

# NYC CVPD Event Format Specification

## Version 2.0 DRAFT

Oct 30, 2019

Updated: October 05, 2020

Document Number

TRANSCORE
Trusted Transportation Solutions

| REVISION HISTORY | | | | | |
|---|---|---|---|---|---|
| **DATE** | **REV.** | **BY** | **SPCR** | **SECTION** | **DESCRIPTION** |
| 5/7/2018 | 1.0 | Keith Patton | | | Release |
| 11/1/2018 | 1.1 | Keith Patton | | | Added message NycCvpdAsdMobility and data frame VehLocRecord |
| 01/30/2019 | 1.2 | Rojer Babu | | | Added new Parameters to DF_ParameterSet; Minor Case Sensitive changes in alignment with J2735. |
| 2/6/2019 | 1.3 | David Benevelli | | | Added pedSig to EventType, EventAlertActive values defined, modified event format specification to eliminate MIB reference. |
| 2/7/2019 | 1.4 | Rojer Babu | | | Minor Case Sensitive changes and punctuation corrections. |
| 2/20/2019 | 1.5 | Rojer Babu | | | Added Serial Number to NycCvpdASDMobility. |
| 3/12/2019 | 1.6 | Rojer Babu | | | Adding Elevation to VehLocRecord |
| 4/30/2019 | 1.7 | Rojer Babu | | | Adding LocationSource Field |

| REVISION HISTORY | | | | | |
|---|---|---|---|---|---|
| **DATE** | **REV.** | **BY** | **SPCR** | **SECTION** | **DESCRIPTION** |
| 10/10/2019 | 1.8 | David Benevelli | | | DE_RFLevel definition updated to reflect ASD output of 0.5 dBm steps.<br>Added:<br>- DE_GroupId<br>- grpId (DE_GroupId) to DF_EventHeader<br>Added vehicle location to define location of vehicle when first and last messages received for:<br>- MAPRecordRf<br>- SPaTRecordRf<br>- TIMRecordRf<br>- Used HV BSM rather than VehLocRecord. |
| 10/14/2019 | 1.9 | Rojer Babu | | | - DE_RFLevel range changed to -300 to 301 to reflect real world receive signal strengths.<br>- Changed all DFullTime definitions to DDateTime definition to be able to record events with milliseconds time accuracy. |

| REVISION HISTORY | | | | | |
|---|---|---|---|---|---|
| **DATE** | **REV.** | **BY** | **SPCR** | **SECTION** | **DESCRIPTION** |
| 10/05/2020 | 2.0 | Rojer Babu | | | - Modified LocationSource ENUMs<br>- Added new parameters to ParameterSet |

# Table of Contents

The following data dictionary specification is provided for the transmittal of Event Data from the ASD to the TMC. Data elements and frames not defined in this document can be referenced from the SAE International's J2735-201603 Dedicated Short Range Communications (DSRC) Message Set Dictionary.

# 1 Messages

## 1.1 MSG_NycCvpdEvent

This message collects the information for each event and puts it in a single structure. Sizes are not specified for the various lists as the maximum number of messages for an event cannot be reliability calculated.

ASN.1 Representation:

```
NycCvpdEvent ::= SEQUENCE {
        eventHeader     EventHeader,
        bsmList         SEQUENCE OF BSMEvent,
        mapList          SEQUENCE OF MAPEvent OPTIONAL,
        spatList         SEQUENCE OF SPaTEvent OPTIONAL,
        timList         SEQUENCE OF TIMEvent OPTIONAL,
        …
}
```

## 1.2 MSG_NycCvpdRSURF

This message collects the BSM messages used to evaluate the RF distribution around an RSU. The TMC will pull this message back on a frequent basis to keep the effective size small.

```
NycCvpdRSURF ::= SEQUENCE OF BSMRecordRSURF
```

## 1.3 MSG_NycCvpdASDRF

This message collects the BSM, MAP, SPaT and TIM messages to evaluate the RF distribution around an ASD.

```
NycCvpdASDRF::= SEQUENCE {
            asdSerialNumber     VisibleString,
            bsmRFList           SEQUENCE OF BSMRecordASDRF OPTIONAL,
            mapRFList           SEQUENCE OF MAPRecordRF OPTIONAL,
            spatRFList          SEQUENCE OF SPaTRecordRF OPTIONAL,
            timRFList           SEQUENCE OF TIMRecordRF OPTIONAL
            }
```

