTheRoadSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.30, page 31, bullet 1
1
wildAnimals
Ref [1], Section A.30, page 31, bullet 2
2
herdOfAnimals
Ref [1], Section A.30, page 31, bullet 3
3
smallAnimals
Ref [1], Section A.30, page 31, bullet 4
4
largeAnimals
Ref [1], Section A.30, page 31, bullet 5
3.24.3.8 Human Presence On The Road Sub Cause
PROTOBUF DEFINITION: humanPresenceOnTheRoadSubCause
DATA TYPE: HumanPresenceOnTheRoadSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.37, page 36, bullet 1
1
childrenOnRoadway
Ref [1], Section A.37, page 36, bullet 2
2
cyclistOnRoadway
Ref [1], Section A.37, page 36, bullet 3
3
motorcyclistOnRoadway
Ref [1], Section A.37, page 36, bullet 4
3.24.3.9 Wrong Way Driving  Sub Cause
PROTOBUF DEFINITION: wrongWayDrivingSubCause
DATA TYPE: WrongWayDrivingSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.99, page 65, bullet 1
1
wrongLane
Ref [1], Section A.99, page 65, bullet 2
2
```

```
wrongDirection
Ref [1], Section A.99, page 65, bullet 3

3.24.3.10  Rescue And Recovery Work In Progress Sub Cause
PROTOBUF DEFINITION: rescueAndRecoveryWorkInProgressSubCause
DATA TYPE: RescueAndRecoveryWorkInProgressSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.64, page 49, bullet 1
1
emergencyVehicles
Ref [1], Section A.64, page 49, bullet 2
2
rescueHelicopterLanding
Ref [1], Section A.64, page 49, bullet 3
3
policeActivityOngoing
Ref [1], Section A.64, page 49, bullet 4
4
medicalEmergencyOngoing
Ref [1], Section A.64, page 49, bullet 5
5
childAbductionInProgress
Ref [1], Section A.64, page 49, bullet 6

3.24.3.11  Adverse Weather Condition Extreme Weather Condition Sub Cause
PROTOBUF DEFINITION: adverseWeatherCondition_ExtremeWeatherConditionSubCause
DATA TYPE: AdverseWeatherCondition_ExtremeWeatherConditionSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.5, page 17, bullet 1
1
strongWinds
Ref [1], Section A.5, page 17, bullet 2
2
damagingHail
Ref [1], Section A.5, page 17, bullet 3
3
hurricane
Ref [1], Section A.5, page 17, bullet 4
4
thunderstorm
Ref [1], Section A.5, page 17, bullet 5
5
tornado
Ref [1], Section A.5, page 17, bullet 6
6
blizzard
Ref [1], Section A.5, page 17, bullet 7
3.24.3.12  Adverse Weather Condition Visibility Sub Cause
PROTOBUF DEFINITION: adverseWeatherCondition_VisibilitySubCause
DATA TYPE: AdverseWeatherCondition_VisibilitySubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
```

```
0
unavailable
Ref [1], Section A.7, page 18, bullet 1
1
fog
Ref [1], Section A.7, page 18, bullet 2
2
smoke
Ref [1], Section A.7, page 18, bullet 3
3
heavySnowFallVisibility
Ref [1], Section A.7, page 18, bullet 4
4
heavyRainVisibility
Ref [1], Section A.7, page 18, bullet 5
5
heavyHail
Ref [1], Section A.7, page 18, bullet 6
6
lowSunGlare
Ref [1], Section A.7, page 18, bullet 7
7
sandstorms
Ref [1], Section A.7, page 18, bullet 8
8
swarmsOfInsects
Ref [1], Section A.7, page 18, bullet 9
3.24.3.13  Adverse Weather Condition Precipitation Sub Cause
PROTOBUF DEFINITION: adverseWeatherCondition_PrecipitationSubCause
DATA TYPE: AdverseWeatherCondition_PrecipitationSubCauseCode
MANDATORY/OPTIONAL:  Optional

MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.6, page 17, bullet 1
1
heavyRainPrecipitation
Ref [1], Section A.6, page 17, bullet 2
2
heavySnowFallPrecipitation
Ref [1], Section A.6, page 17, bullet 3
3
softHail
Ref [1], Section A.6, page 17, bullet 3
3.24.3.14 Slow Vehicle Sub Cause
PROTOBUF DEFINITION: slowVehicleSubCause
DATA TYPE: SlowVehicleSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.70, page 52, bullet 1
1
maintenanceVehicle
Ref [1], Section A.70, page 52, bullet 2
2
vehiclesSlowingToLookAtAccident
Ref [1], Section A.70, page 52, bullet 3
3
abnormalLoad
```

```
Ref [1], Section A.70, page 52, bullet 4
4
abnormalWideLoad
Ref [1], Section A.70, page 52, bullet 5
5
convoy
Ref [1], Section A.70, page 52, bullet 6
6
snowplough
Ref [1], Section A.70, page 52, bullet 7
7
deicing
Ref [1], Section A.70, page 52, bullet 8
8
saltingVehicles
Ref [1], Section A.70, page 52, bullet 9
3.24.3.15 Dangerous End Of Queue Sub Cause
PROTOBUF DEFINITION: dangerousEndOfQueueSubCause
DATA TYPE: DangerousEndOfQueueSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.70, page 52, bullet 1
1
suddenEndOfQueue
Ref [1], Section A.70, page 52, bullet 2
2
queueOverHill
Ref [1], Section A.70, page 52, bullet 3
3
queueAroundBend
Ref [1], Section A.70, page 52, bullet 4
4
queueInTunnel
Ref [1], Section A.70, page 52, bullet 5
3.24.3.16 Vehicle Breakdown Sub Cause
PROTOBUF DEFINITION: vehicleBreakdownSubCause
DATA TYPE: VehicleBreakdownSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.90, page 61, bullet 1
1
lackOfFuel
Ref [1], Section A.90, page 61, bullet 2
2
lackOfBatteryPower
Ref [1], Section A.90, page 61, bullet 3
3
engineProblem
Ref [1], Section A.90, page 61, bullet 4
4
transmissionProblem
Ref [1], Section A.90, page 61, bullet 5
5
engineCoolingProblem
Ref [1], Section A.90, page 61, bullet 6
6
brakingSystemProblem
```

```
Ref [1], Section A.90, page 61, bullet 7
7
steeringProblem
Ref [1], Section A.90, page 61, bullet 8
8
tyrePuncture
Ref [1], Section A.90, page 61, bullet 9
3.24.3.17 Post Crash Sub Cause Code
PROTOBUF DEFINITION: postCrashSubCause
DATA TYPE: PostCrashSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.55, page 45, bullet 1
1
accidentWithoutECallTriggered
Ref [1], Section A.55, page 45, bullet 2
2
accidentWithECallManuallyTriggered
Ref [1], Section A.55, page 45, bullet 3
3
accidentWithECallAutomaticallyTriggered
Ref [1], Section A.55, page 45, bullet 4
4
accidentWithECallTriggeredWithoutAccessToCellularNetwork
Ref [1], Section A.55, page 45, bullet 5
3.24.3.18 Human Problem Sub Cause
PROTOBUF DEFINITION: humanProblemSubCause
DATA TYPE: HumanProblemSubCauseCode
MANDATORY/OPTIONAL:  Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.38, page 37, bullet 1
1
glycemiaProblem
Ref [1], Section A.38, page 37, bullet 2
2
heartProblem
Ref [1], Section A.38, page 37, bullet 3
3.24.3.19  Stationary Vehicle Sub Cause
PROTOBUF DEFINITION: stationaryVehicleSubCause
DATA TYPE: StationaryVehicleSubCauseCode
MANDATORY/OPTIONAL: Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.76, page 55, bullet 1
1
humanProblem
Ref [1], Section A.76, page 55, bullet 2
2
vehicleBreakdown
Ref [1], Section A.76, page 55, bullet 3
3
postCrash
Ref [1], Section A.76, page 55, bullet 4
```

