| Document Title | Specification of Chinese Vehicle-2-X Message |
|---|---|
| **Document Owner** | AUTOSAR |
| **Document Responsibility** | AUTOSAR |
| **Document Identification No** | 990 |

| | |
|---|---|
| **Document Status** | published |
| **Part of AUTOSAR Standard** | Classic Platform |
| **Part of Standard Release** | R22-11 |

| Document Change History | | | |
|---|---|---|---|
| **Date** | **Release** | **Changed by** | **Description** |
| 2022-11-24 | R22-11 | AUTOSAR Release Management | • Initial release |

**Disclaimer**

# Contents

# 1 Introduction and functional overview

This document specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Chinese Vehicle-2-X Message (CnV2xMsg).
The Chinese Vehicle-2-X Message together with the Chinese Vehicle-2-X Network (CnV2xNet), Chinese Vehicle-2-X Management (CnV2xM), Chinese Vehicle-2-X Security (CnV2xSec), Vehicle-2-X Data Manager (V2xDm) and the communication driver layer forms the Chinese V2X stack within the AUTOSAR architecture.
The CnV2xMsg module is designed to be hardware independent. The CnV2xMsg module is dependent on services of Chinese V2X entities in the application layer and on lower CnV2xNet module, and provides services to the V2xDm module.

## 1.1 Architecture Overview

Positioning of the CnV2xMsg module within the AUTOSAR BSW and the Layered Software architecture is shown in Figure 1.

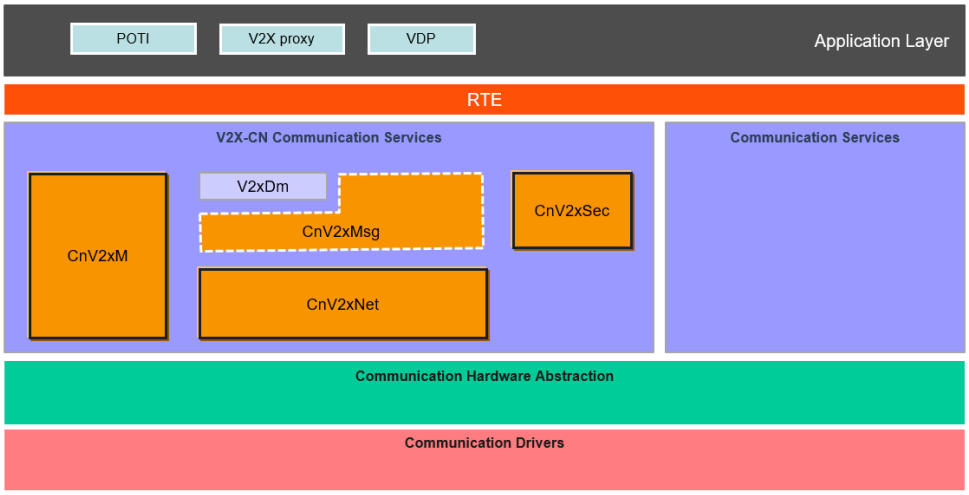


**Figure 1.1: AUTOSAR BSW software architecture - CnV2xMsg scope**

The CnV2xMsg module provides basic services of Basic Safety Message (BSM) and supports related management functions for BSM exchange.

## 1.2 Functional Overview

The CnV2xMsg module implements the basic service of BSM sending and receiving, and RSI/RSM/SPAT/MAP receiving. Besides that, management functions including Frequency Management, POTI management and ID management related to BSM sending are also implemented in current CnV2xMsg module.

### 1.2.1 Basic Safety Message (BSM)

The BSM basic service is a message layer entity that operates the BSM protocol. It provides two services: sending and receiving of BSMs. The BSM basic service generates and sends BSMs to other Vehicles/RSUs or it receives BSMs from Vehicles and provides them to the applications. It may interface with the AUTOSAR application layer in order to collect relevant information for BSM generation. The BSM basic service uses the services provided by the protocol entities of the lower layers of the Chinese V2X stack to disseminate the BSM. Upon receiving a BSM, the BSM basic service makes the content of the BSM available to the V2X applications. Received BSMs can be given to the upper application layer via their standardized AUTOSAR service interface CnV2xApplRxIndicationBsm or via V2xDm.

For sending and receiving BSMs, the BSM basic service part of the CnV2xMsg shall provide the following sub-functions:

- Encode BSM
- Decode BSM
- BSM transmission management
- BSM reception management

For details see [1] chapter 6.

### 1.2.2 Road Side Information (RSI)

The RSI service is a message layer entity that provides receiving of RSI messages. The RSI service receives RSIs from RSU and provides them to applications. Received RSIs can be given to the upper application layer via standardized AUTOSAR service interface CnV2xApplRxIndicationRsi or via V2xDm.

### 1.2.3 Road Side Message (RSM)

The RSM service is a Message layer entity that provides receiving of RSM messages. The RSM service receives RSMs from RSU and provides them to V2X applications. Received RSMs can be given to the upper application layer via standardized AUTOSAR service interface CnV2xApplRxIndicationRsm or via V2xDm.

### 1.2.4 Signal Phase and Time (SPAT)

The SPAT service is a Message layer entity that provides receiving of SPAT messages. The SPAT service receives SPATs from RSU and provides them to V2X appli-

cations. Received SPATs can be given to the upper application layer via standardized AUTOSAR service interface CnV2xApplRxIndicationSpat or via V2xDm.

### 1.2.5 MAP

The MAP service is a Message layer entity that provides receiving of MAP messages. The MAP service receives MAPs from RSU and provides them to V2X applications. Received MAPs can be given to the upper application layer via standardized AUTOSAR service interface CnV2xApplRxIndicationMap or via V2xDm.

### 1.2.6 Position and Time Management(POTI)

POTI management in CnV2xMsg module gets position and time information from application layer and makes is available to itself, and also provides distances to CnV2xSec module.

### 1.2.7 Identity Management

CnV2xMsg shall implement of identity management including Vehicle ID and Message Count. From security and privacy perspective, these identities shall be changed when pseudonym certificate updated.

### 1.2.8 Frequency Management

CnV2xMsg shall control message sending frequency to lower layers according to channel state, vehicle state, Message Type, etc.

### 1.2.9 Messages Reception Service Via V2xDm

If the received V2X messages are sent to application layer or PDUR via V2xDm module, the CnV2xMsg shall provides interface to V2xDm module. Upon receiving a message (BSM/RSI/RSM/SPAT/MAP), the CnV2xMsg makes the content of the message available to the V2xDm module. The received messages are given to the upper application layer by the V2xDm module via the standardized AUTOSAR service interface.

# 2 Acronyms and Abbreviations

| Abbreviation / Acronym: | Description: |
|---|---|
| API | Application programming Interface |
| BS | Basic Service |
| BSW | Basic Software |
| BSM | Basic safety Message |
| C-V2X | Cellular based Vehicle to Everything |
| CCSA | China Communications Standards Association |
| CnV2xMsg | Chinese Vehicle-2-X Message |
| CnV2xNet | Chinese Vehicle-2-X Network |
| CnV2xSec | Chinese Vehicle-2-X Security |
| DE | Data Element |
| DEM | Diagnostic Event Manager |
| DET | Default Error Tracer |
| DF | Data Frame |
| EcuM | Electronic Control Unit Manager |
| IF | Interface |
| NTCAS | National Technical Committee of Auto Standardization |
| NVM | Non-Volatile Memory |
| PH | Path History |
| POTI | Position and Time |
| RSI | Road Side Information |
| RSM | Road Side Message |
| RSU | Roadside Unit |
| SPAT | Signal Phase And Time |
| VDP | Vehicle Data provider |

# 3 Related documentation

## 3.1 Input documents & related standards and norms

[1] GB/T: Technical requirements and test methods of vehicular communication system based on LTE-V2X direct communication (Draft Edition: 2022-04-01)
http://www.catarc.org.cn/

[2] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral

[3] Specification of Default Error Tracer
AUTOSAR_SWS_DefaultErrorTracer

[4] Specification of ECU State Manager
AUTOSAR_SWS_ECUStateManager

[5] Specification of Chinese Vehicle-2-X Network
AUTOSAR_SWS_ChineseV2XNetwork

[6] Requirements on Chinese Vehicle-2-X Communication
AUTOSAR_SRS_ChineseV2XCommunication

[7] YD/T 3709-2020:Technical requirements of Message layer of LTE-based vehicular communication
http://www.ccsa.org.cn/

[8] Specification of Vehicle-2-X Facilities
AUTOSAR_SWS_V2XFacilities

## 3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [2, SWS BSW General], which is also valid for CnV2xMsg.

Thus, the specification SWS BSW General shall be considered as additional and required specification for CnV2xMsg.

# 4 Constraints and assumptions

## 4.1 Limitations

The Chinese V2X modules follow the technical requirements regarding the Day-1 scenarios defined by CCSA and NTCAS. Data types of RSI, RSM, SPAT and MAP messages, which are used in service interfaces, are also planed to develop in future release.
The current version does not yet support Messages Reception Service Via V2xDm because V2xDm is not currently available. This function will be supported in subsequent releases.

## 4.2 Applicability to car domains

This specification is applicable to all car domains.

# 5 Dependencies to other modules

## 5.1 AUTOSAR Default Error Tracer (DET)

In development mode, CnV2xMsg module reports errors through the Det_ReportError function of DET Module [3].

## 5.2 AUTOSAR Ecu State Manager (EcuM)

The EcuM [4] initializes the CnV2xMsg module by calling CnV2xMsg_Init specified in 8.3.1 in this document.

## 5.3 V2X Vehicle Data Provider

The CnV2xMsg module retrieves vehicle relevant data from the VDP application by using the Sender-Receiver-Interface CnV2xMsgVdp (see [CP_SWS_CnV2xMsg_01101]).

## 5.4 V2X Proxy

The V2x Proxy is an Application that listens to every BSM via the Sender-Receiver-Interface CnV2xApplRxIndicationBsm (See [CP_SWS_CnV2xMsg_01103]) and transmits it to one or more ECU's via in-vehicle networks.

The CnV2xMsg module delivers received RSI data to the V2x Proxy by using the Sender-Receiver-Interface CnV2xApplRxIndicationRsi (see [CP_SWS_CnV2xMsg_01105]).

The CnV2xMsg module delivers received RSM data to the V2x Proxy by using the Sender-Receiver-Interface CnV2xApplRxIndicationRsm (see [CP_SWS_CnV2xMsg_01107]).

The CnV2xMsg module delivers received SPAT data to the V2x Proxy by using the Sender-Receiver-Interface CnV2xApplRxIndicationSpat (see [CP_SWS_CnV2xMsg_01109]).

The CnV2xMsg module delivers received MAP data to the V2x Proxy by using the Sender-Receiver-Interface CnV2xApplRxIndicationMap (see [CP_SWS_CnV2xMsg_01111]).

## 5.5 AUTOSAR CnV2xNet

The CnV2xMsg module assumes a transmit request primitive (CnV2xNet_Transmit [5], see CnV2xSec_ReqEncap, CnV2xSec_ReqDecap, and CnV2xSec_VehicleEventFlagsIndication, [CP_SWS_CnV2xMsg_01049]) to be provided by the CnV2xNet.

## 5.6 AUTOSAR CnV2xSec

Security mechanisms are configured by the CnV2xSec and are used by CnV2xMsg.The CnV2xMsg module assumes a request primitive (see [CP_SWS_CnV2xMsg_01049]) to be provided by the CnV2xSec module.

## 5.7 AUTOSAR V2xDm

If the received V2X messages are sent to application layer or PDUR via V2xDm module, the CnV2xMsg module shall delivers the received messages to the V2xDm module. The CnV2xMsg module assumes a request primitive to be provided by the Vehicle-2-X Data Manager (V2xDm) module.

# 6 Requirements Tracing

The following tables reference the requirements specified in [6] and links to the fulfillment of these.

| Requirement | Description | Satisfied by |
|---|---|---|
| **[CP_SRS_CnV2X_-00100]** | The implementation of Chinese V2X communication shall follow technical requirements given by CCSA and NTCAS | [CP_SWS_CnV2xMsg_00105] [CP_SWS_CnV2xMsg_00106] [CP_SWS_CnV2xMsg_00107] [CP_SWS_CnV2xMsg_00108] [CP_SWS_CnV2xMsg_00109] [CP_SWS_CnV2xMsg_00110] [CP_SWS_CnV2xMsg_00111] [CP_SWS_CnV2xMsg_00201] [CP_SWS_CnV2xMsg_00403] [CP_SWS_CnV2xMsg_00405] [CP_SWS_CnV2xMsg_00406] [SWS_CnV2xMsg_00202] |
| **[CP_SRS_CnV2X_-00201]** | The Chinese V2X communication shall use UTC time as the reference clock | [CP_SWS_CnV2xMsg_00404] |
| **[CP_SRS_CnV2X_-00203]** | The Chinese V2X communication shall use GCJ-02 coordinate system as the reference coordinate | [CP_SWS_CnV2xMsg_00401] [CP_SWS_CnV2xMsg_00402] |
| **[CP_SRS_CnV2X_-00501]** | BSM basic service of Chinese V2X message layer shall be compliant to CCSA Specification of Message layer of LTE-based vehicular communication | [CP_SWS_CnV2xMsg_00100] [CP_SWS_CnV2xMsg_00204] [CP_SWS_CnV2xMsg_01002] [CP_SWS_CnV2xMsg_01003] [CP_SWS_CnV2xMsg_01004] [CP_SWS_CnV2xMsg_01009] [CP_SWS_CnV2xMsg_01012] [CP_SWS_CnV2xMsg_01014] [CP_SWS_CnV2xMsg_01018] [CP_SWS_CnV2xMsg_01024] [CP_SWS_CnV2xMsg_01026] [CP_SWS_CnV2xMsg_01030] [CP_SWS_CnV2xMsg_01033] [CP_SWS_CnV2xMsg_01036] [CP_SWS_CnV2xMsg_01038] [CP_SWS_CnV2xMsg_01041] [CP_SWS_CnV2xMsg_01043] [CP_SWS_CnV2xMsg_01045] [CP_SWS_CnV2xMsg_01047] [CP_SWS_CnV2xMsg_01049] [CP_SWS_CnV2xMsg_01050] [CP_SWS_CnV2xMsg_01056] [CP_SWS_CnV2xMsg_01061] [CP_SWS_CnV2xMsg_01102] |

| Requirement | Description | Satisfied by |
|---|---|---|
| | | [CP_SWS_CnV2xMsg_01104]<br>[CP_SWS_CnV2xMsg_01106]<br>[CP_SWS_CnV2xMsg_01108]<br>[CP_SWS_CnV2xMsg_01110]<br>[CP_SWS_CnV2xMsg_01112]<br>[CP_SWS_CnV2xMsg_01201]<br>[CP_SWS_CnV2xMsg_02001]<br>[CP_SWS_CnV2xMsg_02002]<br>[CP_SWS_CnV2xMsg_02003]<br>[CP_SWS_CnV2xMsg_02004]<br>[CP_SWS_CnV2xMsg_02005]<br>[CP_SWS_CnV2xMsg_02006]<br>[CP_SWS_CnV2xMsg_02007]<br>[CP_SWS_CnV2xMsg_02008]<br>[CP_SWS_CnV2xMsg_02009]<br>[CP_SWS_CnV2xMsg_02010]<br>[CP_SWS_CnV2xMsg_02011]<br>[CP_SWS_CnV2xMsg_02012]<br>[CP_SWS_CnV2xMsg_02013]<br>[CP_SWS_CnV2xMsg_02014]<br>[CP_SWS_CnV2xMsg_02015]<br>[CP_SWS_CnV2xMsg_02016]<br>[CP_SWS_CnV2xMsg_02017]<br>[CP_SWS_CnV2xMsg_02018]<br>[CP_SWS_CnV2xMsg_02019]<br>[CP_SWS_CnV2xMsg_02020]<br>[CP_SWS_CnV2xMsg_02021]<br>[CP_SWS_CnV2xMsg_02022]<br>[CP_SWS_CnV2xMsg_02023]<br>[CP_SWS_CnV2xMsg_02024]<br>[CP_SWS_CnV2xMsg_02025]<br>[CP_SWS_CnV2xMsg_02026]<br>[CP_SWS_CnV2xMsg_02027]<br>[CP_SWS_CnV2xMsg_02028]<br>[CP_SWS_CnV2xMsg_02029]<br>[CP_SWS_CnV2xMsg_02030]<br>[CP_SWS_CnV2xMsg_02032]<br>[CP_SWS_CnV2xMsg_02033]<br>[CP_SWS_CnV2xMsg_02034]<br>[CP_SWS_CnV2xMsg_02035]<br>[CP_SWS_CnV2xMsg_02036]<br>[CP_SWS_CnV2xMsg_02037]<br>[CP_SWS_CnV2xMsg_02038]<br>[CP_SWS_CnV2xMsg_02101]<br>[CP_SWS_CnV2xMsg_02102]<br>[CP_SWS_CnV2xMsg_02103]<br>[CP_SWS_CnV2xMsg_02104]<br>[CP_SWS_CnV2xMsg_02105] |

| Requirement | Description | Satisfied by |
|---|---|---|
| | | [CP_SWS_CnV2xMsg_02107]<br>[CP_SWS_CnV2xMsg_02108]<br>[CP_SWS_CnV2xMsg_02109]<br>[CP_SWS_CnV2xMsg_02110]<br>[CP_SWS_CnV2xMsg_02111]<br>[CP_SWS_CnV2xMsg_02112]<br>[CP_SWS_CnV2xMsg_02113]<br>[CP_SWS_CnV2xMsg_02114]<br>[CP_SWS_CnV2xMsg_02115]<br>[CP_SWS_CnV2xMsg_02116]<br>[CP_SWS_CnV2xMsg_02117]<br>[CP_SWS_CnV2xMsg_02118]<br>[CP_SWS_CnV2xMsg_02119]<br>[CP_SWS_CnV2xMsg_02120]<br>[CP_SWS_CnV2xMsg_02121]<br>[CP_SWS_CnV2xMsg_02122]<br>[CP_SWS_CnV2xMsg_02123]<br>[CP_SWS_CnV2xMsg_02124]<br>[CP_SWS_CnV2xMsg_02125]<br>[CP_SWS_CnV2xMsg_02126]<br>[CP_SWS_CnV2xMsg_02127]<br>[CP_SWS_CnV2xMsg_02128]<br>[CP_SWS_CnV2xMsg_02129]<br>[CP_SWS_CnV2xMsg_02130]<br>[CP_SWS_CnV2xMsg_02131]<br>[CP_SWS_CnV2xMsg_02132]<br>[CP_SWS_CnV2xMsg_02133]<br>[CP_SWS_CnV2xMsg_02134]<br>[CP_SWS_CnV2xMsg_02135]<br>[CP_SWS_CnV2xMsg_02136]<br>[CP_SWS_CnV2xMsg_02137]<br>[CP_SWS_CnV2xMsg_02138]<br>[CP_SWS_CnV2xMsg_02139]<br>[CP_SWS_CnV2xMsg_02140]<br>[CP_SWS_CnV2xMsg_02141]<br>[CP_SWS_CnV2xMsg_02142]<br>[CP_SWS_CnV2xMsg_02143]<br>[CP_SWS_CnV2xMsg_02144]<br>[CP_SWS_CnV2xMsg_07001]<br>[CP_SWS_CnV2xMsg_07002]<br>[CP_SWS_CnV2xMsg_07003]<br>[CP_SWS_CnV2xMsg_07004]<br>[CP_SWS_CnV2xMsg_07005]<br>[CP_SWS_CnV2xMsg_07006]<br>[CP_SWS_CnV2xMsg_07007]<br>[SWS_CnV2xMsg_00205] |
| **[CP_SRS_CnV2X_-00502]** | The message layer of Chinese V2X communication shall meet the minimum criteria for data transmission when sending BSM messages | [CP_SWS_CnV2xMsg_00206] |
| **[CP_SRS_CnV2X_-00503]** | The message layer of Chinese V2X communication shall support critical BSM messages | [CP_SWS_CnV2xMsg_00209]<br>[CP_SWS_CnV2xMsg_00210] |