## 1.4 MSG_NycCvpdTravelTime

This message reports the BSMs for calculating travel time at the TMC. These will be the BSMs recorded when the vehicle is in the center of the intersection. This data is collected by the RSU.

```
NycCvpdTravelTime ::= SEQUENCE {
        bsmList                 SEQUENCE OF BSMRecordTravelTime,
        …
}
```

### 1.5 MSG_NycCvpdASDMobility

This message reports the BSMs collected by the ASD to track the overall movement of a vehicle through the system to evaluate overall mobility. This message contains information to identify the vehicle and therefore must be encrypted on the ASD for transmittal to the RSU for upload to the TMC.

```
NycCvpdASDMobility ::= SEQUENCE {
        vehID           VINstring, -- As specified in J2735 paragraph 7.225
        asdSerialNumber         VisibleString,
        locList         SEQUENCE OF VehLocRecord,
        timeRecordResolution    TimeRecordResolution,
        …
}
```

## 2 Data Frames

### 2.1 DF_BSMRecord

The BSMRecord data frame adds a message header to each BSM message recorded by either the ASD or RSU.

```
BSMRecord ::=SEQUENCE {
        msgHeader               MsgHeader, bsmMsg
                                BasicSafetyMessage
}
```

### 2.2 DF_BSMEvent

The BSMEvent data frame adds the EventMsgSeqNum data element to all of the BSMRecord data frames that are part of the event.

```
BSMEvent ::= SEQUENCE  {
        seqNum      EventMsgSeqNum,
        bsmRecord               BSMRecord
}
```

### 2.3 DF_BSMRecordRSURF

This data frame collects the RF data for remote vehicles as seen from an RSU. A change in the tempID of the remote vehicle will trigger the previous BSM recording as the last vehicle sighting and the current BSM as the first sighting of the new vehicle.

```
BSMRecordRSURF::= SEQUENCE {
        firstBSMTime        DDateTime,
        firstBSM            BSMRecord,
        lastBSMTime         DDateTime,
        lastBSM             BSMRecord
        }
```

## 2.4 DF_BSMRecordASDRF

This data frame collects the RF data for remote vehicles as seen from an ASD. The host vehicle BSM must also be recorded to gather information about the location of the host vehicle with respect to the remote vehicle. A change in the tempID of the remote vehicle will trigger the previous BSM recording as the last vehicle sighting and the current BSM as the first sighting of the new vehicle.

```
BSMRecordASDRF::= SEQUENCE {
                firstBSMTime            DDateTime,
                firstHVBSM              BSMRecord,
                firstRVBSM              BSMRecord,
                lastBSMTime             DDateTime,
                lastHVBSM               BSMRecord,
                lastRVBSM               BSMRecord
}
```

## 2.5 DF_BSMRecordTravelTime

The BSMRecordTravelTime is used to collect the BSM that will be used for the travel time calculations at central. Each record will contain the standard message header along with the detection geo zone and the BSM itself.

```
BSMRecordTravelTime ::= SEQUENCE {
                msgHeader               MsgHeader,
                geoZone                 GeoZone,
                bsmMsg                  BasicSafetyMessage
}
```

## 2.6 DF_EventHeader

This data frame captures the meta information about the event. If the event is an internal event (ex. red light violation warning or a speed warning) the hostVehId and the targetVehId will be the same. The triggerHvSeqNum and triggerTvSeqNum will also be the same.

```
EventHeader ::= SEQUENCE {
        eventTimeStamp          DDateTime,
        locationSource          LocationSource,
        asdSerialNumber         VisibleString,
        asdFirmwareVersion      VisibleString,
        eventAlertActive        EventAlertActive,
        eventAlertSent          EventAlertSent,
        eventAlertHeard         EventAlertHeard,
        hostVehID               TemporaryID,
        targetVehID             TemporaryID,
```

triggerHVSeqNum EventMsgSeqNum,
--The sequence number of the HV BSM that triggered the event

```
        triggerTVSeqNum         EventMsgSeqNum,
        --The sequence number of the TV BSM that triggered the event
        eventType               EventType,
        parameters              ParameterSet,
        grpId                   GroupId,
        …
}
```