```
4
publicTransportStop
Ref [1], Section A.76, page 55, bullet 5
5
carryingDangerousGoods
Ref [1], Section A.76, page 55, bullet 6
3.24.3.20 Emergency Vehicle Approaching Sub Cause
PROTOBUF DEFINITION: emergencyVehicleApproachingSubCause
DATA TYPE: EmergencyVehicleApproachingSubCauseCode
MANDATORY/OPTIONAL: Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.26, page 29, bullet 1
1
emergencyVehicleApproaching
Ref [1], Section A.26, page 29, bullet 2
2
prioritizedVehicleApproaching
Ref [1], Section A.26, page 29, bullet 3
3.24.3.21 Hazardous Location Dangerous Curve Sub Cause
PROTOBUF DEFINITION: hazardousLocation_DangerousCurveSubCause
DATA TYPE: HazardousLocation_DangerousCurveSubCauseCode
MANDATORY/OPTIONAL: Optional

MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.31, page 32, bullet 1
1
dangerousLeftTurnCurve
Ref [1], Section A.31, page 32, bullet 2
2
dangerousRightTurnCurve
Ref [1], Section A.31, page 32, bullet 3
3
multipleCurvesStartingWithUnknownTurningDirection
Ref [1], Section A.31, page 32, bullet 4
4
multipleCurvesStartingWithLeftTurn
Ref [1], Section A.31, page 32, bullet 5
5
multipleCurvesStartingWithRightTurn
Ref [1], Section A.31, page 32, bullet 6
3.24.3.22 Collision Risk Sub Cause
PROTOBUF DEFINITION: collisionRiskSubCause
DATA TYPE: CollisionRiskSubCauseCode
MANDATORY/OPTIONAL: Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.12, page 22, bullet 1
1
longitudinalCollisionRisk
Ref [1], Section A.12, page 22, bullet 2
2
crossingCollisionRisk
Ref [1], Section A.12, page 22, bullet 3
```

```
3
lateralCollisionRisk
Ref [1], Section A.12, page 22, bullet 4
4
vulnerableRoadUser
Ref [1], Section A.12, page 22, bullet 5
3.24.3.23 Signal Violation Sub Cause
PROTOBUF DEFINITION: signalViolationSubCauseCodeSubCause
DATA TYPE: SignalViolationSubCauseCode
MANDATORY/OPTIONAL: Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
Ref [1], Section A.69, page 52, bullet 1
1
stopSignViolation
Ref [1], Section A.69, page 52, bullet 2
2
trafficLightViolation
Ref [1], Section A.69, page 52, bullet 3
3
turningRegulationViolation
Ref [1], Section A.69, page 52, bullet 4
3.24.3.24 Dangerous Situation Sub Cause
PROTOBUF DEFINITION: dangerousSituationSubCause
DATA TYPE: DangerousSituationSubCauseCode
MANDATORY/OPTIONAL: Optional
MESSAGE
Enum Position
Position Type
Description
0
unavailable
 Ref [1], Section A.18, page 26, bullet 1
1
emergencyElectronicBrakeEngaged
Ref [1], Section A.18, page 26, bullet 2
2
preCrashSystemEngaged
Ref [1], Section A.18, page 26, bullet 3
3
espEngaged
Ref [1], Section A.18, page 26, bullet 4
4
absEngaged
Ref [1], Section A.18, page 26, bullet 5
5
aebEngaged
Ref [1], Section A.18, page 26, bullet 6
6
brakeWarningEngaged
Ref [1], Section A.18, page 26, bullet 7
7
collisionRiskWarningEngaged
Ref [1], Section A.18, page 26, bullet 8
3.24.4 Relevance Traffic Direction Enum
PROTOBUF DEFINITION: relevanceTrafficDirection
DATA TYPE: RelevanceTrafficDirectionEnum
MANDATORY/OPTIONAL: Optional
MESSAGE
Enum Position
Position Type
Description
0
```

```
allTrafficDirections
Ref [1], Section A.62, page 48
1
sameTraffic
Traffic affected moving in same direction as reporting vehicle
2
oppositeTraffic
Ref [1], Section A.62, page 48

3.24.5 Relevance Event Reference Enum
PROTOBUF DEFINITION: relevanceEventReference
DATA TYPE: RelevanceEventReferenceEnum
MANDATORY/OPTIONAL: Optional
MESSAGE
Enum Position
Position Type
Description
0
allStreamsTraffic
All streams of Traffic affected
1
upStreamTraffic
Ref [1], Section A.62, page 48
2
downStreamTraffic
Ref [1], Section A.62, page 48