| Requirement | Description | Satisfied by |
|---|---|---|
| **[CP_SRS_CnV2X_-00504]** | The message layer of Chinese V2X communication shall support priority setting for different types of BSMs | [CP_SWS_CnV2xMsg_00213] |
| **[CP_SRS_CnV2X_-00506]** | The message layer of Chinese V2X communication shall generate and send path histories in BSMs | [CP_SWS_CnV2xMsg_00211] [CP_SWS_CnV2xMsg_00214] [CP_SWS_CnV2xMsg_00215] [CP_SWS_CnV2xMsg_00216] [CP_SWS_CnV2xMsg_00217] [CP_SWS_CnV2xMsg_00218] [CP_SWS_CnV2xMsg_00219] [CP_SWS_CnV2xMsg_00220] [CP_SWS_CnV2xMsg_00221] [CP_SWS_CnV2xMsg_00222] [CP_SWS_CnV2xMsg_00223] |
| **[CP_SRS_CnV2X_-00507]** | The message layer of Chinese V2X communication shall manage BSM transmission in such a way that no outdated BSM will be transmitted | [CP_SWS_CnV2xMsg_00208] [CP_SWS_CnV2xMsg_00212] [CP_SWS_CnV2xMsg_00306] [CP_SWS_CnV2xMsg_00307] |
| **[CP_SRS_CnV2X_-00508]** | The message layer of Chinese V2X communication shall support receiving RSI messages | [CP_SWS_CnV2xMsg_00101] [CP_SWS_CnV2xMsg_00203] [CP_SWS_CnV2xMsg_00301] [CP_SWS_CnV2xMsg_00306] [CP_SWS_CnV2xMsg_00307] |
| **[CP_SRS_CnV2X_-00509]** | The message layer of Chinese V2X communication shall support receiving RSM messages | [CP_SWS_CnV2xMsg_00102] [CP_SWS_CnV2xMsg_00302] [CP_SWS_CnV2xMsg_00306] [CP_SWS_CnV2xMsg_00307] |
| **[CP_SRS_CnV2X_-00510]** | The message layer of Chinese V2X communication shall support receiving SPAT messages | [CP_SWS_CnV2xMsg_00103] [CP_SWS_CnV2xMsg_00303] [CP_SWS_CnV2xMsg_00306] [CP_SWS_CnV2xMsg_00307] |
| **[CP_SRS_CnV2X_-00511]** | The message layer of Chinese V2X communication shall support receiving MAP messages | [CP_SWS_CnV2xMsg_00104] [CP_SWS_CnV2xMsg_00304] [CP_SWS_CnV2xMsg_00306] [CP_SWS_CnV2xMsg_00307] |
| **[CP_SRS_CnV2X_-00604]** | The Chinese V2X communication shall not transmit BSMs when it has no valid certificates | [CP_SWS_CnV2xMsg_00230] |
| **[CP_SRS_CnV2X_-00605]** | The Chinese V2X communication shall randomize the identifiers related to BSM to in order to support privacy | [CP_SWS_CnV2xMsg_00410] [CP_SWS_CnV2xMsg_00411] [CP_SWS_CnV2xMsg_00413] [CP_SWS_CnV2xMsg_00414] [CP_SWS_CnV2xMsg_00415] [CP_SWS_CnV2xMsg_00416] [CP_SWS_CnV2xMsg_00417] [CP_SWS_CnV2xMsg_00418] |
| **[SRS_BSW_00345]** | BSW Modules shall support pre-compile configuration | [SWS_CnV2xMsg_08001] |

| Requirement | Description | Satisfied by |
|---|---|---|
| **[SRS_V2X_00711]** | The V2X system's CA basic service shall be compliant to ETSI Specification of Cooperative Awareness Basic Service | [CP_SWS_CnV2xMsg_00305] |
| **[SRS_V2X_00741]** | The V2X system's DEN basic service shall be compliant to ETSI Specifications of Decentralized Environmental Notification Basic Service | [CP_SWS_CnV2xMsg_00305] |
| **[SRS_V2X_10001]** | The V2X system's Facility layer shall support receiving IVI messages | [CP_SWS_CnV2xMsg_00305] [CP_SWS_CnV2xMsg_01051] |
| **[SRS_V2X_10003]** | The V2X system's Facility layer shall support receiving MAPEM messages | [CP_SWS_CnV2xMsg_00305] |
| **[SRS_V2X_10004]** | The V2X system's Facility layer shall support receiving SPAT extended messages | [CP_SWS_CnV2xMsg_00305] |

# 7 Functional Specification

The CnV2xMsg module operates the basic services of BSM, RSI, RSM, SPAT and MAP.

**[CP_SWS_CnV2xMsg_00100]**{DRAFT} ⌈The CnV2xMsg module shall implement the BSM Basic Service following technical requirements specified in [1] [7].⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_00101]**{DRAFT} ⌈The CnV2xMsg module shall implement the RSI Basic Service following technical requirements specified in [7].⌋*(CP_SRS_CnV2X_00508)*

**[CP_SWS_CnV2xMsg_00102]**{DRAFT} ⌈The CnV2xMsg module shall implement the RSM Basic Service following technical requirements specified in [7].⌋*(CP_SRS_CnV2X_00509)*

**[CP_SWS_CnV2xMsg_00103]**{DRAFT} ⌈The CnV2xMsg module shall implement the SPAT Basic Service following technical requirements specified in [7].⌋*(CP_SRS_CnV2X_00510)*

**[CP_SWS_CnV2xMsg_00104]**{DRAFT} ⌈The CnV2xMsg module shall implement the MAP Basic Service following technical requirements specified in [7].⌋*(CP_SRS_CnV2X_00511)*

## 7.1 Startup Behavior

**[CP_SWS_CnV2xMsg_00105]**{DRAFT} ⌈The function CnV2xMsg_Init (see Chapter 8.3.1) of the CnV2xMsg shall initialize the internal states of the CnV2xMsg module.⌋*(CP_SRS_CnV2X_00100)*

**[CP_SWS_CnV2xMsg_00106]**{DRAFT} ⌈The function CnV2xMsg_Init shall initialize the basic services of BSM, RSI, RSM, SPAT and MAP if the received V2X messages are directly sent to application layer via RTE.⌋*(CP_SRS_CnV2X_00100)*

**[CP_SWS_CnV2xMsg_00111]**{DRAFT} ⌈The function CnV2xMsg_Init shall initialize message reception service (see chapter 8.5.7) if the received V2X messages are sent to application layer or PDUR via V2xDm module.⌋*(CP_SRS_CnV2X_00100)*

**[CP_SWS_CnV2xMsg_00107]**{DRAFT} ⌈When system start-up, the CnV2xMsg shall read the heading value from NvM as the initial value.⌋*(CP_SRS_CnV2X_00100)*

## 7.2 Shutdown Behavior

**[CP_SWS_CnV2xMsg_00110]**{DRAFT} ⌈When system shutdown, the CnV2xMsg shall store the last known heading value in NvM.⌋*(CP_SRS_CnV2X_00100)*

## 7.3 General Format Specification

**[CP_SWS_CnV2xMsg_00108]**{DRAFT} ⌈The data elements which constitute the content of the BSM shall be compliant to [1] [7].⌋*(CP_SRS_CnV2X_00100)*

**[CP_SWS_CnV2xMsg_00109]**{DRAFT} ⌈The data elements which constitute the content of the RSI, RSM, SPAT and MAP shall be compliant to [7].⌋*(CP_SRS_CnV2X_00100)*

## 7.4 BSM Functional Specification

### 7.4.1 BSM Initialization

**[CP_SWS_CnV2xMsg_00201]**{DRAFT} ⌈BSM basic service initialization shall enable the transmission of BSMs.⌋*(CP_SRS_CnV2X_00100)*

**[SWS_CnV2xMsg_00202]**{DRAFT} ⌈The function CnV2xMsg_Init shall initialize the generation interval of BSM to default value (100ms) according to chapter 6.3.4 [1].⌋*(CP_SRS_CnV2X_00100)*

**[CP_SWS_CnV2xMsg_00230]**{DRAFT} ⌈CnV2xMsg module shall begin to compose and send BSM messages when CnV2xMsg_CommitPseudonymChange is first received.⌋*(CP_SRS_CnV2X_00604)*

### 7.4.2 BSM Generation, Sending and Receiving, Frequency Management

**[CP_SWS_CnV2xMsg_00203]**{DRAFT} ⌈The BSM basic service shall periodically generate BSMs controlled by the frequency management (For details see chapter 6.3.4 [1]).⌋*(CP_SRS_CnV2X_00508)*

**[CP_SWS_CnV2xMsg_00204]**{DRAFT} ⌈The generated BSMs shall be transmitted by the CnV2xNet using the API function CnV2xNet_Transmit (see chapter 8.6.1)⌋*(CP_SRS_CnV2X_00501)*

**[SWS_CnV2xMsg_00205]**{DRAFT} ⌈The BSM basic service shall receive BSMs via the callback function CnV2xMsg_RxIndication (see chapter 8.4.2)⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_00206]**{DRAFT} ⌈The BSM basic service shall transmit a BSM only if the BSM meets the minimum criteria for BSM transmission specified in chapter 6.3.2 [1]. If at any time the BSM basic service cannot formulate a BSM that meets the minimum transmission criteria, the BSM basic service shall stop transmitting BSMs until the criteria is met.⌋*(CP_SRS_CnV2X_00502)*

**[CP_SWS_CnV2xMsg_00208]**{DRAFT} ⌈For the first regular BSM to be transmitted after the vehicle startup, the CnV2xMsg module shall generate this message within [0,100] ms since the minimum transmission criteria is met.⌋*(CP_SRS_CnV2X_00507)*

**[CP_SWS_CnV2xMsg_00209]**{DRAFT} ⌈When a critical-event trigger condition (for details see chapter 6.3.3 [1]) is first satisfied, the CnV2xMsg module shall cancel the next BSM transmission, and generate a critical BSM immediately and sent it out as soon as possible. CnV2xMsg module shall include all valid critical event flags (up to the time of BSM composition) into this BSM. During the time that the trigger condition is valid, the CnV2xMsg module shall generate critical BSM with a default period of 100 ms starting at the time of the above critical BSM is generated.⌋*(CP_SRS_CnV2X_00503)*

**[CP_SWS_CnV2xMsg_00210]**{DRAFT} ⌈When a specific trigger condition is invalid, the corresponding critical key event flag carried in the BSM message shall be canceled.⌋*(CP_SRS_CnV2X_00503)*

**[CP_SWS_CnV2xMsg_00211]**{DRAFT} ⌈The path history information shall be carried in the first BSM after the time elapsed since the last BSM carries path history information is equal to or greater than 500 ms.⌋*(CP_SRS_CnV2X_00506)*

### 7.4.3 BSM Time Requirement

**[CP_SWS_CnV2xMsg_00212]**{DRAFT} ⌈The CnV2xMsg module shall make sure the time deviation between the value indicated by DSecond in BSM and the UTC time generating the BSM less than 150 ms.⌋*(CP_SRS_CnV2X_00507)*

### 7.4.4 BSM Format Specification

For details about BSM data format refer to the following documents:
See [7] chapter 5
See [1] chapter 6.3.1 and chapter 6.3.2

**[CP_SWS_CnV2xMsg_00213]**{DRAFT} ⌈The priority value of a regular BSM message (without carrying critical flags) shall be set to 112 . The priority value of a critical BSM message (carrying critical flags) shall be set to 208 .⌋*(CP_SRS_CnV2X_00504)*

### 7.4.5 Path History

**[CP_SWS_CnV2xMsg_00214]**{DRAFT} ⌈The CnV2xMsg module shall clear path history cache when the security entity changes its pseudonym certificate.⌋*(CP_SRS_CnV2X_00506)*

**[CP_SWS_CnV2xMsg_00215]**{DRAFT} ⌈For the seting of DF_PathHistoryPoint included in DF_PathHistoryPointList for a BSM that includes path history information, The CnV2xMsg module shall select the corresponding data frame format according to the actual size of the data to be sent, and the larger data frame format shall not be used to send the smaller size data.⌋*(CP_SRS_CnV2X_00506)*

**[CP_SWS_CnV2xMsg_00216]**{DRAFT} ⌈CnV2xMsg_PathHistoryType shall not include any additional data that already exist in other part of the BSM.⌋*(CP_SRS_CnV2X_00506)*

**[CP_SWS_CnV2xMsg_00217]**{DRAFT} ⌈The CnV2xMsg module shall include path history point in DF_PathHistory for a BSM that includes path history information, and the length of path history (i.e. the distance between the first path history point and last path history point) shall equal to or greater than vMinPHistDistance (200 m) and no more than vMaxPHistDistance (400 m), unless the following conditions:

- After the vehicle selects a new pseudonym certificate, the physical distance between the current vehicle's position and the position that the vehicles starting to use the current pseudonym certificate is less than vMinPHistDistance (200 m);
- The position information is unavailable, and the length of path history is less than vMinPHistDistance(200 m);
- The number of path history points included in BSM is greater than vMaxPHistPoints, and the length of path history is still less than vMinPHistDistance (200 m).

⌋*(CP_SRS_CnV2X_00506)*
Note: path history related parameter setting is listed in [1], Appendix B.

**[CP_SWS_CnV2xMsg_00218]**{DRAFT} ⌈The CnV2xMsg module shall maintain a vehicle path comprised of data elements derived from the Positioning Subsystem sampled at a periodic time interval (typically the same as the rate of BSM transmissions) representing the vehicle's recent movement over a corresponding distance.⌋*(CP_SRS_CnV2X_00506)*

**[CP_SWS_CnV2xMsg_00219]**{DRAFT} ⌈The CnV2xMsg module shall populate CnV2xMsg_PathHistoryType with path history points such that the perpendicular distance between any point on the vehicle path and the straight line connecting its two adjacent path history points is less than vPathPerpendicularDist (1 m). (For details, see [1] appendix B)⌋*(CP_SRS_CnV2X_00506)*

**[CP_SWS_CnV2xMsg_00220]**{DRAFT} ⌈The CnV2xMsg module shall populate CnV2xMsg_PathHistoryType with the minimum number of path history points, which are selected from a subset of the available vehicle position data.⌋*(CP_SRS_CnV2X_00506)*

**[CP_SWS_CnV2xMsg_00221]**{DRAFT} ⌈The CnV2xMsg module shall populate CnV2xMsg_PathHistoryType with path history points in chronological time-ordered path history points, with the The first path history point being generating time is the closest in time to the current UTC time.⌋*(CP_SRS_CnV2X_00506)*
Note: Time-ordered path history points are not required to be spaced equally in time.

**[CP_SWS_CnV2xMsg_00222]**{DRAFT} ⌈The CnV2xMsg module shall populate CnV2xMsg_PathHistoryType with not more than vMaxPHistPoints points(15) from the computed set of points.⌋*(CP_SRS_CnV2X_00506)*

**[CP_SWS_CnV2xMsg_00223]**{DRAFT} ⌈The offset value of each path history point shall be based on CnV2xMsg_Position3DType in the BSM.⌋*(CP_SRS_CnV2X_00506)*

## 7.5 RSI Functional Specification

### 7.5.1 RSI Reception Management

**[CP_SWS_CnV2xMsg_00301]**{DRAFT} ⌈Upon receiving a RSI, the RSI service makes the content of the RSI available to the V2X applications (for details see [7] chapter 5). Received RSIs can be sent to the upper application layer via standardized AUTOSAR service interface CnV2xAppRxIndicationRsi or via V2xDm. It can be configured by CnV2xMsgV2xDmServiceConfig (See chapter 10.1.5).⌋*(CP_SRS_CnV2X_00508)*

### 7.5.2 RSI Format Specification

For details about RSI data format refer to CCSA standards: [7] chapter 5.

## 7.6 RSM Functional Specification

### 7.6.1 RSM Reception Management

**[CP_SWS_CnV2xMsg_00302]**{DRAFT} ⌈Upon receiving a RSM, the RSM service makes the content of the RSM available to the V2X applications (for details see [7] chapter 5). Received RSMs can be sent to the upper application layer via standardized AUTOSAR service interface CnV2xAppRxIndicationRsm or via V2xDm. It can be configured by CnV2xMsgV2xDmServiceConfig (See chapter 10.1.5).⌋*(CP_SRS_CnV2X_00509)*

### 7.6.2 RSM Format Specification

For details about RSM data format refer to CCSA standards: [7] chapter 5.

## 7.7 SPAT Functional Specification

### 7.7.1 SPAT Reception Management

**[CP_SWS_CnV2xMsg_00303]**{DRAFT} ⌈Upon receiving a SPAT, the SPAT service makes the content of the SPAT available to the V2X applications (for details see [7]

chapter 5). Received SPATs can be sent to the upper application layer via standardized AUTOSAR service interface CnV2xAppRxIndicationSpat or via V2xDm. It can be configured by CnV2xMsgV2xDmServiceConfig (See chapter 10.1.5).⌋*(CP_SRS_CnV2X_00510)*

### 7.7.2 SPAT Format Specification

For details about SPAT data format refer to CCSA standards: [7] chapter 5.

## 7.8 MAP Functional Specification

### 7.8.1 MAP Reception Management

**[CP_SWS_CnV2xMsg_00304]**{DRAFT} ⌈Upon receiving a MAP, the MAP service makes the content of the MAP available to the V2X applications (for details see [7] chapter 5). Received MAPs can be sent to the upper application layer via standardized AUTOSAR service interface CnV2xAppRxIndicationMap or via V2xDm. It can be configured by CnV2xMsgV2xDmServiceConfig (See chapter 10.1.5).⌋*(CP_SRS_CnV2X_00511)*

### 7.8.2 MAP Format Specification

For details about MAP data format refer to CCSA standards: [7] chapter 5.