## 2.7 DF_MAPRecord

This data frame contains a list of Map messages for the event

```
MAPRecord::=SEQUENCE {
        msgHeader       MsgHeader,
        mapMsg          MapData
}
```

## 2.8 DF_MAPEvent

This data frame contains a list of Map messages for the event

```
MAPEvent::=SEQUENCE {
        seqNum          EventMsgSeqNum,
        mapRecAct       MAPRecord
}
```

## 2.9 DF_MAPRecordRF

This data frame collects the information on the received MAP messages to evaluate the RF effectiveness.

```
MAPRecordRF::= SEQUENCE {
            firstMAPTime            DDateTime,
            firstMAPRecord          MAPRecord,
            lastMAPTime             DDateTime,
            lastMAPRecord           MAPRecord,
            firstMAPLocation        BasicSafetyMessage,
            lastMAPLocation         BasicSafetyMessage
}
```

## 2.10 DF_MsgHeader

This data frame is attached to each message (BSM, SPaT, TIM, MAP) by the receiving device (ASD or RSU).

```
MsgHeader ::= SEQUENCE { myRFLevel
                        RFLevel,
        authenticated   MsgAuthenticated
}
```

## 2.11 DF_ParameterSet

This data frame captures the parameters in force in the ASD at the time of the event.

```
ParameterSet ::=SEQUENCE {
        recordingROI            NumericString OPTIONAL,
        -- the number of seconds before the event
        timeRecordBefore        NumericString,
        -- the number of seconds after the event
        timeRecordFollow        NumericString,
        timeRecordResolution    TimeRecordResolution,
        -- for this specific type of event minSpdThreshold
                                NumericString,
        timeToCrash             NumericString OPTIONAL,
        excessiveCurveSpd       NumericString OPTIONAL,
        excessiveSpd            NumericString OPTIONAL,
        excessiveSpdTime        NumericString OPTIONAL,
        excessiveCurveSpdTime   NumericString OPTIONAL,
        excessiveZoneSpd        NumericString OPTIONAL,
        excessiveZoneSpdTime    NumericString OPTIONAL,
        minCurveSpd             NumericString OPTIONAL,
        minZoneSpd              NumericString OPTIONAL,
        stopBarTolerance        NumericString OPTIONAL,
        yellowDurationTolerance NumericString OPTIONAL,
        hardBrakingThreshold    NumericString OPTIONAL,
        assumedDriverBraking    NumericString OPTIONAL,
        postedHeightLimit       NumericString OPTIONAL,
        postedSizeLimit         NumericString OPTIONAL,
        postedZoneSpeed         NumericString OPTIONAL,
        regulatorySpeed         NumericString OPTIONAL,
        …
}
```

## 2.12 DF_SPaTRecord

This data frame assembles the additional information required to record SPaT messages.

```
SPaTRecord::= SEQUENCE {
        msgHeader   MsgHeader,
        spatMsg     SPAT

}
```

## 2.13 DF_SPaTRecordRF

This data frame collects the SPaT data for evaluation of RF effectiveness.

```
SPaTRecordRF ::= SEQUENCE {
        firstSPaTTime           DDateTime,
```

firstSPaTRecord SPaTRecord,

```
        lastSPaTTime            DDateTime,
        lastSPaTRecord
                                SPaTRecord
        ,
        firstSPaTLocation       BasicSafetyMessage,
        lastSPaTLocation        BasicSafetyMessage
}
```

## 2.14 DF_SPaTEvent

This data frame adds the message sequence number for the SPaT message within an event.

```
SPaTEvent::= SEQUENCE {
        seqNum                  EventMsgSeqNum,
         spatRecord             SPaTRecord
}
```

## 2.15 DF_TIMRecord

This data Frame adds the additional information to a TIM for recording and evaluation.

```
TIMRecord::= SEQUENCE {
        msgHeader   MsgHeader,
        timMsg      TravelerInformation
}
```