3.24.6 Relevance Distance
PROTOBUF DEFINITION: relevanceDistance
DATA TYPE: RelevanceDistance
MANDATORY/OPTIONAL: Optional
MESSAGE
Enum Position
Position Type
Description
0
lessThan50M
Ref [1], Section A.61, page 47
1
lessThan100M
Ref [1], Section A.61, page 47
2
lessThan200M
Ref [1], Section A.61, page 47
3
lessThan500M
Ref [1], Section A.61, page 47
4
lessThan1000M
Ref [1], Section A.61, page 47
5
lessThan5KM
Ref [1], Section A.61, page 47
6
lessThan10KM
Ref [1], Section A.61, page 47
7
over10KM
Ref [1], Section A.61, page 47
```

Note: [1] ETSI TS 102 894-2 V1.2.1 (2014-09) For information on Specific Observed Event cause and sub causes, see *http://www.etsi.org/deliver/etsi_ts/102800_102899/10289402/01.02.01_60/ts_10289402v010201p.pdf*.

Creative Commons Notice

```
b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your
contributions to Adapted Material in accordance with the terms and conditions
 of this Public License.

c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright
including, without limitation, performance,
 broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are
labeled or categorized. For purposes of this Public License, the
 rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. Effective Technological Measures means those measures that, in the absence of proper authority,
may not be circumvented under laws fulfilling obligations under
 Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international
agreements.

e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or
limitation to Copyright and Similar Rights that applies to Your use of the
 Licensed Material.

f. Licensed Material means the artistic or literary work, database, or other material to which the
Licensor applied this Public License.

g. Licensed Rights means the rights granted to You subject to the terms and conditions of this
Public License, which are limited to all Copyright and Similar Rights
 that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

i. Share means to provide material to the public by any means or process that requires permission
under the Licensed Rights, such as reproduction, public display,
 public performance, distribution, dissemination, communication, or importation, and to make
material available to the public including in ways that members of the
 public may access the material from a place and at a time individually chosen by them.

j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC
of the European Parliament and of the Council of 11 March 1996 on the
 legal protection of databases, as amended and/or succeeded, as well as other essentially
equivalent rights anywhere in the world.

k. You means the individual or entity exercising the Licensed Rights under this Public License.
Your has a corresponding meaning.

Section 2 – Scope.

a. License grant.

 1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a
worldwide, royalty-free, non-sublicensable, non-exclusive,
  irrevocable license to exercise the Licensed Rights in the Licensed Material to:

   A. reproduce and Share the Licensed Material, in whole or in part; and

   B. produce, reproduce, and Share Adapted Material.

 2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply
to Your use, this Public License does not apply, and You do
  not need to comply with its terms and conditions.

 3. Term. The term of this Public License is specified in Section 6(a).

 4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the
Licensed Rights in all media and formats whether now known
  or hereafter created, and to make technical modifications necessary to do so. The Licensor waives
and/or agrees not to assert any right or authority to forbid
  You from making technical modifications necessary to exercise the Licensed Rights, including
technical modifications necessary to circumvent Effective
```

```
   Technological Measures. For purposes of this Public License, simply making modifications
 authorized by this Section 2(a)(4) never produces Adapted Material.

  5. Downstream recipients.
    A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material
 automatically receives an offer from the Licensor to exercise the
     Licensed Rights under the terms and conditions of this Public License.

    B. No downstream restrictions. You may not offer or impose any additional or different terms or
 conditions on, or apply any Effective Technological Measures
     to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any
 recipient of the Licensed Material.

  6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to
 assert or imply that You are, or that Your use of the Licensed
  Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor
 or others designated to receive attribution as provided in
  Section 3(a)(1)(A)(i).

 b. Other rights.

  1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor
 are publicity, privacy, and/or other similar personality rights;
   however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights
 held by the Licensor to the limited extent necessary to allow
   You to exercise the Licensed Rights, but not otherwise.