## 7.9 Position and Time

**[CP_SWS_CnV2xMsg_00401]**{DRAFT} ⌈GCJ-02 shall be used as the reference coordinate system as defined in [1].⌋*(CP_SRS_CnV2X_00203)*

**[CP_SWS_CnV2xMsg_00402]**{DRAFT} ⌈Heading shall describe the direction of the vehicle reference point, and its value increases clockwise from north as defined in [7].⌋*(CP_SRS_CnV2X_00203)*

**[CP_SWS_CnV2xMsg_00403]**{DRAFT} ⌈The function CnV2xMsg_CheckDistance shall provide the currently distance between current position and the position where the current Pseudonym beginning to be used.⌋*(CP_SRS_CnV2X_00100)*

**[CP_SWS_CnV2xMsg_00404]**{DRAFT} ⌈The function CnV2xMsg_GetRefTimePtr shall provide an address pointer to 32 bit data containing the current UTC Time.⌋*(CP_SRS_CnV2X_00201)*

**[CP_SWS_CnV2xMsg_00405]**{DRAFT} ⌈The function CnV2xMsg_CalcDistance shall calculate the distance between two geographical points.⌋*(CP_SRS_CnV2X_00100)*

**[CP_SWS_CnV2xMsg_00406]**{DRAFT} ⌈CnV2xMsg module shall update and record the vehicle position when received CnV2xMsg_CommitPseudonymChange, which is used for calculating the distance by the function CnV2xMsg_CheckDistance.⌋*(CP_SRS_CnV2X_00100)*

## 7.10 ID Management

**[CP_SWS_CnV2xMsg_00410]**{DRAFT} ⌈The CnV2xMsg module shall implement the identity management. Specific modules shall be notified with the current identity to ensure a consistent value is used in each layer of Chinese V2X stack.⌋*(CP_SRS_CnV2X_00605)*

**[CP_SWS_CnV2xMsg_00411]**{DRAFT} ⌈When received the pseudonym certificate change from CnV2xSec, CnV2xMsg module shall change application identifiers (Vehicle ID and Message count), and inform the CnV2xNet module the changes. Those changes are necessary to ensure the privacy of the vehicle.⌋*(CP_SRS_CnV2X_00605)*

**[CP_SWS_CnV2xMsg_00413]**{DRAFT} ⌈The CnV2xMsg_Mgt_MainFunction shall be used to manage identifier changes.⌋*(CP_SRS_CnV2X_00605)*

**[CP_SWS_CnV2xMsg_00414]**{DRAFT} ⌈The CnV2xMsg shall initiate a change of the identifiers within two phases. A first prepare phase and a second commit or abort phase. The second phase depends on the result of all called modules within the first phase. If the first phase was successful, the commit phase shall be initiated, if the first phase was unsuccessful, the abort phase shall be initiated.⌋*(CP_SRS_CnV2X_00605)*

**[CP_SWS_CnV2xMsg_00415]**{DRAFT} ⌈In the prepare phase, the API CnV2xMsg_PreparePseudonymChange() shall be called by CnV2xSec and then CnV2xNet_PrepareAppLayerIdChange() shall be called by CnV2xMsg.⌋*(CP_SRS_CnV2X_00605)*

**[CP_SWS_CnV2xMsg_00416]**{DRAFT} ⌈In the commit phase, the API CnV2xMsg_CommitPseudonymChange() shall be called by CnV2xSec and then CnV2xNet_CommitAppLayerIdChange() shall be called by CnV2xMsg. After that new Pseudonym certificate and Pseudonym Count value shall take effect, V2X Message with old Pseudonym count value shall be discarded.⌋*(CP_SRS_CnV2X_00605)*

**[CP_SWS_CnV2xMsg_00417]**{DRAFT} ⌈In the abort phase, the API CnV2xMsg_CommitPseudonymChange() shall be called by CnV2xSec and then CnV2xNet_AbortAppLayerIdChange() shall be called.⌋*(CP_SRS_CnV2X_00605)*

**[CP_SWS_CnV2xMsg_00418]**{DRAFT} ⌈When the vehicle Event Flags are changed to the status that all bits are unset or from the status that all bits are unset to the status that any bit is set, the function CnV2xMsg_GetVehicleEventFlagsStatus shall be called by CnV2xSec to initiate a change of the pseudonym certificate.⌋*(CP_SRS_CnV2X_00605)*

## 7.11 Messages Reception Service Via V2xDm

**[CP_SWS_CnV2xMsg_00305]**{DRAFT} ⌈If the received V2X messages are configured to be sent to V2xDm module, the received messages shall be sent via the callback function V2xDm_RxIndication (see chapter 8.6.2).⌋*(SRS_V2X_00711, SRS_V2X_00741, SRS_V2X_10001, SRS_V2X_10003, SRS_V2X_10004)*

**[CP_SWS_CnV2xMsg_00306]**{DRAFT} ⌈

AIDs need to be assigned to the corresponding instance of the configuration container of CnV2xMsgConfig (see Chapter 10.1.5). The CnV2xMsg module shall check whether the AID of the received message matches the configuration as specified in [SWS_CnV2xMsg_00307]. If not, the message shall be discarded.⌋*(CP_SRS_CnV2X_00507, CP_SRS_CnV2X_00508, CP_SRS_CnV2X_00509, CP_SRS_CnV2X_00510, CP_SRS_CnV2X_00511)*

**[CP_SWS_CnV2xMsg_00307]**{DRAFT} ⌈

| *Message Type* | *AID* | *Rx/Tx* |
|---|---|---|
| BSM | 111(Non-Emergeny vehicle,regular BSM)<br>112 (Non-Emergency vehicle, event-triggered BSM)<br>113(Emergency vehicle, regular BSM)<br>114(Emergency vehicle, event-triggered BSM)<br>3617(for V2X terminal installed after market ) | Rx and Tx |
| RSI | 3620(Static roadside information)<br>3621(Semi-dynamic roadside information)<br>3622(Dynamic roadside information) | Rx only |
| RSM | 3623 | Rx only |
| SPAT | 3619 | Rx only |
| MAP | 3618 | Rx only |

⌋*(CP_SRS_CnV2X_00507, CP_SRS_CnV2X_00508, CP_SRS_CnV2X_00509, CP_SRS_CnV2X_00510, CP_SRS_CnV2X_00511)*

## 7.12 Error Classification

### 7.12.1 Development Errors

**[CP_SWS_CnV2xMsg_00501] Development Error Types** ⌈

| *Type of error* | *Related error code* | *Value [hex]* |
|---|---|---|
| API service called with wrong parameter | CNV2XMSG_E_PARAM | 0x01 |
| API service called with invalid pointer | CNV2XMSG_E_PARAM_POINTER | 0x02 |
| CnV2xMsg initialization failed | CNV2XMSG_E_INIT_FAILED | 0x03 |
| API function called before the CnV2xMsg module has been fully initialized | CNV2XMSG_E_UNINIT | 0x04 |

⌋*()*

### 7.12.2 Runtime Errors

There is no runtime errors.

### 7.12.3 Transient Faults

There is no Transient Faults.

### 7.12.4 Production Errors

There is no production errors.

### 7.12.5 Extended Production Errors

There is no extended production errors.

# 8 API specification

## 8.1 Imported types

In this chapter all types included from the following files are listed.

**[CP_SWS_CnV2xMsg_01001]** ⌈

| *Module* | *Header File* | *Imported Type* |
|---|---|---|
| CnV2xNet | CnV2x_GeneralTypes.h | CnV2xNet_TxParamsPresenceType (draft) |
| | CnV2x_GeneralTypes.h | CnV2x_CbrType (draft) |
| | CnV2x_GeneralTypes.h | CnV2x_Layer2IdType (draft) |
| | CnV2x_GeneralTypes.h | CnV2x_MaxDataRateType (draft) |
| | CnV2x_GeneralTypes.h | CnV2x_NetTxResultType (draft) |
| | CnV2x_GeneralTypes.h | CnV2x_NetworkProtocolType (draft) |
| | CnV2x_GeneralTypes.h | CnV2x_TrafficPeriodType (draft) |
| | CnV2xNet.h | CnV2xNet_TxParamsType (draft) |
| CnV2xSec | CnV2x_GeneralTypes.h | CnV2xSec_SecReportType (draft) |
| | CnV2x_Sec.h | CnV2xSec_SecProfileType (draft) |
| | CnV2x_Sec.h | CnV2xSec_SecReturnType (draft) |
| Std | Std_Types.h | Std_ReturnType |
| | Std_Types.h | Std_VersionInfoType |

⌋()

## 8.2 Type definitions

### 8.2.1 CnV2xMsg_RxParamsType

**[CP_SWS_CnV2xMsg_01002]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_RxParamsType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | presence | |
| | ***Type*** | CnV2xMsg_RxParamsPresenceType |
| | ***Comment*** | Mark optional child present or not |
| | DsmpVersion | |
| | ***Type*** | uint8 |
| | ***Comment*** | DSMP protocol version type. Range: 0..7 |
| | Aid | |
| | ***Type*** | uint64 |
| | ***Comment*** | The value of the AID (Application Identifier) |
| | SourceLayer2Id | |

▽

△

| | | |
|---|---|---|
| | ***Type*** | CnV2x_Layer2IdType |
| | ***Comment*** | Source MAC address of V2X-CN packet |
| | DestinationLayer2Id | |
| | ***Type*** | CnV2x_Layer2IdType |
| | ***Comment*** | Destination MAC address of V2X-CN packet |
| | Priority | |
| | ***Type*** | uint8 |
| | ***Comment*** | Specify the priority of V2X-CN message |
| | Cbr | |
| | ***Type*** | CnV2x_CbrType |
| | ***Comment*** | Indication of Channel busy ratio |
| | MaxDataRate | |
| | ***Type*** | CnV2x_MaxDataRateType |
| | ***Comment*** | Indication of Max data rate |
| ***Description*** | Wraps Network layer parameters from CnV2xNet<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | CnV2xMsg.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.2.2 CnV2xMsg_RxParamsPresenceType

**[CP_SWS_CnV2xMsg_01056]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_RxParamsPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | SourceMACAddr | 0x08 | Bit 3: Optional child present |
| | bit | DestinationLayer2Id | 0x04 | Bit 2: Optional child present |
| | bit | Cbr | 0x02 | Bit 1: Optional child present |
| | bit | MaxdataRate | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_RxParamsType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | CnV2xMsg.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

## 8.3 Function definitions

### 8.3.1 CnV2xMsg_Init

**[CP_SWS_CnV2xMsg_01003]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_Init (draft) | |
| ***Syntax*** | `void CnV2xMsg_Init (`<br>`  void* CfgPtr`<br>`)` | |
| ***Service ID [hex]*** | 0x1 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | CfgPtr | Points to a null pointer |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Initialize the CnV2xMsg module<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01053]**{DRAFT} ⌈If development error detection is enabled: the function shall check the parameter CfgPtr for containing a valid configuration. If the check fails, the function shall raise the development error CNV2XMSG_E_INIT_FAILED.⌋*()*

### 8.3.2 CnV2xMsg_GetVersionInfo

**[CP_SWS_CnV2xMsg_01004]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_GetVersionInfo (draft) | |
| ***Syntax*** | `void CnV2xMsg_GetVersionInfo (`<br>`  Std_VersionInfoType* VersionInfoPtr`<br>`)` | |
| ***Service ID [hex]*** | 0x2 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | None | |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | VersionInfoPtr | Pointer to where to store the version information of this module. |
| ***Return value*** | None | |
| ***Description*** | Returns the version information of this module.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01005]**{DRAFT} ⌈If CnV2xMsgDevErrorDetect (for details see Chapter 10.1.3) is enabled: If the VersionInfoPtr pointer parameter is invalid (e.g. NULL), the error-code CNV2XMSG_E_PARAM_POINTER shall be reported to the DET module.⌋*()*

### 8.3.3 CnV2xMsg_GetRefTimePtr

**[CP_SWS_CnV2xMsg_01009]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_GetRefTimePtr (draft) | |
| ***Syntax*** | `Std_ReturnType CnV2xMsg_GetRefTimePtr ( const uint32** RefTimePtr )` | |
| ***Service ID [hex]*** | 0x3 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | None | |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | RefTimePtr | Pointer to the current time information. |
| ***Return value*** | Std_ReturnType | E_OK: request successful E_NOT_OK: request failed |
| ***Description*** | Provides a pointer to the time reference of the Chinese V2X Stack.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01010]**{DRAFT} ⌈If development error detection is enabled: the function shall check that the service CnV2xMsg_Init was previously called. If the check fails, the function shall raise the development error CNV2XMSG_E_UNINIT.⌋*()*

**[CP_SWS_CnV2xMsg_01011]**{DRAFT} ⌈If development error detection is enabled: the function shall check the parameter RefTimePtr for being valid. If the check fails, the function shall raise the development error CNV2XMSG_E_PARAM_POINTER.⌋*()*

### 8.3.4 CnV2xMsg_CheckDistance

**[CP_SWS_CnV2xMsg_01012]**{DRAFT} ⌈

| | |
|---|---|
| ***Service Name*** | CnV2xMsg_CheckDistance (draft) |
| ***Syntax*** | `Std_ReturnType CnV2xMsg_CheckDistance ( float32* Distance )` |
| ***Service ID [hex]*** | 0x4 |
| ***Sync/Async*** | Synchronous |
| ***Reentrancy*** | Reentrant |

▽

△

| | | |
|---|---|---|
| ***Parameters (in)*** | None | |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | Distance | Distance between geographical points A and B [m] |
| ***Return value*** | Std_ReturnType | E_OK: operation successful E_NOT_OK: pseudonym certificate change rejected |
| ***Description*** | Check the distance between the current geographical point and the point when the CnV2xSec commit the pseudonym certificate change on elevation 0.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01013]**{DRAFT} ⌈If development error detection is enabled: the function shall check the parameter Distance for being valid. If the check fails, the function shall raise the development error CNV2XMSG_E_PARAM_POINTER.⌋*()*

### 8.3.5 CnV2xMsg_GetVehickeEventFlagsStatus

**[CP_SWS_CnV2xMsg_01061]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_GetVehickeEventFlagsStatus (draft) | |
| ***Syntax*** | `Std_ReturnType CnV2xMsg_GetVehickeEventFlagsStatus (`<br>`  CnV2xMsg_VehicleEventFlagsType** vehicleEventFlagsPtr`<br>`)` | |
| ***Service ID [hex]*** | 0x5 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | None | |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | vehicleEventFlagsPtr | Pointer to the current Event flags status. |
| ***Return value*** | Std_ReturnType | E_OK: operation successful E_NOT_OK: pseudonym certificate change rejected |
| ***Description*** | Provides a pointer to the current vehicle event status.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01062]**{DRAFT} ⌈If development error detection is enabled: the function shall check the parameter vehicleEventFlagsPtr for being valid. If the check fails, the function shall raise the development error CNV2XMSG_E_PARAM_POINTER⌋*()*

### 8.3.6 CnV2xMsg_PreparePseudonymChange

**[CP_SWS_CnV2xMsg_01014]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_PreparePseudonymChange (draft) | |
| ***Syntax*** | `void CnV2xMsg_PreparePseudonymChange (`<br>`  uint16 msgClass,`<br>`  uint16 pseudonymCount16`<br>`)` | |
| ***Service ID [hex]*** | 0x6 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | msgClass | Indicate message Class |
| | pseudonymCount16 | Oder of the Pseudonym certificate change correspond to specific message type . This count value is created in the CnV2xSec module. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | By this API primitive the CnV2xMsg module gets an indication that the given Pseudonym certificate and hereby the Msg count and Vehicle ID is about to be changed.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01015]**{DRAFT} ⌈The function CnV2xMsg_PreparePseudonymChange shall prepare the setting of message count and vehicle ID used for packet transmission.⌋*()*

**[CP_SWS_CnV2xMsg_01016]**{DRAFT} ⌈If development error detection is enabled: the function shall check that the service CnV2xMsg_Init was previously called. If the check fails, the function shall raise the development error CNV2XMSG_E_UNINIT.⌋*()*

### 8.3.7 CnV2xMsg_CommitPseudonymChange

**[CP_SWS_CnV2xMsg_01018]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_CommitPseudonymChange (draft) | |
| ***Syntax*** | `Std_ReturnType CnV2xMsg_CommitPseudonymChange (`<br>`  uint16 msgClass,`<br>`  uint16 pseudonymCount16`<br>`)` | |
| ***Service ID [hex]*** | 0x7 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | msgClass | Indicate message Class |

▽

△

| | | |
|---|---|---|
| | pseudonymCount16 | Oder of the Pseudonym certificate change correspond to specific message type . This count value is created in the CnV2xSec module. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | Std_ReturnType | E_OK: operation successful E_NOT_OK: pseudonym certificate change rejected |
| ***Description*** | This function is called by the CnV2xSec module when all modules are OK with the pseudonym certificate change and the change is to be committed.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋(*CP_SRS_CnV2X_00501*)

**[CP_SWS_CnV2xMsg_01019]**{DRAFT} ⌈The function CnV2xMsg_CommitPseudonymChange shall set the message count and vehicle ID used for packet transmission and clean the path history.⌋*()*

**[CP_SWS_CnV2xMsg_01020]**{DRAFT} ⌈If development error detection is enabled: the function shall check that the service CnV2xMsg_Init was previously called. If the check fails, the function shall raise the development error CNV2XMSG_E_UNINIT.⌋*()*

Note: The function requires previous preparation of the pseudonym certificate via an API call to CnV2xMsg_PreparePseudonymChange.

### 8.3.8 CnV2xMsg_AbortPseudonymChange

**[CP_SWS_CnV2xMsg_01021]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_AbortPseudonymChange (draft) | |
| ***Syntax*** | `Std_ReturnType CnV2xMsg_AbortPseudonymChange ( uint16 msgClass, uint16 pseudonymCount16 )` | |
| ***Service ID [hex]*** | 0x8 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | msgClass | Indicate message Class |
| | pseudonymCount16 | Oder of the Pseudonym certificate change correspond to specific message type . This count value is created in the CnV2xSec module. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | Std_ReturnType | E_OK: operation successful E_NOT_OK: pseudonym certificate change rejected |

▽

△

| | |
|---|---|
| ***Description*** | This function is called by the CnV2xSec module when not all modules are OK with the pseudonym certificate change and the change is to be rolled back.<br>**Tags:** atp.Status=draft |
| ***Available via*** | CnV2xMsg.h |

⌋()

**[CP_SWS_CnV2xMsg_01022]**{DRAFT} ⌈The function CnV2xMsg_AbortPseudonymChange shall roll back the prepared pseudonym certificate change.⌋()

**[CP_SWS_CnV2xMsg_01023]**{DRAFT} ⌈If development error detection is enabled: the function shall check that the service CnV2xMsg_Init was previously called. If the check fails, the function shall raise the development error CNV2XMSG_E_UNINIT.⌋()

Note: The function requires previous preparation of the pseudonym certificate via an API call to CnV2xMsg_PreparePseudonymChange.

## 8.4 Callback notifications

This is a list of functions provided for other modules.

### 8.4.1 CnV2xMsg_TxConfirmation

**[CP_SWS_CnV2xMsg_01024]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_TxConfirmation (draft) | |
| ***Syntax*** | `void CnV2xMsg_TxConfirmation ( uint16 TransactionId16 )` | |
| ***Service ID [hex]*** | 0x9 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | TransactionId16 | TransactionId of the packet that has been transmitted |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | By this API primitive, the CnV2xMsg module gets a confirmation that the V2X message with a certain ID was send successfully.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋(*CP_SRS_CnV2X_00501*)

**[CP_SWS_CnV2xMsg_01025]**{DRAFT} ⌈If development error detection is enabled: the function shall check that the service CnV2xMsg_Init was previously called. If the check fails, the function shall raise the development error CNV2XMSG_E_UNINIT.⌋*()*

### 8.4.2 CnV2xMsg_RxIndication

**[CP_SWS_CnV2xMsg_01026]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_RxIndication (draft) | |
| ***Syntax*** | `void CnV2xMsg_RxIndication (`<br>`  uint32 TransactionId32,`<br>`  CnV2xMsg_RxParamsType* ReceiveParams,`<br>`  uint16 Length,`<br>`  const uint8* DataPtr`<br>`)` | |
| ***Service ID [hex]*** | 0xa | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | TransactionId32 | ID of the received packet. This ID is created in the CnV2xNet module and handed up in the protocol stack to be used for verification on demand. |
| | ReceiveParams | Wraps RxIndication parameters. |
| | Length | Length of the data pointed by DataPtr. |
| | DataPtr | Payload of the received Network packet. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | By this API primitive the CnV2xMsg module gets a confirmation that the V2X message with a certain ID was send successfully. This API primitive is called by the CnV2xNet module providing the data and the Network parameters of a received DSMP packet to CnV2xMsg module.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01027]**{DRAFT} ⌈If development error detection is enabled: the function shall check that the service CnV2xMsg_Init was previously called. If the check fails, the function shall raise the development error CNV2XMSG_E_UNINIT.⌋*()*

**[CP_SWS_CnV2xMsg_01028]**{DRAFT} ⌈If development error detection is enabled: the function shall check the parameter ReceiveParams for being valid. If the check fails, the function shall raise the development error CNV2XMSG_E_PARAM_POINTER.⌋*()*

**[CP_SWS_CnV2xMsg_01029]**{DRAFT} ⌈If development error detection is enabled: the function shall check the parameter DataPtr for being valid. If the check fails, the function shall raise the development error CNV2XMSG_E_PARAM_POINTER.⌋*()*

### 8.4.3 CnV2xMsg_EncapConfirmation

**[CP_SWS_CnV2xMsg_01030]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_EncapConfirmation (draft) | |
| ***Syntax*** | `void CnV2xMsg_EncapConfirmation ( uint16 TransactionId16, uint16* SecuredDataLength, uint8* SecuredDataPtr )` | |
| ***Service ID [hex]*** | 0xb | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | TransactionId16 | TransactionId of the encapsulated packet |
| | SecuredDataLength | lenght of Secured Data |
| | SecuredDataPtr | Pointer of Secured Data |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | This function is called by the V2xSecCN module when an encapsulation has been finished.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01031]**{DRAFT} ⌈The function CnV2xMsg_EncapConfirmation shall finalize the packet transmission by transmitting the packet to the lower layer.⌋*()*

**[CP_SWS_CnV2xMsg_01032]**{DRAFT} ⌈If development error detection is enabled: the function shall check that the service CnV2xMsg_Init was previously called. If the check fails, the function shall raise the development error CNV2XMSG_E_UNINIT.⌋*()*

### 8.4.4 CnV2xMsg_DecapConfirmation

**[CP_SWS_CnV2xMsg_01033]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Service Name*** | CnV2xMsg_DecapConfirmation (draft) | |
| ***Syntax*** | `void CnV2xMsg_DecapConfirmation ( uint32 TransactionId32, CnV2x_SecReportType SecReport, uint64 CertificateId, uint64 Aid )` | |
| ***Service ID [hex]*** | 0xc | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | TransactionId32 | ID of the decapsulated packet |
| | SecReport | The security report. |

▽

△

| | | |
|---|---|---|
| | CertificateId | The identification of the used for verification (by certificate hash) |
| | Aid | The numerical value of the AID |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | This function is called by the CnV2xSec module when a decapsulation has been finished.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | CnV2xMsg.h | |

⌋(*CP_SRS_CnV2X_00501*)

**[CP_SWS_CnV2xMsg_01034]**{DRAFT} ⌈The function CnV2xMsg_DecapConfirmation shall continue the processing of a received packet by proceeding with CnV2xMsg operations.⌋*()*

**[CP_SWS_CnV2xMsg_01035]**{DRAFT} ⌈If development error detection is enabled: the function shall check that the service CnV2xMsg_Init was previously called. If the check fails, the function shall raise the development error CNV2XMSG_E_UNINIT.⌋*()*

## 8.5 Scheduled functions

These functions are directly called by Basic Software Scheduler. The following functions shall have no return value and no parameter. All functions shall be non reentrant.