## 2.16 DF_TIMRecordRF

This data frame collects the information on the received TIM messages to evaluate the RF effectiveness.

```
TIMRecordRF::= SEQUENCE {
        firstTIMTime            DDateTime,
        firstTIMRecord          TIMRecord,
        lastTIMTime             DDateTime,
        lastTIMRecord           TIMRecord,
        firstTIMLocation        BasicSafetyMessage,
        lastTIMLocation         BasicSafetyMessage
}
```

## 2.17 DF_TIMEvent

This data frame captures additional information for TIM messages received during an event.

```
TIMEvent::=SEQUENCE {
        seqNum                  EventMsgSeqNum,
        timRecord   TIMRecord
}
```

### 2.18 DF_ VehLocRecord

This data frame captures the vehicle location at a specific time. Since this is providing data to evaluate overall mobility, time resolution greater than 1 second is not required.

```
VehLocRecord::= SEQUENCE {
        timeStamp                DDateTime
        , longitude              Longitude,
        latitude                 Latitude,
        elevation                Elevation
}
```

# 3 Data Elements

## 3.1 DE_EventAlertActive

This data element records the control state of audible notifications – they are either Silent (FALSE) or Active (TRUE) mode, with the default being the Active mode. If this data element is TRUE, then audible notifications are active in the ASD.

EventAlertActive::=BOOLEAN

## 3.2 DE_EventAlertHeard

This data element flags if the alert generated by this event was heard/detected. If this event did not send an alert, indicated when EventAlertSent is FALSE, then this data element must be FALSE.

EventAlertHeard::=BOOLEAN

## 3.3 DE_EventAlertSent

This data element flags the event as an event generating an alert. In the case of multiple events occurring simultaneously the performance analysis needs to know which event generated an alert. If this event sent an alert the value will be TRUE. A higher priority event may occur and result in this event's alert being suppressed in which case this event's value will be FALSE.

EventAlertSent::=BOOLEAN

## 3.4 DE_EventMsgSeqNum

This data element captures the order in which messages are received or stored for the event the oldest message (BSM, SPaT, TIM or MAP) will be 1. The first message in each event will start at 1.

EventMsgSeqNum ::=INTEGER (1..16384)

## 3.5 DE_EventType

This data element reports the event type.

```
EventType::=ENUMERATED {
        unknown (0),
```

```
        fcw (1),
        eebl (2),
        bsw (3),
        lcw (4),
        ima (5),
        vtrw (6),
        spdcomp (7),
        cspdcomp (8),
        spdcompwz (9),
        rlvw (10),
        ovcturnprohibit (11),
        ovcclearancelimit (12),
        evacinfo (13),
        pedinxwalk (14),
        pedSig (15)

}
```

## 3.6 DE_GeoZone

This data element is part of the BsmRecord and records the geozone for BSMs recorded to support the central travel time calculations. A value of 0 for GeoZone indicates that the BSM was not in a defined detection geo zone.

NOTE: The range is set at 0 to 128. We do not expect more than 16 zones per RSU.

```
GeoZone::= INTEGER(0..128)
```

## 3.7 DE_RFLevel

This data element captures the RF level of the received message. The measurement shall be in increments of 0.5 dBm. Internal messages will have a reading of 301.

```
RFLevel ::= INTEGER (-300..301)
```

## 3.8 DE_TimeRecordResolution

This is a data element that specifies the rate at which event data is logged.

```
TimeRecordResolution ::= ENUMERATED {
            resOneTenthOfSec, -- every 1/10 second
            resTwoTenthOfSec, -- every 2/10 second
            resFiveTenthsOfSec, -- every 5/10 second
            resEverySec, -- every 1 second
            resEveryTwoSec, -- every 2 seconds
            resEveryFiveSec, -- every 5 seconds
            resEveryTenSec  -- every 10 seconds
}
```

## 3.9 DE_MsgAuthenticated

This data element is a Boolean value to indicate if the message was authenticated by the ASD.

MsgAuthenticated ::= BOOLEAN

## 3.10 DE_LocationSource

This data element defines the data fusion process used to enhance location accuracy.

```
LocationSource ::= ENUMERATED {
        unknown (0),
        directgps (1),
        gps (2),
        gpsrsu (3)
}
```

## 3.11 DE_GroupId

This data element contains the assigned group identification number for managing the OTA updates. See the list of OTA Groups.

```
GroupId ::= INTEGER (0..255)
```