  2. Patent and trademark rights are not licensed under this Public License.

  3. To the extent possible, the Licensor waives any right to collect royalties from You for the
 exercise of the Licensed Rights, whether directly or through a
   collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In
 all other cases the Licensor expressly reserves any right to
   collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

 a. Attribution.

  1. If You Share the Licensed Material (including in modified form), You must:

   A. retain the following if it is supplied by the Licensor with the Licensed Material:
    i. identification of the creator(s) of the Licensed Material and any others designated to
 receive attribution, in any reasonable manner requested by the
     Licensor (including by pseudonym if designated);
    ii. a copyright notice;
    iii.a notice that refers to this Public License;
    iv. a notice that refers to the disclaimer of warranties;
    v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
   B. indicate if You modified the Licensed Material and retain an indication of any previous
 modifications; and
   C. indicate the Licensed Material is licensed under this Public License, and include the text of,
 or the URI or hyperlink to, this Public License.

  2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium,
 means, and context in which You Share the Licensed Material.
   For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a
 resource that includes the required information.

  3. If requested by the Licensor, You must remove any of the information required by Section 3(a)
(1)(A) to the extent reasonably practicable.

  4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent
 recipients of the Adapted Material from complying with this
   Public License.
```

```
Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed
 Material:

 a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce,
 and Share all or a substantial portion of the contents of the
  database;

 b. if You include all or a substantial portion of the database contents in a database in which You
 have Sui Generis Database Rights, then the database in which You
  have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

 c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of
 the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under
 this Public License where the Licensed Rights include other Copyright
and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

 a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor
 offers the Licensed Material as-is and as-available, and makes no
  representations or warranties of any kind concerning the Licensed Material, whether express,
 implied, statutory, or other. This includes, without limitation,
  warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence
 of latent or other defects, accuracy, or the presence or absence
  of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed
 in full or in part, this disclaimer may not apply to You.

 b. To the extent possible, in no event will the Licensor be liable to You on any legal theory
 (including, without limitation, negligence) or otherwise for any direct,
  special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs,
 expenses, or damages arising out of this Public License or use of the
  Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs,
 expenses, or damages. Where a limitation of liability is not
  allowed in full or in part, this limitation may not apply to You.

 c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in
 a manner that, to the extent possible, most closely approximates an
  absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

 a. This Public License applies for the term of the Copyright and Similar Rights licensed here.
 However, if You fail to comply with this Public License, then Your rights
  under this Public License terminate automatically.

 b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

  1. automatically as of the date the violation is cured, provided it is cured within 30 days of
 Your discovery of the violation; or

  2. upon express reinstatement by the Licensor.

 For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to
 seek remedies for Your violations of this Public License.

 c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate
 terms or conditions or stop distributing the Licensed Material at any
  time; however, doing so will not terminate this Public License.

 d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.
```

```
 a. The Licensor shall not be bound by any additional or different terms or conditions communicated
 by You unless expressly agreed.

 b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated
 herein are separate from and independent of the terms and conditions of
  this Public License.

Section 8 – Interpretation.

 a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to,
 reduce, limit, restrict, or impose conditions on any use of the
  Licensed Material that could lawfully be made without permission under this Public License.

 b. To the extent possible, if any provision of this Public License is deemed unenforceable, it
 shall be automatically reformed to the minimum extent necessary to make
  it enforceable. If the provision cannot be reformed, it shall be severed from this Public License
 without affecting the enforceability of the remaining terms and
  conditions.

 c. No term or condition of this Public License will be waived and no failure to comply consented to
 unless expressly agreed to by the Licensor.

 d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver
 of, any privileges and immunities that apply to the Licensor or You,
  including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect
 to apply one of its public licenses to material it publishes and in those
instances will be considered the “Licensor.” The text of the Creative Commons public licenses is
 dedicated to the public domain under the CC0 Public Domain Dedication. Except
for the limited purpose of indicating that material is shared under a Creative Commons public
 license or as otherwise permitted by the Creative Commons policies published at
creativecommons.org/policies, Creative Commons does not authorize the use of the trademark “Creative
 Commons” or any other trademark or logo of Creative Commons without its
prior written consent including, without limitation, in connection with any unauthorized
 modifications to any of its public licenses or any other arrangements, understandings,
or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does
 not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.
```