### 8.5.1 CnV2xMsg_BsmBs_MainFunction

**[CP_SWS_CnV2xMsg_01036]**{DRAFT} ⌈

| | |
|---|---|
| ***Service Name*** | CnV2xMsg_BsmBs_MainFunction (draft) |
| ***Syntax*** | `void CnV2xMsg_BsmBs_MainFunction (`<br>`  void`<br>`)` |
| ***Service ID [hex]*** | 0x0d |
| ***Description*** | This is the main processing function of the BSM basic service<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_CnV2xMsg.h |

⌋(*CP_SRS_CnV2X_00501*)

**[CP_SWS_CnV2xMsg_01037]**{DRAFT} ⌈The function shall process the BSMs as described in chapter 7.4.⌋*()*

### 8.5.2 CnV2xMsg_Mgt_MainFunction

**[CP_SWS_CnV2xMsg_01038]**{DRAFT} ⌈

| | |
|---|---|
| ***Service Name*** | CnV2xMsg_Mgt_MainFunction (draft) |
| ***Syntax*** | `void CnV2xMsg_Mgt_MainFunction ( void )` |
| ***Service ID [hex]*** | 0x0e |
| ***Description*** | Scheduled Management Function of CnV2xMsg<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_CnV2xMsg.h |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01039]**{DRAFT} ⌈The function shall handle sending frequency management, ID management, Position and Time management and Path History Generation.⌋*()*

### 8.5.3 CnV2xMsg_RsiS_MainFunction

**[CP_SWS_CnV2xMsg_01041]**{DRAFT} ⌈

| | |
|---|---|
| ***Service Name*** | CnV2xMsg_RsiS_MainFunction (draft) |
| ***Syntax*** | `void CnV2xMsg_RsiS_MainFunction ( void )` |
| ***Service ID [hex]*** | 0x0f |
| ***Description*** | This is the main processing function of the RSI service<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_CnV2xMsg.h |

⌋*(CP_SRS_CnV2X_00501)*

**[CP_SWS_CnV2xMsg_01042]**{DRAFT} ⌈The function shall process the received RSIs as described in chapter 7.5.⌋*()*

### 8.5.4 CnV2xMsg_RsmS_MainFunction

**[CP_SWS_CnV2xMsg_01043]**{DRAFT} ⌈

| | |
|---|---|
| ***Service Name*** | CnV2xMsg_RsmS_MainFunction (draft) |
| ***Syntax*** | `void CnV2xMsg_RsmS_MainFunction ( void )` |
| ***Service ID [hex]*** | 0x10 |

▽

△

| | |
|---|---|
| ***Description*** | This is the main processing function of the RSM service<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_CnV2xMsg.h |

⌋(*CP_SRS_CnV2X_00501*)

**[CP_SWS_CnV2xMsg_01044]**{DRAFT} ⌈The function shall process the received RSMs as described in chapter 7.6.⌋*()*

### 8.5.5 CnV2xMsg_SpatS_MainFunction

**[CP_SWS_CnV2xMsg_01045]**{DRAFT} ⌈

| | |
|---|---|
| ***Service Name*** | CnV2xMsg_SpatS_MainFunction (draft) |
| ***Syntax*** | `void CnV2xMsg_SpatS_MainFunction (`<br>`  void`<br>`)` |
| ***Service ID [hex]*** | 0x11 |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non Reentrant |
| ***Parameters (in)*** | None |
| ***Parameters (inout)*** | None |
| ***Parameters (out)*** | None |
| ***Return value*** | None |
| ***Description*** | This is the main processing function of the SPAT service<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_CnV2xMsg.h |

⌋(*CP_SRS_CnV2X_00501*)

**[CP_SWS_CnV2xMsg_01046]**{DRAFT} ⌈The function shall process the received SPATs as described in chapter 7.7.⌋*()*

### 8.5.6 CnV2xMsg_MapS_MainFunction

**[CP_SWS_CnV2xMsg_01047]**{DRAFT} ⌈

| | |
|---|---|
| ***Service Name*** | CnV2xMsg_MapS_MainFunction (draft) |
| ***Syntax*** | `void CnV2xMsg_MapS_MainFunction (`<br>`  void`<br>`)` |
| ***Service ID [hex]*** | 0x12 |
| ***Description*** | This is the main processing function of the MAP service<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_CnV2xMsg.h |

⌋(*CP_SRS_CnV2X_00501*)

**[CP_SWS_CnV2xMsg_01048]**{DRAFT} ⌈The function shall process the received MAPs as described in chapter 7.8.⌋*()*

### 8.5.7 CnV2xMsg_RxS_MainFunction

**[CP_SWS_CnV2xMsg_01051]**{DRAFT} ⌈

| | |
|---|---|
| ***Service Name*** | CnV2xMsg_RxS_MainFunction (draft) |
| ***Syntax*** | `void CnV2xMsg_RxS_MainFunction ( void )` |
| ***Service ID [hex]*** | 0x15 |
| ***Description*** | This is the main processing function of the message reception service when the received V2X messages are sent to application layer or PDUR via V2xDm module.<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_CnV2xMsg.h |

⌋*(SRS_V2X_10001)*

**[CP_SWS_CnV2xMsg_01052]**{DRAFT} ⌈When the received V2X messages are sent to application layer or PDUR via V2xDm module, the function shall process the message reception service as described in chapter 7.11.⌋*()*

## 8.6 Expected interfaces

In this chapter all interfaces required from other modules are listed.

### 8.6.1 Mandatory interfaces

This section defines all interfaces, which are required to fulfill the core functionality of the module.

**[CP_SWS_CnV2xMsg_01049]** ⌈

| *API Function* | *Header File* | *Description* |
|---|---|---|
| CnV2xNet_AbortAppLayerIdChange (draft) | CnV2xNet.h | The CnV2xMsg module calls this function when not all modules are OK with the pseudonym certificate change and the change is to be rolled back.<br>**Tags:** atp.Status=draft |
| CnV2xNet_CommitAppLayerIdChange (draft) | CnV2xNet.h | The CnV2xMsg module calls this function when all modules are OK with the pseudonym certificate change and the change is to be committed.<br>**Tags:** atp.Status=draft |

▽

△

| API Function | Header File | Description |
|---|---|---|
| CnV2xNet_PrepareAppLayerIdChange (draft) | CnV2xNet.h | By this API primitive the CnV2xNet module gets an indication that Application Layer Id is about to change and hereby source Layer-2 ID is about to be changed.<br>**Tags:** atp.Status=draft |
| CnV2xNet_Transmit (draft) | CnV2xNet.h | This API is called by the CvxMsgCN module to request sending a Network Layer V2X PDU to the peer Network entity.<br>**Tags:** atp.Status=draft |
| CnV2xSec_ReqDecap (draft) | CnV2xSec.h | This function is called by the CnV2xMsg to decapsulate the SPDU. An asynchronous CnV2x Msg_DecapConfirmation call will be used to notify CnV2xMsg of the result.<br>**Tags:** atp.Status=draft |
| CnV2xSec_ReqEncap (draft) | CnV2xSec.h | This function is called by the CnV2xMsg to generate the SPDU, which includes the V2X message, the signature and pseudonym. An asynchronous CnV2x Msg_EncapConfirmation call will be used to notify CnV2xMsg of the result.<br>**Tags:** atp.Status=draft |

⌋(*CP_SRS_CnV2X_00501*)

### 8.6.2 Optional interfaces

This section defines all interfaces, which are required to fulfill an optional functionality of the module.

**[CP_SWS_CnV2xMsg_01050]** ⌈

| API Function | Header File | Description |
|---|---|---|
| Det_ReportError | Det.h | Service to report development errors. |

⌋(*CP_SRS_CnV2X_00501*)

## 8.7 Service Interfaces

### 8.7.1 Sender-Receiver-Interfaces

#### 8.7.1.1 CnV2xMsgVdp

**[CP_SWS_CnV2xMsg_01101]**{DRAFT} ⌈The CnV2xMsg requires an interface CnV2xMsgVdp as defined below to get data from the VDP application.⌋*()*

**[CP_SWS_CnV2xMsg_01102]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsgVdp (draft) | |
| ***Comment*** | Interface to receive data from VDP application<br>**Tags:** atp.Status=draft | |
| ***IsService*** | false | |
| ***Variation*** | – | |
| ***Data Elements*** | VdpData | |
| | ***Type*** | CnV2xMsg_BsmType |
| | ***Variation*** | – |

⌋*(CP_SRS_CnV2X_00501)*

### 8.7.1.2 CnV2xApplRxIndicationBsm

**[CP_SWS_CnV2xMsg_01103]**{DRAFT} ⌈For the CnV2xMsg, an interface CnV2xApplRxIndicationBsm shall be provided as defined below to provide the capability of delivering received BSMs to applications.⌋*()*

**[CP_SWS_CnV2xMsg_01104]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xApplRxIndicationBsm (draft) | |
| ***Comment*** | Deliver received BSMs to Applications<br>**Tags:** atp.Status=draft | |
| ***IsService*** | true | |
| ***Variation*** | – | |
| ***Data Elements*** | BsmData | |
| | ***Type*** | CnV2xMsg_BsmRootType |
| | ***Variation*** | – |

⌋*(CP_SRS_CnV2X_00501)*

### 8.7.1.3 CnV2xApplRxIndicationzRsi

**[CP_SWS_CnV2xMsg_01105]**{DRAFT} ⌈For the CnV2xMsg, an interface CnV2xApplRxIndicationRsi shall be provided as defined below to provide the capability of delivering received RSIs to applications.⌋*()*

**[CP_SWS_CnV2xMsg_01106]**{DRAFT} ⌈

| | |
|---|---|
| ***Name*** | CnV2xApplRxIndicationRsi (draft) |
| ***Comment*** | Deliver received RSIs to Applications<br>**Tags:** atp.Status=draft |
| ***IsService*** | true |
| ***Variation*** | – |

▽

△

| | | |
|---|---|---|
| ***Data Elements*** | RsiData | |
| | ***Type*** | CnV2xMsg_RsiRootType |
| | ***Variation*** | – |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.1.4 CnV2xApplRxIndicationRsm

**[CP_SWS_CnV2xMsg_01107]**{DRAFT} ⌈For the CnV2xMsg, an interface CnV2xApplRxIndicationRsm shall be provided as defined below to provide the capability of delivering received RSMs to applications.⌋*()*

**[CP_SWS_CnV2xMsg_01108]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xApplRxIndicationRsm (draft) | |
| ***Comment*** | Deliver received RSMs to Applications<br>**Tags:** atp.Status=draft | |
| ***IsService*** | true | |
| ***Variation*** | – | |
| ***Data Elements*** | RsmData | |
| | ***Type*** | CnV2xMsg_RsmRootType |
| | ***Variation*** | – |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.1.5 CnV2xApplRxIndicationSpat

**[CP_SWS_CnV2xMsg_01109]**{DRAFT} ⌈For the CnV2xMsg, an interface CnV2xApplRxIndicationSpat shall be provided as defined below to provide the capability of delivering received SPATs to applications.⌋*()*

**[CP_SWS_CnV2xMsg_01110]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xApplRxIndicationSpat (draft) | |
| ***Comment*** | Deliver received SPATs to Applications<br>**Tags:** atp.Status=draft | |
| ***IsService*** | true | |
| ***Variation*** | – | |
| ***Data Elements*** | SpatData | |
| | ***Type*** | CnV2xMsg_SpatRootType |
| | ***Variation*** | – |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.1.6 CnV2xApplRxIndicationMap

**[CP_SWS_CnV2xMsg_01111]**{DRAFT} ⌈For the CnV2xMsg, an interface CnV2xApplRxIndicationMap shall be provided as defined below to provide the capability of delivering received MAPs to applications.⌋*()*

**[CP_SWS_CnV2xMsg_01112]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xApplRxIndicationMap (draft) | |
| ***Comment*** | Deliver received MAPs to Applications<br>**Tags:** atp.Status=draft | |
| ***IsService*** | true | |
| ***Variation*** | – | |
| ***Data Elements*** | MapData | |
| | ***Type*** | CnV2xMsg_MapRootType |
| | ***Variation*** | – |

⌋*(CP_SRS_CnV2X_00501)*

### 8.7.2 Client-Server-Interfaces

#### 8.7.2.1 CnV2xMsgPoti

**[CP_SWS_CnV2xMsg_01201]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsgPoti (draft) | | |
| ***Comment*** | Interfaces for CnV2xMsg to get and set Position and time in the BSW CNV2X-Stack<br>**Tags:** atp.Status=draft | | |
| ***IsService*** | true | | |
| ***Variation*** | – | | |
| ***Possible Errors*** | 0 | E_OK | Operation successful |
| | 1 | E_NOT_OK | Operation failed |

| | | |
|---|---|---|
| ***Operation*** | GetTime32 | |
| ***Comment*** | Service to get the current reference time | |
| ***Mapped to API*** | CnV2xMsg_GetTime32 | |
| ***Variation*** | – | |
| ***Parameters*** | Time32 | |
| | ***Type*** | uint32 |
| | ***Direction*** | OUT |
| | ***Comment*** | UTC reference time, Timestamp [1 ms] |
| | ***Variation*** | – |
| ***Possible Errors*** | E_OK<br>E_NOT_OK | |

| | | |
|---|---|---|
| ***Operation*** | SetPositionAndTime | |
| ***Comment*** | Service for setting positional and time information relevant for the V2X-Stack | |
| ***Mapped to API*** | CnV2xMsg_SetPositionAndTime | |
| ***Variation*** | – | |
| ***Parameters*** | PositionAndTime | |
| | ***Type*** | CnV2xMsg_PositionAndTimeType |
| | ***Direction*** | IN |
| | ***Comment*** | – |
| | ***Variation*** | – |
| ***Possible Errors*** | E_OK<br>E_NOT_OK | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3 Implementation Data Types

#### 8.7.3.1 BSM Data Element Types

##### 8.7.3.1.1 CnV2xMsg_BrakePedalStatusType

**[CP_SWS_CnV2xMsg_02001]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_BrakePedalStatusType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_<br>BRAKEPEDALSTATUS_<br>UNAVAILABLE | 0x00 | Vehicle brake pedal detector is unavailable |
| | CNV2XMSG_<br>BRAKEPEDALSTATUS_<br>OFF | 0x01 | Vehicle's brake pedal is not pressed |
| | CNV2XMSG_<br>BRAKEPEDALSTATUS_ON | 0x02 | Vehicle's brake pedal is pressed |
| ***Description*** | Enumeration of DE_BrakePedalStatus as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.2 CnV2xMsg_BrakeAppliedStatusType

**[CP_SWS_CnV2xMsg_02002]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_BrakeAppliedStatusType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | Unavailable | 0x10 | Bit 4: When set, the brake applied status is unavailable |
| | bit | Leftfront | 0x08 | Bit 3: left front active |
| | bit | LeftRear | 0x04 | Bit 2: left rear active |
| | bit | RightFront | 0x02 | Bit 1: right front active |
| | bit | RightRear | 0x01 | Bit 0: right rear active |
| ***Description*** | BitString DE_AccelerationControl as defined in CCSA YD/T 3709-2020<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.3 CnV2xMsg_TractionControlStatusType

**[CP_SWS_CnV2xMsg_02003]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_TractionControlStatusType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_TRACTION-CONTROLSTATUS_UNAVAILABLE | 0x00 | Not equipped or unavailable |
| | CNV2XMSG_TRACTION-CONTROLSTATUS_OFF | 0x01 | Traction control is off |
| | CNV2XMSG_TRACTION-CONTROLSTATUS_ON | 0x02 | Traction control is on |
| | CNV2XMSG_TRACTION-CONTROLSTATUS_ENGAGED | 0x03 | Traction control is engaged |
| ***Description*** | Enumeration of DE_TractionControlStatus as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.4 CnV2xMsg_AntiLockBrakeStatusType

**[CP_SWS_CnV2xMsg_02004]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_AntiLockBrakeStatusType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ANTILOCK-BRAKESTATUS_ UNAVAILABLE | 0x00 | Not equipped or unavailable |
| | CNV2XMSG_ANTILOCK-BRAKESTATUS_ OFF | 0x01 | Vehicle's ABS is off |
| | CNV2XMSG_ANTILOCK-BRAKESTATUS_ ON | 0x02 | Vehicle's ABS is on |
| | CNV2XMSG_ANTILOCK-BRAKESTATUS_ ENGAGED | 0x03 | Vehicle's ABS is engaged |
| ***Description*** | Enumeration of DE_AntiLockBrakeStatus as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.5 CnV2xMsg_StabilityControlStatusType

**[CP_SWS_CnV2xMsg_02005]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_StabilityControlStatusType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_STABILITY-CONTROLSTATUS_ UNAVAILABLE | 0x00 | Not equipped or unavailable |
| | CNV2XMSG_STABILITY-CONTROLSTATUS_ OFF | 0x01 | Vehicle's stability control is off |
| | CNV2XMSG_STABILITY-CONTROLSTATUS_ ON | 0x02 | Vehicle's stability control is on |
| | CNV2XMSG_STABILITY-CONTROLSTATUS_ ENGAGED | 0x03 | Vehicle's stability control is engaged |
| ***Description*** | Enumeration of DE_StabilityControlStatus as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.6 CnV2xMsg_BrakeBoostAppliedType

**[CP_SWS_CnV2xMsg_02006]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_BrakeBoostAppliedType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ BRAKEBOOSTAPPLIED_ UNAVAILABLE | 0x00 | Not equipped or unavailable |
| | CNV2XMSG_ BRAKEBOOSTAPPLIED_ OFF | 0x01 | Vehicle's brake boost is off |
| | CNV2XMSG_ BRAKEBOOSTAPPLIED_ ON | 0x02 | Vehicle's brake boost is on |
| ***Description*** | Enumeration of DE_BrakeBoostApplied as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.7 CnV2xMsg_AuxiliaryBrakeStatusType

**[CP_SWS_CnV2xMsg_02007]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_AuxiliaryBrakeStatusType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_AUXILIARY-BRAKESTATUS_ UNAVAILABLE | 0x00 | Not equipped or unavailable |
| | CNV2XMSG_AUXILIARY-BRAKESTATUS_ OFF | 0x01 | Vehicle's AUX brakes is off |
| | CNV2XMSG_AUXILIARY-BRAKESTATUS_ ON | 0x02 | Vehicle's AUX brakes is on |
| | CNV2XMSG_AUXILIARY-BRAKESTATUS_ RESERVED | 0x03 | reserved |
| ***Description*** | Enumeration of DE_AuxiliaryBrakeStatus as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.8 CnV2xMsg_TransmissionStateType

**[CP_SWS_CnV2xMsg_02008]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_TransmissionStateType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ TRANSMISSIONSTATE_ NEUTRAL | 0x00 | Neutral |
| | CNV2XMSG_ TRANSMISSIONSTATE_ PARK | 0x01 | Park |
| | CNV2XMSG_ TRANSMISSIONSTATE_ FORWARDGEARS | 0x02 | Forward gears |
| | CNV2XMSG_ TRANSMISSIONSTATE_ REVERSEGEARS | 0x03 | Reverse gears |
| | CNV2XMSG_ TRANSMISSIONSTATE_ RESERVED1 | 0x04 | Reserved |
| | CNV2XMSG_ TRANSMISSIONSTATE_ RESERVED2 | 0x05 | Reserved |
| | CNV2XMSG_ TRANSMISSIONSTATE_ RESERVED3 | 0x06 | Reserved |
| | CNV2XMSG_ TRANSMISSIONSTATE_ UNAVAILABLE | 0x07 | not-equipped or unavailable value |
| ***Description*** | Enumeration of DE_TransmissionState as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.9 CnV2xMsg_TimeConfidenceType

**[CP_SWS_CnV2xMsg_02009]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_TimeConfidenceType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ TIMECONFIDENCE_ UNAVAILABLE | 0x00 | Not Equipped or unavailable |
| | CNV2XMSG_TIMECONFI-DENCE_100_000 | 0x01 | Better than 100 Seconds |
| | CNV2XMSG_TIMECONFI-DENCE_050_000 | 0x02 | Better than 50 Seconds |

▽

△

| | | | |
|---|---|---|---|
| | CNV2XMSG_TIMECONFIDENCE_020_000 | 0x03 | Better than 20 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_010_000 | 0x04 | Better than 10 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_002_000 | 0x05 | Better than 2 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_001_000 | 0x06 | Better than 1 Second |
| | CNV2XMSG_TIMECONFIDENCE_000_500 | 0x07 | Better than 0.5 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_200 | 0x08 | Better than 0.2 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_100 | 0x09 | Better than 0.1 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_050 | 0x0a | Better than 0.05 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_020 | 0x0b | Better than 0.02 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_010 | 0x0c | Better than 0.01 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_005 | 0x0d | Better than 0.005 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_002 | 0x0e | Better than 0.002 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_001 | 0x0f | Better than 0.001 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_5 | 0x10 | Better than 0.000,5 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_2 | 0x11 | Better than 0.000,2 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_1 | 0x12 | Better than 0.000,1 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_05 | 0x13 | Better than 0.000,05 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_02 | 0x14 | Better than 0.000,02 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_01 | 0x15 | Better than 0.000,01 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_005 | 0x16 | Better than 0.000,005 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_002 | 0x17 | Better than 0.000,002 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_001 | 0x18 | Better than 0.000,001 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_5 | 0x19 | Better than 0.000,000,5 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_2 | 0x1a | Better than 0.000,000,2 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_1 | 0x1b | Better than 0.000,000,1 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_05 | 0x1c | Better than 0.000,000,05 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_02 | 0x1d | Better than 0.000,000,02 Seconds |

▽

△

| | | | |
|---|---|---|---|
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_01 | 0x1e | Better than 0.000,000,01 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_005 | 0x1f | Better than 0.000,000,005 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_002 | 0x20 | Better than 0.000,000,002 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_001 | 0x21 | Better than 0.000,000,001 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_000_5 | 0x22 | Better than 0.000,000,000,5 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_000_2 | 0x23 | Better than 0.000,000,000,2 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_000_1 | 0x24 | Better than 0.000,000,000,1 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_000_05 | 0x25 | Better than 0.000,000,000,05 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_000_02 | 0x26 | Better than 0.000,000,000,02 Seconds |
| | CNV2XMSG_TIMECONFIDENCE_000_000_000_000_01 | 0x27 | Better than 0.000,000,000,01 Seconds |
| ***Description*** | Enumeration of DE_TimeConfidence as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.1.10 CnV2xMsg_GNSSStatusType

**[CP_SWS_CnV2xMsg_02010]**{DRAFT} ⌈

| ***Name*** | CnV2xMsg_GNSSStatusType (draft) | | | |
|---|---|---|---|---|
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | Unavailable | 0x80 | Bit 7: Not Equipped or unavailable |
| | bit | isHealthy | 0x40 | Bit 6: When set, GNSS is healthy |
| | bit | isMonitored | 0x20 | Bit 5: When set, GNSS is monitored |
| | bit | baseStationType | 0x10 | Bit 4: Set to zero if a moving base station |
| | bit | aPDOPofUnder5 | 0x08 | Bit 3: A dilution of precision greater than 5 |
| | bit | inViewOfUnder5 | 0x04 | Bit 2: Less than 5 satellites in view |
| | bit | localCorrectionsPresent | 0x02 | Bit 1: DGPS type corrections used |
| | bit | networkCorrectionsPresent | 0x01 | Bit 0: RTK type corrections used |

▽

△

| | |
|---|---|
| ***Description*** | BitString DE_GNSSStatus as defined in CCSA YD/T 3709-2020<br>**Tags:** atp.Status=draft |
| ***Variation*** | – |
| ***Available via*** | Rte_CnV2xMsg_Type.h |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.11 CnV2xMsg_OffsetLLB12Type

**[CP_SWS_CnV2xMsg_02011]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_OffsetLLB12Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint16 | | |
| ***Range*** | -2048..2047 | – | – |
| ***Description*** | DE_OffsetLL-B12 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.12 CnV2xMsg_OffsetLLB14Type

**[CP_SWS_CnV2xMsg_02012]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_OffsetLLB14Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint16 | | |
| ***Range*** | -8192..8191 | – | – |
| ***Description*** | DE_OffsetLL-B14 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.13 CnV2xMsg_OffsetLLB16Type

**[CP_SWS_CnV2xMsg_02013]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_OffsetLLB16Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint16 | | |
| ***Range*** | -32768..32767 | – | – |
| ***Description*** | DE_OffsetLL-B16 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.14 CnV2xMsg_OffsetLLB18Type

**[CP_SWS_CnV2xMsg_02014]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_OffsetLLB18Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint32 | | |
| ***Range*** | -131072..131071 | – | – |
| ***Description*** | DE_OffsetLL-B18 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.15 CnV2xMsg_OffsetLLB22Type

**[CP_SWS_CnV2xMsg_02015]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_OffsetLLB22Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint32 | | |
| ***Range*** | -2097152..2097151 | – | – |
| ***Description*** | DE_OffsetLL-B22 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.16 CnV2xMsg_OffsetLLB24Type

**[CP_SWS_CnV2xMsg_02016]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_OffsetLLB24Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint32 | | |
| ***Range*** | -8388608..8388607 | – | – |
| ***Description*** | DE_OffsetLL-B24 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.17 CnV2xMsg_LongitudeType

**[CP_SWS_CnV2xMsg_02017]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_LongtitudeType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint32 | | |
| ***Range*** | -1799999999..1800000001 | – | – |
| ***Description*** | 1/10 micro degree; The value 1800000001 shall be used for invalid; DE_Longtitude as defined in CCSA YD/T 3709-2020;<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.18 CnV2xMsg_LatitudeType

**[CP_SWS_CnV2xMsg_02018]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_LatitudeType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint32 | | |
| ***Range*** | -900000000..900000001 | – | – |
| ***Description*** | 1/10 micro degree. The value 900000001 shall be used for invalid; DE_Latitude as defined in CCSA YD/T 3709-2020<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.19 CnV2xMsg_VerOffsetB07Type

**[CP_SWS_CnV2xMsg_02019]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_VerOffsetB07Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint8 | | |
| ***Range*** | -64..63 | – | – |
| ***Description*** | DE_VertOffset-B07 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.20 CnV2xMsg_VerOffsetB08Type

**[CP_SWS_CnV2xMsg_02020]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_VerOffsetB08Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint8 | | |
| ***Range*** | -128..127 | – | – |
| ***Description*** | DE_VertOffset-B08 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.21 CnV2xMsg_VerOffsetB09Type

**[CP_SWS_CnV2xMsg_02021]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_VerOffsetB09Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint16 | | |
| ***Range*** | -256..255 | – | – |
| ***Description*** | DE_VertOffset-B09 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.22 CnV2xMsg_VerOffsetB10Type

**[CP_SWS_CnV2xMsg_02022]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_VerOffsetB10Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint16 | | |
| ***Range*** | -512..511 | – | – |
| ***Description*** | DE_VertOffset-B10 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.23 CnV2xMsg_VerOffsetB11Type

**[CP_SWS_CnV2xMsg_02023]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_VerOffsetB11Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint16 | | |
| ***Range*** | -1024..1023 | – | – |
| ***Description*** | DE_VertOffset-B11 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.24 CnV2xMsg_VerOffsetB12Type

**[CP_SWS_CnV2xMsg_02024]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_VerOffsetB12Type (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | sint16 | | |
| ***Range*** | -2048..2047 | – | – |
| ***Description*** | DE_VertOffset-B12 as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.25 CnV2xMsg_ResponseTypeType

**[CP_SWS_CnV2xMsg_02025]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_ResponseTypeType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ RESPONSETYPE_ UNAVAILABLE | 0x00 | Not In Use Or Not Equipped |
| | CNV2XMSG_ RESPONSETYPE_ EMERGENCY | 0x01 | active service call at emergency level |
| | CNV2XMSG_ RESPONSETYPE_ NONEMERGENCY | 0x02 | also used when returning from service call |
| | CNV2XMSG_ RESPONSETYPE_ PURSUIT | 0x03 | sender driving may be erratic |
| | CNV2XMSG_ RESPONSETYPE_ STATIONARY | 0x04 | sender is not moving, stopped along roadside |
| | CNV2XMSG_ RESPONSETYPE_ SLOWMOVING | 0x05 | such as a litter trucks, etc |
| | CNV2XMSG_ RESPONSETYPE_ STOPANDGOMOVEMENT | 0x06 | such as school bus or garbage truck |
| ***Description*** | Enumeration of DE_ResponseType as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋*(CP_SRS_CnV2X_00501)*

#### 8.7.3.1.26 CnV2xMsg_SirenInUseType

**[CP_SWS_CnV2xMsg_02026]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_SirenInUseType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ SIRENINUSE_ UNAVAILABLE | 0x00 | Unavailable or not equipped |
| | CNV2XMSG_ SIRENINUSE_NOTINUSE | 0x01 | Not in use |
| | CNV2XMSG_ SIRENINUSE_INUSE | 0x02 | In use |
| | CNV2XMSG_ SIRENINUSE_RESERVED | 0x03 | For future use |

▽

△

| | |
|---|---|
| ***Description*** | Enumeration of DE_SirenInUse as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft |
| ***Variation*** | – |
| ***Available via*** | Rte_CnV2xMsg_Type.h |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.27 CnV2xMsg_LightbarInUseType

**[CP_SWS_CnV2xMsg_02027]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_LightbarInUseType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_<br>LIGHTBARINUSE_<br>UNAVAILABLE | 0x00 | Unavailable or not equipped |
| | CNV2XMSG_<br>LIGHTBARINUSE_<br>NOTINUSE | 0x01 | None active |
| | CNV2XMSG_<br>LIGHTBARINUSE_INUSE | 0x02 | In use |
| | CNV2XMSG_<br>LIGHTBARINUSE_<br>YELLOWCAUTIONLIGHTS | 0x03 | Yellow caution lights |
| | CNV2XMSG_<br>LIGHTBARINUSE_<br>SCHOOLBUSLIGHTS | 0x04 | School bus lights |
| | CNV2XMSG_<br>LIGHTBARINUSE_<br>ARROWSIGNSACTIVE | 0x05 | Arrow signs active |
| | CNV2XMSG_<br>LIGHTBARINUSE_<br>SLOWMOVINGVEHICLE | 0x06 | Slow moving vehicle |
| | CNV2XMSG_<br>LIGHTBARINUSE_<br>FREQSTOPS | 0x07 | Frequent stops |
| ***Description*** | Enumeration of DE_LightbarInUse as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.28 CnV2xMsg_VehicleEventFlagsType

**[CP_SWS_CnV2xMsg_02028]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_VehicleEventFlagsType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint16 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | eventHazardLights | 0x1000 | Bit 12: Hazard Lights |
| | bit | eventStopLineViolation | 0x800 | Bit 11: Stop Line Violation |
| | bit | eventABSactivated | 0x400 | Bit 10: ABS activated |
| | bit | eventTractionControlLoss | 0x200 | Bit 9: Traction Control |
| | bit | eventStabilityControlactivated | 0x100 | Bit 8: Stability Control |
| | bit | eventHazardousMaterials | 0x80 | Bit 7: Hazardous Materials |
| | bit | eventReserved1 | 0x40 | Bit 6: Reserved |
| | bit | eventHardBraking | 0x20 | Bit 5: Hard Braking |
| | bit | eventLightsChanged | 0x10 | Bit 4: Lights Changed |
| | bit | eventWipersChanged | 0x08 | Bit 3: Wipers Changed |
| | bit | eventFlatTire | 0x04 | Bit 2: Flat tire |
| | bit | eventDisabledVehicle | 0x02 | Bit 1: Disabled Vehicle |
| | bit | eventAirBagDeployment | 0x01 | Bit 0: Air Bag Deploymen |
| ***Description*** | BitString DE_VehicleEventFlags as defined in CCSA YD/T 3709-2020<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋*(CP_SRS_CnV2X_00501)*

#### 8.7.3.1.29 CnV2xMsg_ExteriorLightsType

**[CP_SWS_CnV2xMsg_02029]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_ExteriorLightsType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint16 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | lowBeamHeadlightsOn | 0x100 | Bit 8: lowBeamHeadlightsOn |
| | bit | highBeamHeadlightsOn | 0x80 | Bit 7: highBeamHeadlightsOn |
| | bit | leftTurnSignalOn | 0x40 | Bit 6: leftTurnSignalOn |
| | bit | rightTurnSignalOn | 0x20 | Bit 5: rightTurnSignalOn |
| | bit | hazardSignalOn | 0x10 | Bit 4: hazardSignalOn |
| | bit | automaticLightControlOn | 0x08 | Bit 3: automaticLightControlOn |
| | bit | daytimeRunningLightsOn | 0x04 | Bit 2: daytimeRunningLightsOn |
| | bit | fogLightOn | 0x02 | Bit 1: fogLightOn |
| | bit | parkingLightsOn | 0x01 | Bit 0: parkingLightsOn |

▽

△

| | |
|---|---|
| ***Description*** | BitString DE_ExteriorLights as defined in CCSA YD/T 3709-2020<br>**Tags:** atp.Status=draft |
| ***Variation*** | – |
| ***Available via*** | Rte_CnV2xMsg_Type.h |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.1.30 CnV2xMsg_BasicVehicleClassType

**[CP_SWS_CnV2xMsg_02030]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_BasicVehicleClassType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_VC_UNKOWN | 0x0 | Not known or unavailable |
| | CNV2XMSG_VC_SPECIAL | 0x01 | Special Vehicle including specical bus, special purpose passenger car, motor caravan, armoured passenger car, hearse, special operating vehicle, special goods vehicle. |
| | CNV2XMSG_VC_ PASSENGER | 0x0A | Passenger cars,including saloon, convertible sallon, pullman saloon, coupe, convertible, hatchback, station wagon, multipurpose passenger car, forward control passenger car and off-road passenger car |
| | CNV2XMSG_VC_GOODS_ LIGHT | 0x14 | Light goods vehicle |
| | CNV2XMSG_VC_GOODS_ SEMITRAILER | 0x19 | Semi-trailer towing vehicle |
| | CNV2XMSG_VC_BUS | 0x32 | Basic Bus type, including minibus, city-bus, interurban coach, articulated bus, trolly bus and off-road bus |
| | CNV2XMSG_VC_EM_ FIRETRUCK_LIGHT | 0x3E | Emergency vehicle: Light fire truck |
| | CNV2XMSG_VC_EM_ FIRETRUCK_HEAVY | 0x3F | Emergency vehicle: Heavy fire truck |
| | CNV2XMSG_VC_EM_ NURSING | 0x40 | Emergency vehicle: Nursing car |
| | CNV2XMSG_VC_EM_ AMBULANCE | 0x41 | Emergency vehicle: ambulence |
| | CNV2XMSG_VC_EM_ POLICE_LIGHT | 0x42 | Emergency vehicle: Light police car |
| | CNV2XMSG_VC_EM_ POLICE_HEAVY | 0x43 | Emergency vehicle: Heavy police car |
| | CNV2XMSG_VC_EM_ ENGINEERING | 0x44 | Emergency vehicle: Engineering vehicle |

▽

△

| | |
|---|---|
| ***Description*** | Integer of DE_BasicVehicleClass see "GB/T Technical Requirements of Vehicular Communication System based on LTE-V2X Direct Communication"<br>**Tags:** atp.Status=draft |
| ***Variation*** | – |
| ***Available via*** | Rte_CnV2xMsg_Type.h |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.31 CnV2xMsg_VehicleIDType

**[CP_SWS_CnV2xMsg_02032]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_VehicleIDType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Values | |
| | ***Type*** | Array of uint8 |
| | ***Size*** | 8 |
| | ***Comment*** | – |
| ***Description*** | Vehicle ID as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.32 CnV2xMsg_PositionConfidenceType

**[CP_SWS_CnV2xMsg_02033]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_PositionConfidenceType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ POSITIONCONFIDENCE_ POS_UNAVAILABLE | 0x00 | Not equipment or unavailable |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_500_00 | 0x01 | the position accuracy is equal to or less than 500 meter |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_200_00 | 0x02 | the position accuracy is equal to or less than 200 meter |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_100_00 | 0x03 | the position accuracy is equal to or less than 100 meter |

▽

△

| | | | |
|---|---|---|---|
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_050_00 | 0x04 | the position accuracy is equal to or less than 50 meter |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_020_00 | 0x05 | the position accuracy is equal to or less than 20 meter |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_010_00 | 0x06 | the position accuracy is equal to or less than 10 meter |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_005_00 | 0x07 | the position accuracy is equal to or less than 5 meters |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_002_00 | 0x08 | the position accuracy is equal to or less than 2 meters |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_001_00 | 0x09 | the position accuracy is equal to or less than 1 meters |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_000_50 | 0x0a | the position accuracy is equal to or less than 0.5 meters |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_000_20 | 0x0b | the position accuracy is equal to or less than 0.2 meters |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_000_10 | 0x0c | the position accuracy is equal to or less than 0.1 meters |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_000_05 | 0x0d | the position accuracy is equal to or less than 0.05 meters |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_000_02 | 0x0e | he position accuracy is equal to or less than 0.02 meters |
| | CNV2XMSG_ POSITIONCONFIDENCE_ POS_000_01 | 0x0f | he position accuracy is equal to or less than 0.01 meters |
| ***Description*** | Enumeration of DE_PositionConfidence as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.1.33 CnV2xMsg_ElevationConfidenceType

**[CP_SWS_CnV2xMsg_02034]**{DRAFT} ⌈

| | |
|---|---|
| ***Name*** | CnV2xMsg_ElevationConfidenceType (draft) |
| ***Kind*** | Type |
| ***Derived from*** | uint8 |

▽

△

| | | | |
|---|---|---|---|
| ***Range*** | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_UNAVAILABLE | 0x00 | Not equipment or unavailable |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_500_00 | 0x01 | the elevation accuracy is equal to or less than 500 meter |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_200_00 | 0x02 | the elevation accuracy is equal to or less than 200 meter |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_100_00 | 0x03 | the elevation accuracy is equal to or less than 100 meter |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_050_00 | 0x04 | the elevation accuracy is equal to or less than 50 meter |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_020_00 | 0x05 | the elevation accuracy is equal to or less than 20 meter |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_010_00 | 0x06 | the elevation accuracy is equal to or less than 10 meter |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_005_00 | 0x07 | the elevation accuracy is equal to or less than 5 meters |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_002_00 | 0x08 | the elevation accuracy is equal to or less than 2 meters |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_001_00 | 0x09 | the elevation accuracy is equal to or less than 1 meters |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_000_50 | 0x0a | the elevation accuracy is equal to or less than 0.5 meters |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_000_20 | 0x0b | the elevation accuracy is equal to or less than 0.2 meters |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_000_10 | 0x0c | the elevation accuracy is equal to or less than 0.1 meters |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_000_05 | 0x0d | the elevation accuracy is equal to or less than 0.05 meters |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_000_02 | 0x0e | the elevation accuracy is equal to or less than 0.02 meters |
| | CNV2XMSG_ ELEVATIONCONFIDENCE_ ALT_000_01 | 0x0f | the elevation accuracy is equal to or less than 0.01 meters |
| ***Description*** | Enumeration of DE_ElevationConfidence as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.34 CnV2xMsg_SpeedConfidenceType

**[CP_SWS_CnV2xMsg_02035]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_SpeedConfidenceType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ SPEEDCONFIDENCE_ SPD_UNAVAILABLE | 0x00 | Not equipment or unavailable |
| | CNV2XMSG_ SPEEDCONFIDENCE_ SPD_100_00 | 0x01 | the speed accuracy is equal to or less than 100 meter / sec |
| | CNV2XMSG_ SPEEDCONFIDENCE_ SPD_010_00 | 0x02 | the speed accuracy is equal to or less than 10 meter /sec |
| | CNV2XMSG_ SPEEDCONFIDENCE_ SPD_005_00 | 0x03 | the speed accuracy is equal to or less than 5 meter /sec |
| | CNV2XMSG_ SPEEDCONFIDENCE_ SPD_001_00 | 0x04 | the speed accuracy is equal to or less than 1 meter /sec |
| | CNV2XMSG_ SPEEDCONFIDENCE_ SPD_000_10 | 0x05 | the speed accuracy is equal to or less than 0.1 meter /sec |
| | CNV2XMSG_ SPEEDCONFIDENCE_ SPD_000_05 | 0x06 | the speed accuracy is equal to or less than 0.05 meter /sec |
| | CNV2XMSG_ SPEEDCONFIDENCE_ SPD_000_01 | 0x07 | the speed accuracy is equal to or less than 0.01 meters /sec |
| ***Description*** | Enumeration of DE_SpeedConfidence as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.35 CnV2xMsg_HeadingConfidenceType

**[CP_SWS_CnV2xMsg_02036]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_HeadingConfidenceType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ HEADINGCONFIDENCE_ HEAD_UNAVAILABLE | 0x00 | Not equipment or unavailable |
| | CNV2XMSG_ HEADINGCONFIDENCE_ HEAD_10_0000 | 0x01 | the heading accuracy is equal to or less than 10 degree |

▽

△

| | | | |
|---|---|---|---|
| | CNV2XMSG_ HEADINGCONFIDENCE_ HEAD_05_0000 | 0x02 | the heading accuracy is equal to or less than 5 degree |
| | CNV2XMSG_ HEADINGCONFIDENCE_ HEAD_01_0000 | 0x03 | the heading accuracy is equal to or less than 1 degree |
| | CNV2XMSG_ HEADINGCONFIDENCE_ HEAD_00_1000 | 0x04 | the heading accuracy is equal to or less than 0.1 degree |
| | CNV2XMSG_ HEADINGCONFIDENCE_ HEAD_00_0500 | 0x05 | the heading accuracy is equal to or less than 0.05 degree |
| | CNV2XMSG_ HEADINGCONFIDENCE_ HEAD_00_0100 | 0x06 | the heading accuracy is equal to or less than 0.01 degree |
| | CNV2XMSG_ HEADINGCONFIDENCE_ HEAD_00_0125 | 0x07 | the heading accuracy is equal to or less than 0.0125 degree |
| ***Description*** | Enumeration of DE_HeadingConfidence as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.1.36 CnV2xMsg_SteeringWheelAngleConfidenceType

**[CP_SWS_CnV2xMsg_02037]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_SteeringWheelAngleConfidenceType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | CNV2XMSG_ STEERINGWHEELANGLE-CONFIDENCE_SWA_ UNAVAILABLE | 0x00 | Not equipment or unavailable |
| | CNV2XMSG_ STEERINGWHEELANGLE-CONFIDENCE_ SWA_2_00 | 0x01 | the steering wheel angle accuracy is equal to or less than 2 degree |
| | CNV2XMSG_ STEERINGWHEELANGLE-CONFIDENCE_ SWA_1_00 | 0x02 | the steering wheel angle accuracy is equal to or less than 1 degree |
| | CNV2XMSG_ STEERINGWHEELANGLE-CONFIDENCE_ SWA_0_02 | 0x03 | the steering wheel angle accuracy is equal to or less than 0.02 degree |
| ***Description*** | Enumeration of DE_SteeringWheelAngleConfidence as defined in CCSA YD/T 3709-2020.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.1.37 CnV2xMsg_FuelType

**[CP_SWS_CnV2xMsg_02038]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_FuelType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint8 | | |
| ***Range*** | 0..15 | – | – |
| ***Description*** | unknownFuel FuelType::= 0<br>gasoline FuelType::= 1 – Gasoline Powered<br>ethanol FuelType::= 2 – Including blends<br>diesel FuelType::= 3 – All types<br>electric FuelType::= 4<br>hybrid FuelType::= 5 – All types<br>hydrogen FuelType::= 6<br>natGasLiquid FuelType::= 7 – Liquefied<br>natGasComp FuelType::= 8 – Compressed<br>propane FuelType::= 9<br>as defined in CCSA YD/T 3709-2020.<br><br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2 BSM Data Frame Types

#### 8.7.3.2.1 CnV2xMsg_Position3DType

**[CP_SWS_CnV2xMsg_02101]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_Position3DType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_Position3DPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | Latitude | |
| | ***Type*** | sint32 |
| | ***Comment*** | Latitude of the geographical point, 1/10 micro degree. Range: -900000000..900000001; The value 900000001 shall be used for invalid; |
| | Longtitude | |
| | ***Type*** | sint32 |
| | ***Comment*** | Longtitude of the geographical point, 1/10 micro degree. Range: -1799999999..1800000001; The value 1800000001 shall be used for invalid |
| | Elevation | |
| | ***Type*** | sint32 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Elevation of the geographical point, in units of 10 cm steps above or below the reference ellipsoid. Range: -4096..61439 |
| ***Description*** | DF_Position3D as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.2 CnV2xMsg_Position3DPresenceType

**[CP_SWS_CnV2xMsg_02140]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_Position3DPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | Elevation | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_Position3DType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.3 CnV2xMsg_PositionAccuracyType

**[CP_SWS_CnV2xMsg_02102]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionAccuracyType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | SemiMajorAxisAccuracy | |
| | ***Type*** | uint8 |
| | ***Comment*** | semi-major axis accuracy at one standard dev; Range: 0..255 (0-12.7 meter) Value 254: any value equal or greater than 12.70 meter ; Value 255: unavailable semi-major axis value |
| | SemiMinorAxisAccuracy | |
| | ***Type*** | uint8 |
| | ***Comment*** | semi-minor axis accuracy at one standard dev; Range: 0..255 (0-12.7 meter) Value 254: any value equal or greater than 12.70 meter ; Value 255: unavailable semi-major axis |
| | SemiMajorAxisOrientation | |
| | ***Type*** | uint16 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Orientation of semi-major axis ; Units of 360/65535 deg = 0.0054932479; Range: 0..65536 – a value of 0 shall be 0 degrees – a value of 1 shall be 0.0054932479 degrees – a value of 65534 shall be 359.9945078786 deg – a value of 65535 shall be used for orientation unavailable |
| ***Description*** | DF_PositionAccuracy as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.4 CnV2xMsg_PositionConfidenceSetType

**[CP_SWS_CnV2xMsg_02103]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionConfidenceSetType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_PositionConfidenceSetPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | PositionConfidence | |
| | ***Type*** | CnV2xMsg_PositionConfidenceType |
| | ***Comment*** | Absolute accuracy of a reported latitude and longtitude value |
| | Elevationconfidence | |
| | ***Type*** | CnV2xMsg_ElevationConfidenceType |
| | ***Comment*** | Absolute accuracy of a reported elevation value |
| ***Description*** | DF_PositionConfidenceSet as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.5 CnV2xMsg_PositionConfidenceSetPresenceType

**[CP_SWS_CnV2xMsg_02141]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_PositionConfidenceSetPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |

▽

△

| | | | | |
|---|---|---|---|---|
| | bit | ElevationConfidence | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_PositionConfidenceSetType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.6 CnV2xMsg_MotionConfidenceSetType

**[CP_SWS_CnV2xMsg_02104]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_MotionConfidenceSetType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_MotionConfidenceSetType |
| | ***Comment*** | Mark optional childs present or not |
| | SpeedConfidence | |
| | ***Type*** | CnV2xMsg_SpeedConfidenceType |
| | ***Comment*** | Absolute accuracy of speed value |
| | HeadingConfidence | |
| | ***Type*** | CnV2xMsg_HeadingConfidenceType |
| | ***Comment*** | Absolute accuracy of Heading value |
| | SteeringWheelAngleConfidence | |
| | ***Type*** | CnV2xMsg_SteeringWheelAngleConfidenceType |
| | ***Comment*** | Absolute accuracy of steering wheelAngle value |
| ***Description*** | DF_MotionConfidenceSet as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.7 CnV2xMsg_MotionConfidenceSetPresenceType

**[CP_SWS_CnV2xMsg_02142]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_MotionConfidenceSetPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |

▽

△

| | | | | |
|---|---|---|---|---|
| | bit | SteeringWheelAngleConfidence | 0x04 | Bit 2: Optional child present |
| | bit | HeadingConfidence | 0x02 | Bit 1: Optional child present |
| | bit | SpeedConfidence | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_MotionConfidenceSetType<br>**Tags:** atp.Status=draft | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.8 CnV2xMsg_AccelerationSet4WayType

**[CP_SWS_CnV2xMsg_02105]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_AccelerationSet4WayType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | LongAcceleration | |
| | ***Type*** | sint16 |
| | ***Comment*** | acceleration at longitudinal direction,<br>– LSB units are 0.01 m/sˆ2<br>– the value 2000 shall be used for values greater than 2000<br>– the value -2000 shall be used for values less than -2000<br>– a value of 2001 shall be used for Unavailable<br><br>Range: -2000..2001 |
| | LatAcceleration | |
| | ***Type*** | sint16 |
| | ***Comment*** | acceleration at latitude direction<br>– LSB units are 0.01 m/sˆ2<br>– the value 2000 shall be used for values greater than 2000<br>– the value -2000 shall be used for values less than -2000<br>– a value of 2001 shall be used for Unavailable<br>Range: -2000..2001 |
| | VerticalAcceleration | |
| | ***Type*** | sint8 |
| | ***Comment*** | Vehicle acceleration at vertical direction<br>– LSB units of 0.02 G steps over -2.52 to +2.54 G<br>– The value +127 shall be used for ranges >= 2.54 G<br>– The value -126 shall be used for ranges <= 2.52 G<br>– The value -127 shall be used for unavailable<br>Rang: -127..127 |
| | YawRate | |
| | ***Type*** | sint16 |
| | ***Comment*** | rotation around z-axis, LSB units of 0.01 degrees per second (signed)<br>Range: -32767..32767 |
| ***Description*** | DF_AccelerationSet4Way as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.9 CnV2xMsg_BrakeSystemStatusType

**[CP_SWS_CnV2xMsg_02107]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_BrakeSystemStatusType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_BrakeSystemStatusPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | BrakePedalStatus | |
| | ***Type*** | CnV2xMsg_BrakePedalStatusType |
| | ***Comment*** | Indicate the Vehicle pedal status |
| | BrakeAppliedStatus | |
| | ***Type*** | CnV2xMsg_BrakeAppliedStatusType |
| | ***Comment*** | Indicate the vehicle multiple brakes status |
| | TractionControlStatus | |
| | ***Type*** | CnV2xMsg_TractionControlStatusType |
| | ***Comment*** | Indicate vehicle traction control status |
| | AntiLockBrakeStatus | |
| | ***Type*** | CnV2xMsg_AntiLockBrakeStatusType |
| | ***Comment*** | Indicate vehicle ABS status |
| | StabilityControlStatus | |
| | ***Type*** | CnV2xMsg_StabilityControlStatusType |
| | ***Comment*** | Indicate stability control status |
| | BrakeBoostApplied | |
| | ***Type*** | CnV2xMsg_BrakeBoostAppliedType |
| | ***Comment*** | Indicate vehicle brake boost status |
| | AuxiliaryBrakeStatus | |
| | ***Type*** | CnV2xMsg_AuxiliaryBrakeStatusType |
| | ***Comment*** | Indicate auxiliary brake status |
| ***Description*** | DF_BrakeSystemStatus as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.10 CnV2xMsg_BrakeSystemStatusPresenceType

**[CP_SWS_CnV2xMsg_02108]**{DRAFT} ⌈

| | |
|---|---|
| ***Name*** | CnV2xMsg_BrakeSystemStatusPresenceType (draft) |
| ***Kind*** | Bitfield |
| ***Derived from*** | uint8 |

▽

△

| Elements | Kind | Name | Mask | Description |
|---|---|---|---|---|
| | bit | AntiLockBrakeStatus | 0x08 | Bit 3: Optional child present |
| | bit | StabilityControlStatus | 0x04 | Bit 2: Optional child present |
| | bit | BrakeBoostApplied | 0x02 | Bit 1: Optional child present |
| | bit | AuxiliaryBrakeStatus | 0x01 | Bit 0 (LSB): Optional child present |
| **Description** | Presence flags for CnV2xMsg_BrakeSystemStatusType<br>**Tags:** atp.Status=draft | | | |
| **Variation** | – | | | |
| **Available via** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.11 CnV2xMsg_VehicleSizeType

**[CP_SWS_CnV2xMsg_02109]**{DRAFT} ⌈

| | | |
|---|---|---|
| **Name** | CnV2xMsg_VehicleSizeType (draft) | |
| **Kind** | Structure | |
| **Elements** | Presence | |
| | **Type** | CnV2xMsg_VehicleSizePresenceType |
| | **Comment** | Mark optional childs present or not |
| | VehicleWidth | |
| | **Type** | uint16 |
| | **Comment** | Vehicle width, LSB units are 1 cm Range: 0..1023 |
| | VehicleLength | |
| | **Type** | uint16 |
| | **Comment** | Vehicle length, LSB units of 1 cm Range: 0..4095 |
| | VehicleHeight | |
| | **Type** | uint8 |
| | **Comment** | Vehicle height, LSB units of 5 cm Range: 0..127 |
| **Description** | DF_VehicleSize as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| **Variation** | – | |
| **Available via** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.12 CnV2xMsg_VehicleSizePresenceType

**[CP_SWS_CnV2xMsg_02110]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_VehicleSizePresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | VehicleHeight | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_VehicleSizeType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.13 CnV2xMsg_VehicleClassificationType

**[CP_SWS_CnV2xMsg_02111]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_VehicleClassificationType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_VehicleClassificationPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | BasicVehicleClass | |
| | ***Type*** | CnV2xMsg_BasicVehicleClassType |
| | ***Comment*** | Vehicle basic type |
| | FuelType | |
| | ***Type*** | CnV2xMsg_FuelType |
| | ***Comment*** | Vehicle fule type |
| ***Description*** | DF_VehicleClassification as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.14 CnV2xMsg_VehicleClassificationPresenceType

**[CP_SWS_CnV2xMsg_02112]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_VehicleClassificationPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | FuelType | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_VehicleClassificationType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.15 CnV2xMsg_DDateTimeType

**[CP_SWS_CnV2xMsg_02113]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_DDateTimeType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_DDateTimePresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | DYear | |
| | ***Type*** | uint16 |
| | ***Comment*** | Indicate calendar year, 0 indicate unkown Range: 0..4095 |
| | DMonth | |
| | ***Type*** | uint8 |
| | ***Comment*** | Indicate months of a year, 0 indicate unkown Range: 0..12 |
| | DDay | |
| | ***Type*** | uint8 |
| | ***Comment*** | Indicate Days of a month, 0 indicate unkown Range:0..31 |
| | DHour | |
| | ***Type*** | uint8 |
| | ***Comment*** | Indicate hours in a day, =24 present unkonwn Range:0..31 |
| | DMinute | |
| | ***Type*** | uint8 |
| | ***Comment*** | Indicate minutes in one hour, 60 present unkown Range: 0..60 |
| | DSecond | |
| | ***Type*** | uint16 |
| | ***Comment*** | unit: millisecond, indicate milliseconds in a minute, =60000 present unknown Range: 0..65536 |
| | DTimeoffset | |

▽

△

| | | |
|---|---|---|
| | ***Type*** | sint16 |
| | ***Comment*** | Indicates the minute difference from UTC time Range: -840..840 |
| ***Description*** | DF_DDateTime as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.16 CnV2xMsg_DDateTimePresenceType

**[CP_SWS_CnV2xMsg_02144]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_DDateTimePresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | DYear | 0x40 | Bit 6: Optional child present |
| | bit | DMonth | 0x20 | Bit 5: Optional child present |
| | bit | DDay | 0x10 | Bit 4: Optional child present |
| | bit | DHour | 0x08 | Bit 3: Optional child present |
| | bit | DMinute | 0x04 | Bit 2: Optional child present |
| | bit | DSecond | 0x02 | Bit 1: Optional child present |
| | bit | DTimeOffset | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_DDateTimeType<br>**Tags:** atp.Status=draft | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.17 CnV2xMsg_PositionOffsetLL24BType

**[CP_SWS_CnV2xMsg_02114]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLL24BType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Lon | |
| | ***Type*** | CnV2xMsg_OffsetLLB12Type |
| | ***Comment*** | 12-bit value indicating latitude and longitude deviation |
| | Lat | |
| | ***Type*** | CnV2xMsg_OffsetLLB12Type |
| | ***Comment*** | 12-bit value indicating latitude and longitude deviation |

▽

△

| | |
|---|---|
| ***Description*** | DF_PositionOffset-LL-24B as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft |
| ***Available via*** | Rte_CnV2xMsg_Type.h |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.18 CnV2xMsg_PositionOffsetLL28BType

**[CP_SWS_CnV2xMsg_02115]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLL28BType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Lon | |
| | ***Type*** | CnV2xMsg_OffsetLLB14Type |
| | ***Comment*** | 14-bit value indicating latitude and longitude deviation |
| | Lat | |
| | ***Type*** | CnV2xMsg_OffsetLLB14Type |
| | ***Comment*** | 14-bit value indicating latitude and longtitude deviation |
| ***Description*** | DF_PositionOffset-LL-28B as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.19 CnV2xMsg_PositionOffsetLL32BType

**[CP_SWS_CnV2xMsg_02116]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLL32BType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Lon | |
| | ***Type*** | CnV2xMsg_OffsetLLB16Type |
| | ***Comment*** | 16-bit value indicating latitude and longitude deviation |
| | Lat | |
| | ***Type*** | CnV2xMsg_OffsetLLB16Type |
| | ***Comment*** | 16-bit value indicating latitude and longitude deviation |
| ***Description*** | DF_PositionOffset-LL-32B as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.20 CnV2xMsg_PositionOffsetLL36BType

**[CP_SWS_CnV2xMsg_02117]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLL36BType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Lon | |
| | ***Type*** | CnV2xMsg_OffsetLLB18Type |
| | ***Comment*** | 18-bit value indicating latitude and longitude deviation |
| | Lat | |
| | ***Type*** | CnV2xMsg_OffsetLLB18Type |
| | ***Comment*** | 18-bit value indicating latitude and longitude deviation |
| ***Description*** | DF_PositionOffset-LL-36B as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.21 CnV2xMsg_PositionOffsetLL44BType

**[CP_SWS_CnV2xMsg_02118]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLL44BType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Lon | |
| | ***Type*** | CnV2xMsg_OffsetLLB22Type |
| | ***Comment*** | 22-bit value indicating latitude and longitude deviation |
| | Lat | |
| | ***Type*** | CnV2xMsg_OffsetLLB22Type |
| | ***Comment*** | 22-bit value indicating latitude and longitude deviation |
| ***Description*** | DF_PositionOffset-LL-44B as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.22 CnV2xMsg_PositionOffsetLL48BType

**[CP_SWS_CnV2xMsg_02119]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLL48BType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Lon | |
| | ***Type*** | CnV2xMsg_OffsetLLB24Type |
| | ***Comment*** | 24-bit value indicating latitude and longitude deviation |
| | Lat | |
| | ***Type*** | CnV2xMsg_OffsetLLB24Type |
| | ***Comment*** | 24-bit value indicating latitude and longitude deviation |
| ***Description*** | DF_PositionOffset-LL-48B as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.23 CnV2xMsg_PositionOffsetLL64BType

**[CP_SWS_CnV2xMsg_02120]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLL64BType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Lon | |
| | ***Type*** | CnV2xMsg_LongtitudeType |
| | ***Comment*** | 32-bit value indicating latitude and longitude deviation |
| | Lat | |
| | ***Type*** | CnV2xMsg_LatitudeType |
| | ***Comment*** | 32-bit value indicating latitude and longitude deviation |
| ***Description*** | DF_PositionOffset-LL-64B as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.24 CnV2xMsg_PositionOffsetLLType

**[CP_SWS_CnV2xMsg_02121]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLLType | |
| ***Kind*** | Union | |
| ***Elements*** | PositionLL24B | |
| | ***Type*** | CnV2xMsg_PositionOffsetLL24BType |
| | ***Comment*** | 12-bit value indicating latitude and longitude deviation |
| | PositionLL28B | |
| | ***Type*** | CnV2xMsg_PositionOffsetLL28BType |
| | ***Comment*** | 14-bit value indicating latitude and longitude deviation |
| | PositionLL32B | |
| | ***Type*** | CnV2xMsg_PositionOffsetLL32BType |
| | ***Comment*** | 16-bit value indicating latitude and longitude deviation |
| | PositionLL36B | |
| | ***Type*** | CnV2xMsg_PositionOffsetLL36BType |
| | ***Comment*** | 18-bit value indicating latitude and longitude deviation |
| | PositionLL44B | |
| | ***Type*** | CnV2xMsg_PositionOffsetLL44BType |
| | ***Comment*** | 22-bit value indicating latitude and longitude deviation |
| | PositionLL48B | |
| | ***Type*** | CnV2xMsg_PositionOffsetLL48BType |
| | ***Comment*** | 24-bit value indicating latitude and longtitude deviation |
| | PositionLL64B | |
| | ***Type*** | CnV2xMsg_PositionOffsetLL64BType |
| | ***Comment*** | 32-bit value indicating latitude and longtitude deviation |
| ***Description*** | DF_PositionOffsetLL as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document. | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.25 CnV2xMsg_VertcalOffsetType

**[CP_SWS_CnV2xMsg_02122]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_VeritcalOffsetType | |
| ***Kind*** | Union | |
| ***Elements*** | VerOffsetB07 | |
| | ***Type*** | CnV2xMsg_VerOffsetB07Type |
| | ***Comment*** | 7-bit value indicating vertical deviation |
| | VerOffsetB08 | |
| | ***Type*** | CnV2xMsg_VerOffsetB08Type |
| | ***Comment*** | 8-bit value indicating vertical deviation |
| | VerOffsetB09 | |

▽

△

| | | |
|---|---|---|
| | ***Type*** | CnV2xMsg_VerOffsetB09Type |
| | ***Comment*** | 9-bit value indicating vertical deviation |
| | VerOffsetB10 | |
| | ***Type*** | CnV2xMsg_VerOffsetB10Type |
| | ***Comment*** | 10-bit value indicating vertical deviation |
| | VerOffsetB11 | |
| | ***Type*** | CnV2xMsg_VerOffsetB11Type |
| | ***Comment*** | 11-bit value indicating vertical deviation |
| | VerOffsetB12 | |
| | ***Type*** | CnV2xMsg_VerOffsetB12Type |
| | ***Comment*** | 12-bit value indicating vertical deviation |
| ***Description*** | DF_VeritcalOffset as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document. | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.26 CnV2xMsg_PositionOffsetLLVType

**[CP_SWS_CnV2xMsg_02123]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLLVType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_PositionOffsetLLVPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | PositionOffsetLLTypeIndicator | |
| | ***Type*** | uint8 |
| | ***Comment*** | Indicatiing the exact Union type of PositionOffsetLL 0x00: Positon LL24B<br>0x01: PositonLL28B<br>0x02: PositonLL32B<br>0x03: PositonLL36B<br>0x04: PositonLL44B<br>0x05: PositonLL48B<br>0x06: PositonLL64B |
| | PositionOffsetLL | |
| | ***Type*** | CnV2xMsg_PositionOffsetLLType |
| | ***Comment*** | Indicating latitude and longitude deviation |
| | VerticalOffset | |
| | ***Type*** | CnV2xMsg_VerticalOffsetType |
| | ***Comment*** | Indicating vertical deviation |
| | VerticalOffsetTypeIndicator | |
| | ***Type*** | uint8 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Indicating the exact Union type of VerticalOffset, 0x00: VerOffsetB07, 0x01: VerOffsetB08, 0x02: VerOffsetB09, 0x03: VerOffsetB10, 0x04: VerOffsetB11, 0x05: VerOffsetB12 |
| ***Description*** | DF_PositionOffsetLLV as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.27 CnV2xMsg_PositionOffsetLLVPresenceType

**[CP_SWS_CnV2xMsg_02124]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_PositionOffsetLLVPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | VerticalOffset | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_PositionOffsetLLVType<br>**Tags:** atp.Status=draft | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.28 CnV2xMsg_PathPredictionType

**[CP_SWS_CnV2xMsg_02125]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PathPredictionType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | radiusOfCurve | |
| | ***Type*** | uint16 |
| | ***Comment*** | Radius of curvature, Unit is 0.1m Range: 0..65535 |
| | Confidence | |
| | ***Type*** | uint8 |
| | ***Comment*** | Confidence of path prediction, LSB units of 0.5 percent. Range: 0..200 |
| ***Description*** | DF_PathPrediction as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.29 CnV2xMsg_VehicleEmergencyExtensionsType

**[CP_SWS_CnV2xMsg_02126]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_VehicleEmergencyExtensionsType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_VehicleEmergencyExtensionsPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | ResponseType | |
| | ***Type*** | CnV2xMsg_ResponseTypeType |
| | ***Comment*** | Response type |
| | SirenInUse | |
| | ***Type*** | CnV2xMsg_SirenInUseType |
| | ***Comment*** | Siren status |
| | LightbarInUse | |
| | ***Type*** | CnV2xMsg_LightbarInUseType |
| | ***Comment*** | Light bar status |
| ***Description*** | DF_VehicleEmergencyExtensions as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋*(CP_SRS_CnV2X_00501)*

#### 8.7.3.2.30 CnV2xMsg_VehicleEmergencyExtensionsPresenceType

**[CP_SWS_CnV2xMsg_02143]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_VehicleEmergencyExtensionsPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | ResponseType | 0x04 | Bit 2: Optional child present |
| | bit | SirenInUse | 0x02 | Bit 1: Optional child present |
| | bit | LightBarInUse | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_VehicleEmergencyExtensionsType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋*(CP_SRS_CnV2X_00501)*

#### 8.7.3.2.31 CnV2xMsg_PathHistoryPointType

**[CP_SWS_CnV2xMsg_02129]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PathHistoryPointType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_PathHistoryPointPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | PositionOffsetLLV | |
| | ***Type*** | CnV2xMsg_PositionOffsetLLVType |
| | ***Comment*** | Indicate vehicle 3D position offset |
| | TimeOffset | |
| | ***Type*** | uint16 |
| | ***Comment*** | Indicate time offset of reference time point, LSB units of of 10 mSec. Range: 1..65535 ; A value of 65534 to be used for 655.34 seconds or greater, a value of 65535 to be unavailable |
| | Speed | |
| | ***Type*** | uint16 |
| | ***Comment*** | Indicate vehicle tspeed, Units of 0.02 m/s. Range: 0..8191; The value 8191 indicates that speed is unavailable |
| | PositonConfidenceSet | |
| | ***Type*** | CnV2xMsg_PositionConfidenceSetType |
| | ***Comment*** | Indicate confidence of Vehicle position |
| | CroseHeading | |
| | ***Type*** | uint8 |
| | ***Comment*** | Indicate vehicle heading, LSB is in units of 1.5 degrees. Range: 0..240; the value 240 shall be used for unavailable |
| ***Description*** | DF_PathHistoryPoint as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.32 CnV2xMsg_PathHistoryPointPresenceType

**[CP_SWS_CnV2xMsg_02130]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_PathHistoryPointPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | PositonConfidenceSet | 0x01 | Bit 0 (LSB): Optional child present |

▽

△

| | |
|---|---|
| ***Description*** | Presence flags for CnV2xMsg_PathHistoryPointType<br>**Tags:** atp.Status=draft |
| ***Available via*** | Rte_CnV2xMsg_Type.h |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.33 CnV2xMsg_PathHistoryPointListType

**[CP_SWS_CnV2xMsg_02131]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PathHistoryPointListType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Count | |
| | ***Type*** | uint8 |
| | ***Comment*** | Number of valid elements within array. |
| | PositionOffsetLLV | |
| | ***Type*** | Array of CnV2xMsg_PathHistoryPointListType |
| | ***Size*** | 23 |
| | ***Comment*** | Indicate vehicle 3D position offset |
| ***Description*** | DF_PathHistoryPointList as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.34 CnV2xMsg_PathHistoryType

**[CP_SWS_CnV2xMsg_02132]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PathHistoryType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_PathHistoryPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | InitialPositionFullVector | |
| | ***Type*** | CnV2xMsg_FullPositionVectorType |
| | ***Comment*** | Indicate initial vehicle position vecor |
| | GNSSStatus | |
| | ***Type*** | CnV2xMsg_GNSSStatusType |
| | ***Comment*** | Indicate time offset |
| | CrumbData | |

▽

△

| | | |
|---|---|---|
| | ***Type*** | CnV2xMsg_PathHistoryPointListType |
| | ***Comment*** | Indicate path history points list |
| ***Description*** | DF_PathHistory as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.35 CnV2xMsg_PathHistoryPresenceType

**[CP_SWS_CnV2xMsg_02133]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_PathHistoryPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | InitialPositionFullVector | 0x02 | Bit 1: Optional child present |
| | bit | GNSSStatus | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_PathHistoryType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.36 CnV2xMsg_FullPositionVectorType

**[CP_SWS_CnV2xMsg_02127]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_FullPositionVectorType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_FullPositionVectorPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | Positon3D | |
| | ***Type*** | CnV2xMsg_Position3DType |
| | ***Comment*** | Indicate vehicle 3D position |
| | Heading | |
| | ***Type*** | uint16 |
| | ***Comment*** | Indicate vehicle heading |

▽

△

| | | |
|---|---|---|
| | TransmissionState | |
| | ***Type*** | CnV2xMsg_TransmissionStateType |
| | ***Comment*** | Indicate vehicle transmission state |
| | Speed | |
| | ***Type*** | uint16 |
| | ***Comment*** | Indicate vehicle speed |
| | PositionConfidenceSet | |
| | ***Type*** | CnV2xMsg_PositionConfidenceSetType |
| | ***Comment*** | Indicate vehicle position confidence |
| | TimeConfidence | |
| | ***Type*** | CnV2xMsg_TimeConfidenceType |
| | ***Comment*** | Indicate time confidence |
| | MotionConfidenceSet | |
| | ***Type*** | CnV2xMsg_MotionConfidenceSetType |
| | ***Comment*** | Indicate vehicle Motion confidence |
| ***Description*** | DF_FullPositionVector as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.37 CnV2xMsg_FullPositionVectorPresenceType

**[CP_SWS_CnV2xMsg_02128]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_FullPositionVectorPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | DDataTime | 0x40 | Bit 3: Optional child present |
| | bit | Heading | 0x20 | Bit 5:Optional child present |
| | bit | TransmissionState | 0x10 | Bit 4:Optional child present |
| | bit | Speed | 0x08 | Bit 3:Optional child present |
| | bit | PositionConfidenceSet | 0x04 | Bit 2: Optional child present |
| | bit | TimeConfidence | 0x02 | Bit 1: Optional child present |
| | bit | MotionConfidenceSet | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_FullPositionVectorType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.38 CnV2xMsg_VehicleSafetyExtensionsType

**[CP_SWS_CnV2xMsg_02134]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_VehicleSafetyExtensionsType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_VehicleSafetyExtensionsPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | VehicleEventFlags | |
| | ***Type*** | CnV2xMsg_VehicleEventFlagsType |
| | ***Comment*** | Mark optional childs present or not |
| | PathHistory | |
| | ***Type*** | CnV2xMsg_PathHistoryType |
| | ***Comment*** | Mark optional childs present or not |
| | PathPrediction | |
| | ***Type*** | CnV2xMsg_PathPredictionType |
| | ***Comment*** | Mark optional childs present or not |
| | ExteriorLights | |
| | ***Type*** | CnV2xMsg_ExteriorLightsType |
| | ***Comment*** | Mark optional childs present or not |
| ***Description*** | DF_VehicleSafetyExtensions as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.39 CnV2xMsg_VehicleSafetyExtensionsPresenceType

**[CP_SWS_CnV2xMsg_02135]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_VehicleSafetyExtensionsPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | VehicleEventFlags | 0x04 | Bit 2: Optional child present |
| | bit | PathPrediction | 0x02 | Bit 1: Optional child present |
| | bit | ExteriorLights | 0x01 | Bit 0 (LSB): Optional child present |
| ***Description*** | Presence flags for CnV2xMsg_VehicleSafetyExtensionsType<br>**Tags:** atp.Status=draft | | | |
| ***Variation*** | – | | | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.40 CnV2xMsg_BsmType

**[CP_SWS_CnV2xMsg_02136]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_BsmType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Presence | |
| | ***Type*** | CnV2xMsg_BsmPresenceType |
| | ***Comment*** | Mark optional childs present or not |
| | MsgCount | |
| | ***Type*** | uint8 |
| | ***Comment*** | Msg count, Range: 0..127; After the number reaches 127, the next one goes back to 0 |
| | Id | |
| | ***Type*** | CnV2xMsg_VehicleIDType |
| | ***Comment*** | Vehicle ID |
| | DSecond | |
| | ***Type*** | uint16 |
| | ***Comment*** | Indicate milliseconds in a minute, Range: 0..65535; a value =6000 indicate invalid value |
| | TimeConfidence | |
| | ***Type*** | CnV2xMsg_TimeConfidenceType |
| | ***Comment*** | Indicate time confidence |
| | Position3D | |
| | ***Type*** | CnV2xMsg_Position3DType |
| | ***Comment*** | Indicate vehicle 3D position |
| | PositionAccuracy | |
| | ***Type*** | CnV2xMsg_PositionAccuracyType |
| | ***Comment*** | Accuracy for GNSS system |
| | PositionConfidenceSet | |
| | ***Type*** | CnV2xMsg_PositonConfidenceSetType |
| | ***Comment*** | Realtime position confidence |
| | TransmissionState | |
| | ***Type*** | CnV2xMsg_TransmissionStateType |
| | ***Comment*** | Indicate vehicle transmission state |
| | Speed | |
| | ***Type*** | uint16 |
| | ***Comment*** | Indicate vehicle speed, Units of 0.02 m/s, Range: 0..8191; The value 8191 indicates that speed is unavailable |
| | Heading | |
| | ***Type*** | uint16 |
| | ***Comment*** | Indicate vehicle heading, LSB of 0.0125 degrees Range: 0..28800 |
| | SteeringWheelAngle | |
| | ***Type*** | sint8 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Absolute accuracy of steering wheelAngle value, Units of 1.5 degrees. Range: -126..127; A range of 189 to +189 degrees, +127 to be used for unavailable |
| | MotionConfidenceSet | |
| | ***Type*** | CnV2xMsg_MotionConfidenceSetType |
| | ***Comment*** | Indicate vehicle Motion confidence |
| | AccelerationSet4Way | |
| | ***Type*** | CnV2xMsg_AccelerationSet4WayType |
| | ***Comment*** | Indicate 4 way acceleration |
| | BrakeSystemStatus | |
| | ***Type*** | CnV2xMsg_BrakeSystemStatusType |
| | ***Comment*** | Indicate vehicle brake system status |
| | VehicleSize | |
| | ***Type*** | CnV2xMsg_VehicleSizeType |
| | ***Comment*** | Indicate vehicle size |
| | VehicleClassification | |
| | ***Type*** | CnV2xMsg_VehicleClassificationType |
| | ***Comment*** | Indicate vehicle types |
| | VehicleSafetyExtensions | |
| | ***Type*** | CnV2xMsg_VehicleSafetyExtensionsType |
| | ***Comment*** | Vehicle safety auxiliary information |
| | VehicleEmergencyExtensions | |
| | ***Type*** | CnV2xMsg_VehicleEmergencyExtensionsType |
| | ***Comment*** | Auxiliary information for emergency vehicles |
| ***Description*** | BSM frame as defined in CCSA YD/T 3709-2020. Values for data elements within this structure shall be used according that document.<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.3.2.41 CnV2xMsg_BsmPresenceType

**[CP_SWS_CnV2xMsg_02137]**{DRAFT} ⌈

| | | | | |
|---|---|---|---|---|
| ***Name*** | CnV2xMsg_BsmPresenceType (draft) | | | |
| ***Kind*** | Bitfield | | | |
| ***Derived from*** | uint8 | | | |
| ***Elements*** | ***Kind*** | ***Name*** | ***Mask*** | ***Description*** |
| | bit | TimeConfidence | 0x04 | Bit 2: Optional child present |
| | bit | MotionConfidenceSet | 0x02 | Bit 1: Optional child present |
| | bit | VehicleEmergencyExtesnsions | 0x01 | Bit 0 (LSB): Optional child present |

▽

△

| | |
|---|---|
| ***Description*** | Presence flags for CnV2xMsg_BsmType<br>**Tags:** atp.Status=draft |
| ***Variation*** | – |
| ***Available via*** | Rte_CnV2xMsg_Type.h |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.42 CnV2xMsg_BsmRootType

**[CP_SWS_CnV2xMsg_02138]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_BsmRootType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Bsm | |
| | ***Type*** | CnV2xMsg_BsmType |
| | ***Comment*** | Structure of the BSM data |
| | TransactionID | |
| | ***Type*** | uint32 |
| | ***Comment*** | TransactionId for received BSM |
| | RxParams | |
| | ***Type*** | CnV2xMsg_RxParamsType |
| | ***Comment*** | Rx parameters of the received BSM packet |
| ***Description*** | BSM root message structure delivered to Applications<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.3.2.43 CnV2xMsg_PositionAndTimeType

**[CP_SWS_CnV2xMsg_02139]**{DRAFT} ⌈

| | | |
|---|---|---|
| ***Name*** | CnV2xMsg_PositionAndTimeType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Position3D | |
| | ***Type*** | CnV2xMsg_Position3DType |
| | ***Comment*** | Indicate 3D position |
| | PositionAccuracy | |
| | ***Type*** | CnV2xMsg_PositionAccuracyType |
| | ***Comment*** | Accuracy for GNSS system |
| | Timestamp | |
| | ***Type*** | uint32 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Timestamp [1 ms] |
| | Heading | |
| | ***Type*** | uint16 |
| | ***Comment*** | Heading [0.0125 degree] Range: 0..28800 |
| | Speed | |
| | ***Type*** | uint16 |
| | ***Comment*** | Speed [0.02 m/s] Range: 0..8192 |
| | Position3DValid | |
| | ***Type*** | boolean |
| | ***Comment*** | Indicates that position3Dis valid |
| | PositionAccuracyValid | |
| | ***Type*** | boolean |
| | ***Comment*** | Indicates that PositionAccuracy is valid |
| ***Description*** | Position and time related information as defined within CCSA YD/T 3709-2020<br>**Tags:** atp.Status=draft | |
| ***Variation*** | – | |
| ***Available via*** | Rte_CnV2xMsg_Type.h | |

⌋(*CP_SRS_CnV2X_00501*)

## 8.7.4 Ports

### 8.7.4.1 CnV2xMsg_CnV2xMsg_Vdp

**[CP_SWS_CnV2xMsg_07001]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_Vdp (draft) | | |
| ***Kind*** | RequiredPort | ***Interface*** | CnV2xMsgVdp |
| ***Description*** | Port for retrieving data from VDP application<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |

⌋(*CP_SRS_CnV2X_00501*)

### 8.7.4.2 CnV2xMsg_CnV2xMsg_Cnv2xApplRxIndicationBSM

**[CP_SWS_CnV2xMsg_07002]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_CnV2xApplRxIndicationBSM (draft) | | |
| ***Kind*** | ProvidedPort | ***Interface*** | CnV2xApplRxIndicationBsm |
| ***Description*** | Port for delivering received BSMs to application layer<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.4.3 CnV2xMsg_CnV2xMsg_Poti

**[CP_SWS_CnV2xMsg_07003]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_Poti (draft) | | |
| ***Kind*** | ProvidedPort | ***Interface*** | CnV2xMsgPoti |
| ***Description*** | Service port for exchange of Postion and Time info.<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.4.4 CnV2xMsg_CnV2xMsg_Cnv2xApplRxIndicationRSI

**[CP_SWS_CnV2xMsg_07004]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_CnV2xApplRxIndicationRSI (draft) | | |
| ***Kind*** | ProvidedPort | ***Interface*** | CnV2xApplRxIndicationRsi |
| ***Description*** | Port for delivering received RSIs to application layer<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.4.5 CnV2xMsg_CnV2xMsg_Cnv2xApplRxIndicationRSM

**[CP_SWS_CnV2xMsg_07005]**{DRAFT} ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_CnV2xApplRxIndicationRSM (draft) | | |
| ***Kind*** | ProvidedPort | ***Interface*** | CnV2xApplRxIndicationRsm |
| ***Description*** | Port for delivering received RSMs to application layer<br>**Tags:** atp.Status=draft | | |
| ***Variation*** | – | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.4.6 CnV2xMsg_CnV2xMsg_Cnv2xApplRxIndicationSPAT

**[CP_SWS_CnV2xMsg_07006]** ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_CnV2xApplRxIndicationSPAT | | |
| ***Kind*** | ProvidedPort | ***Interface*** | CnV2xApplRxIndicationSpat |
| ***Description*** | Port for delivering received SPATs to application layer | | |
| ***Variation*** | – | | |

⌋(*CP_SRS_CnV2X_00501*)

#### 8.7.4.7 CnV2xMsg_CnV2xMsg_Cnv2xApplRxIndicationMAP

**[CP_SWS_CnV2xMsg_07007]** ⌈

| | | | |
|---|---|---|---|
| ***Name*** | CnV2xMsg_CnV2xApplRxIndicationMAP | | |
| ***Kind*** | ProvidedPort | ***Interface*** | CnV2xApplRxIndicationMap |
| ***Description*** | Port for delivering received MAPs to application layer | | |
| ***Variation*** | – | | |

⌋(*CP_SRS_CnV2X_00501*)

# 9 Sequence diagrams

## 9.1 time Initialization

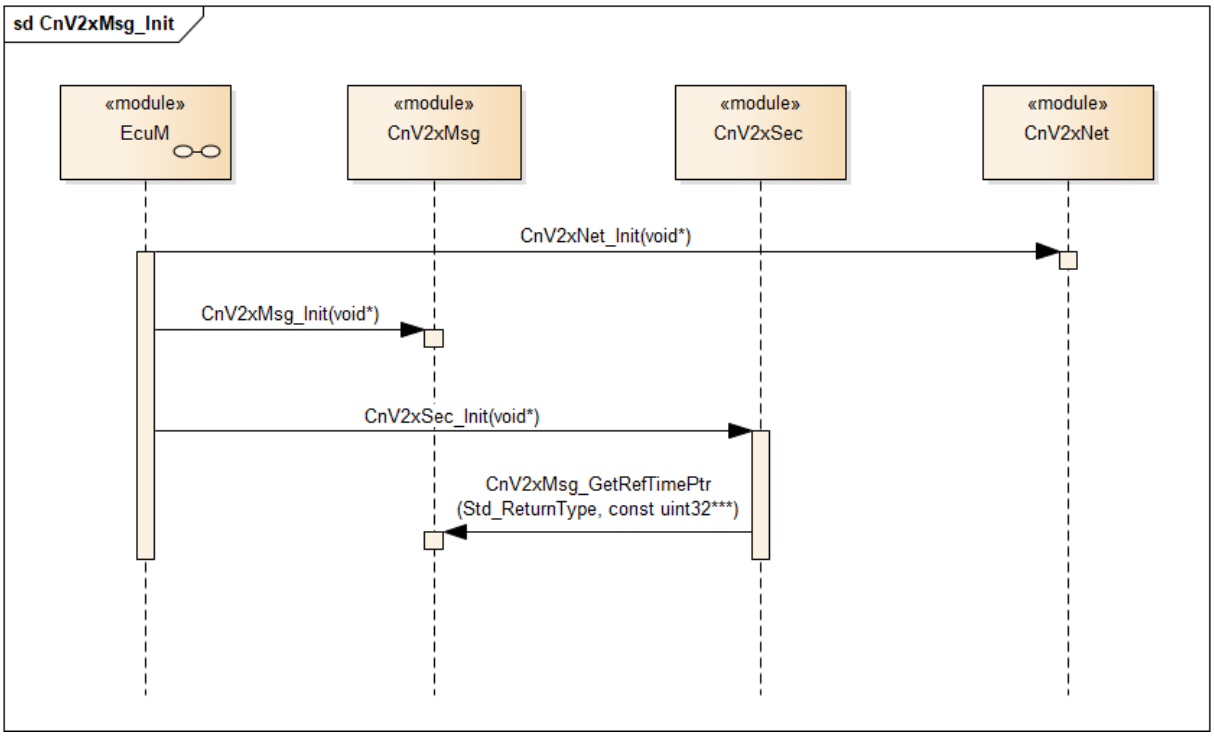


**Figure 9.1: Time Initialization**

## 9.2 Position and Time Update

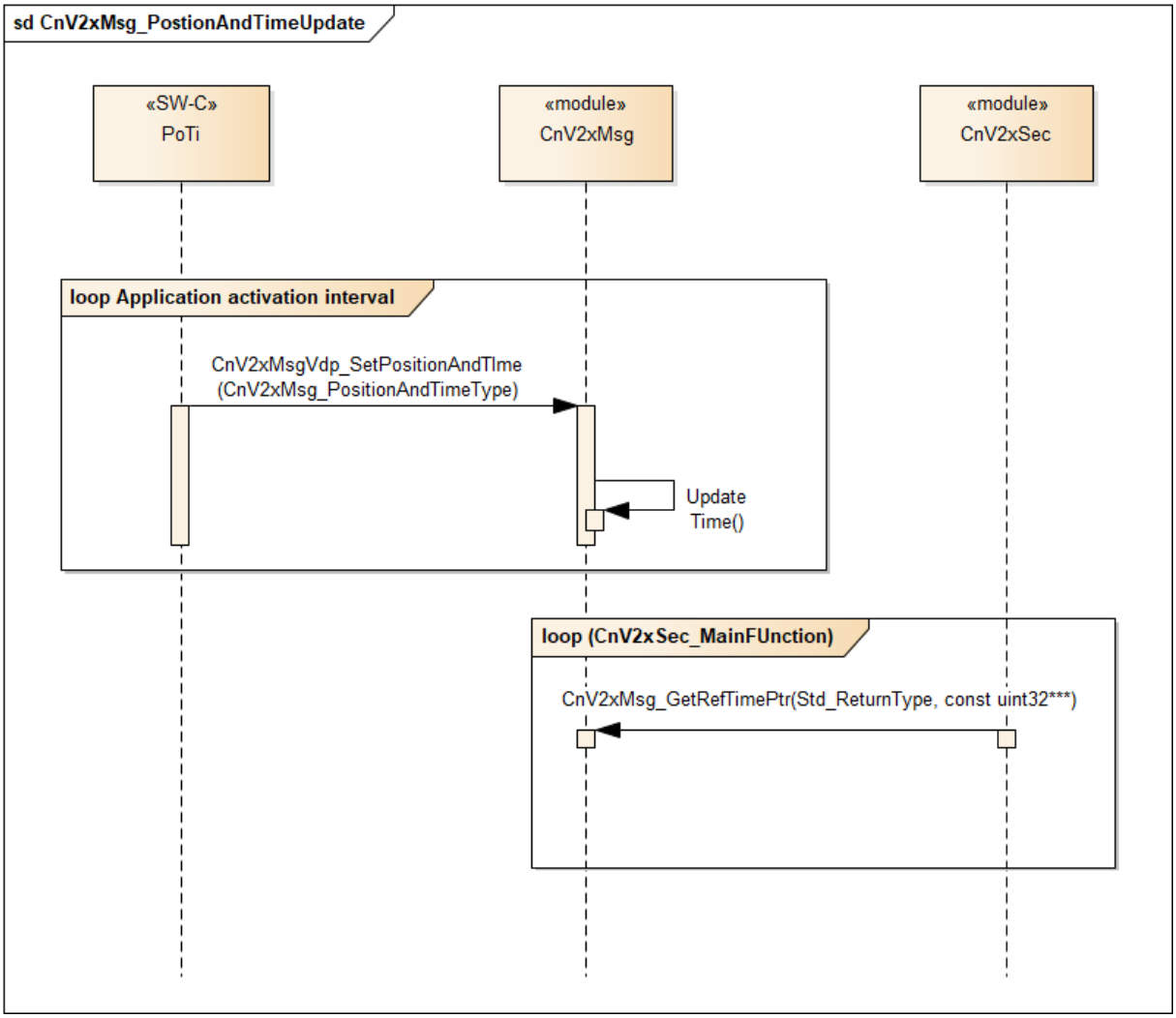


**Figure 9.2: Position and Time Update**

## 9.3 BSM Generation and Transmission

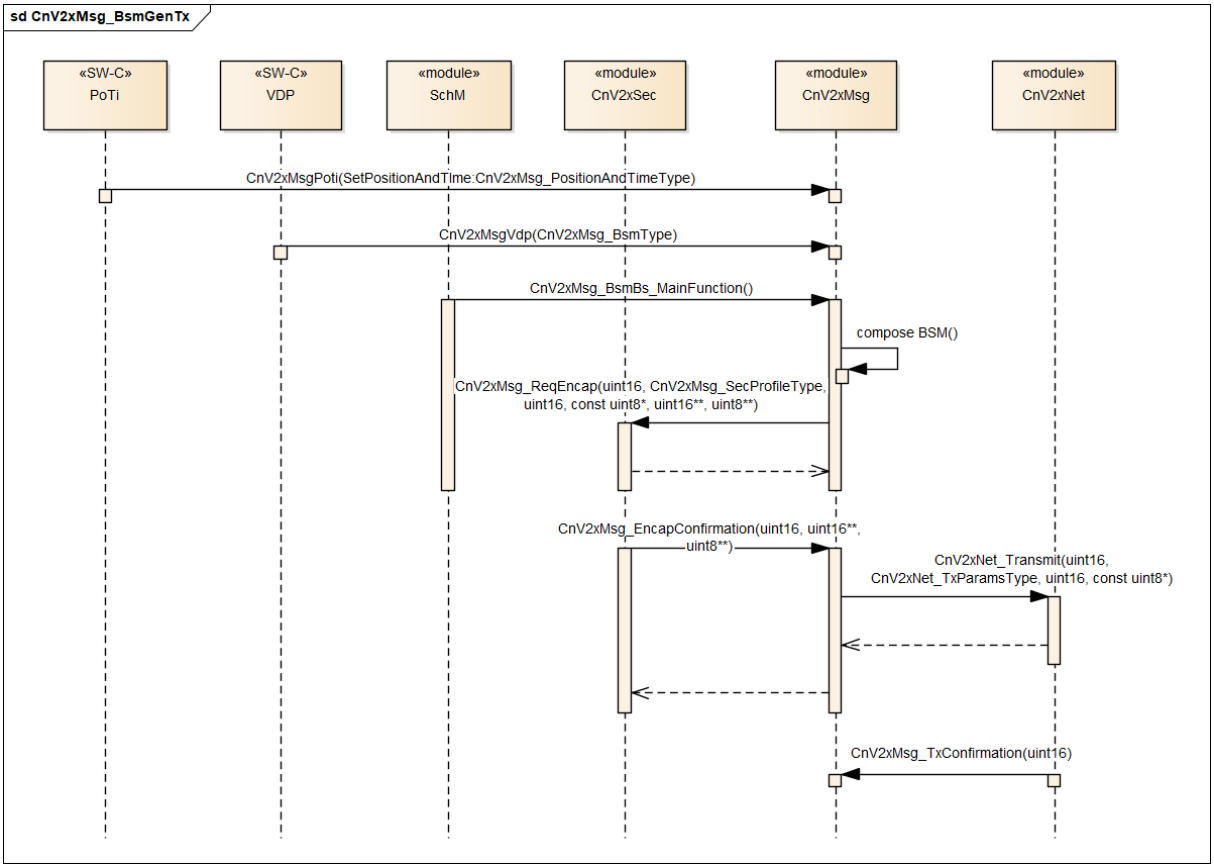


**Figure 9.3: BSM Generation and Transmission**

## 9.4 BSM Reception

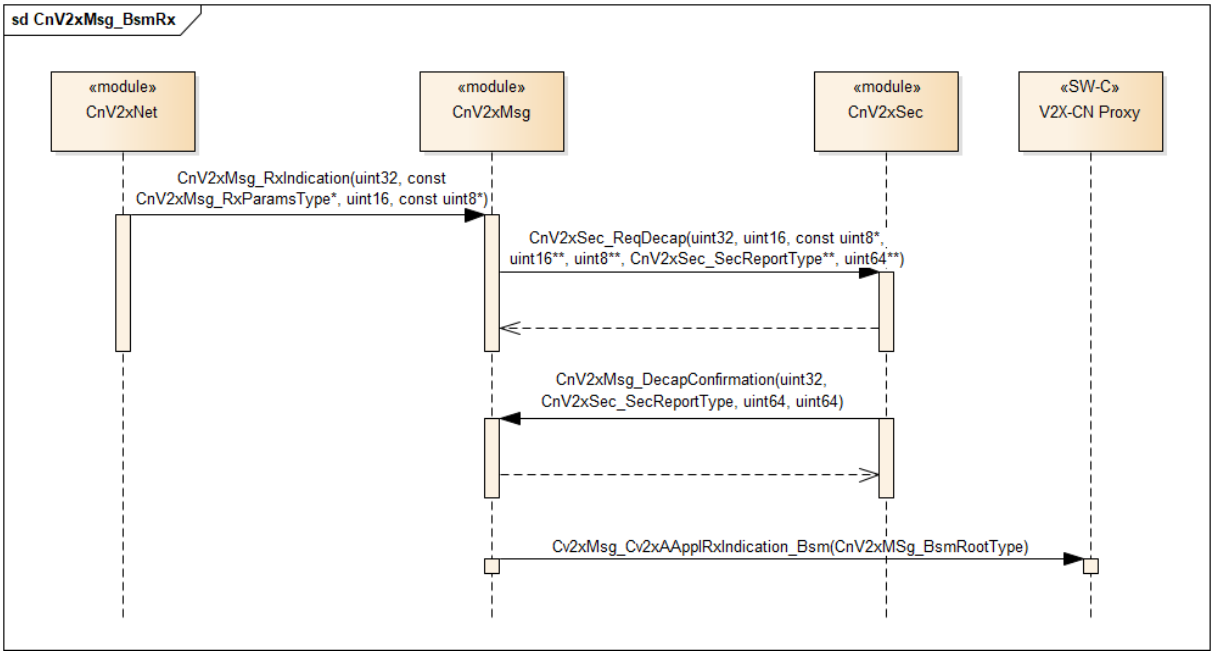


**Figure 9.4: BSM Reception**

## 9.5 RSI Reception

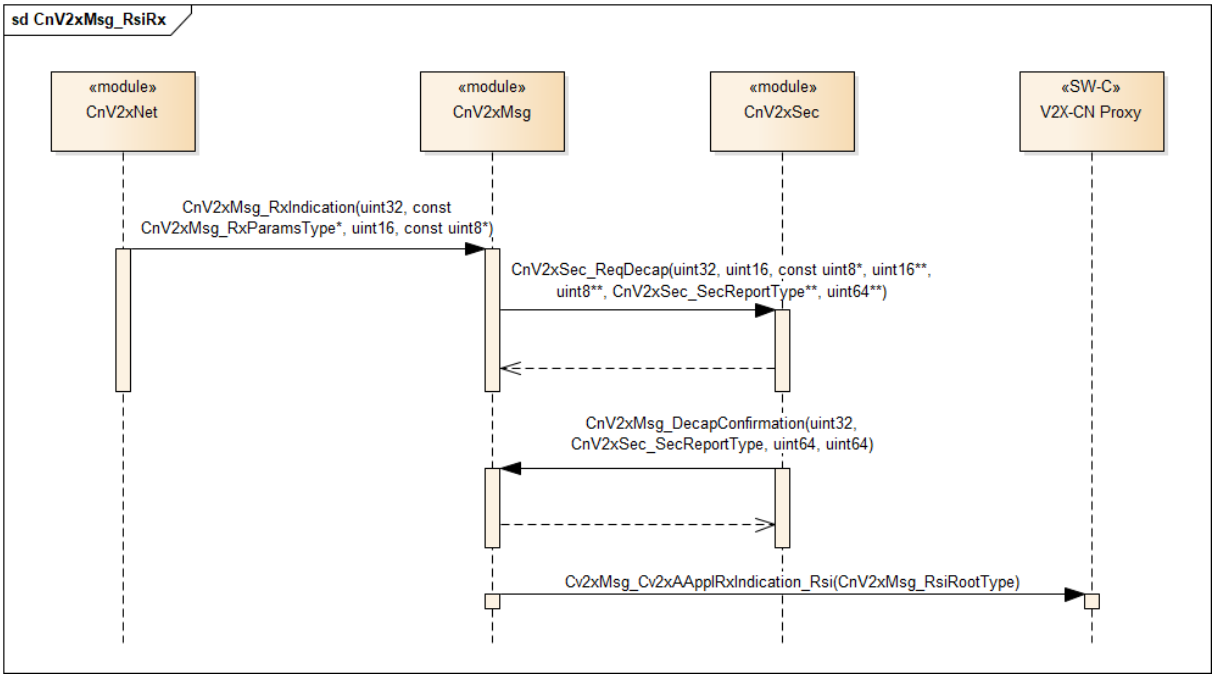


**Figure 9.5: RSI Reception**

## 9.6 RSM Reception

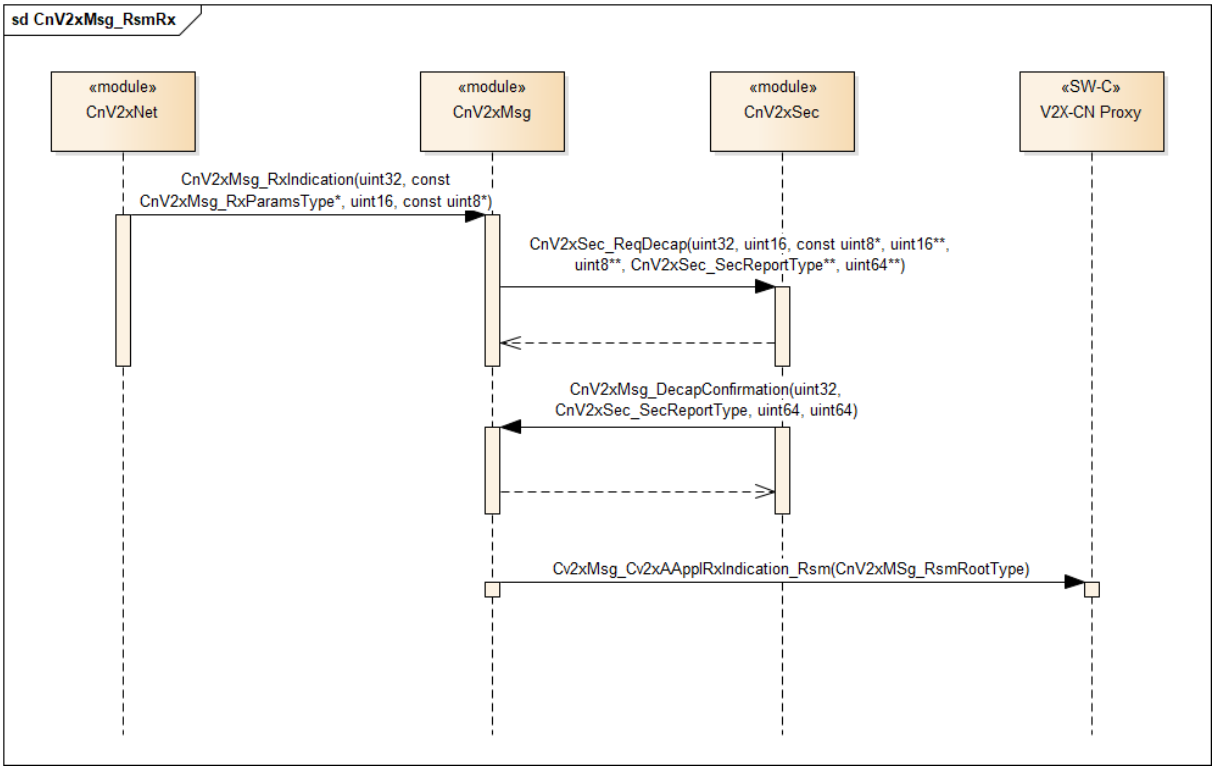


**Figure 9.6: RSM Reception**

## 9.7 SPAT Reception

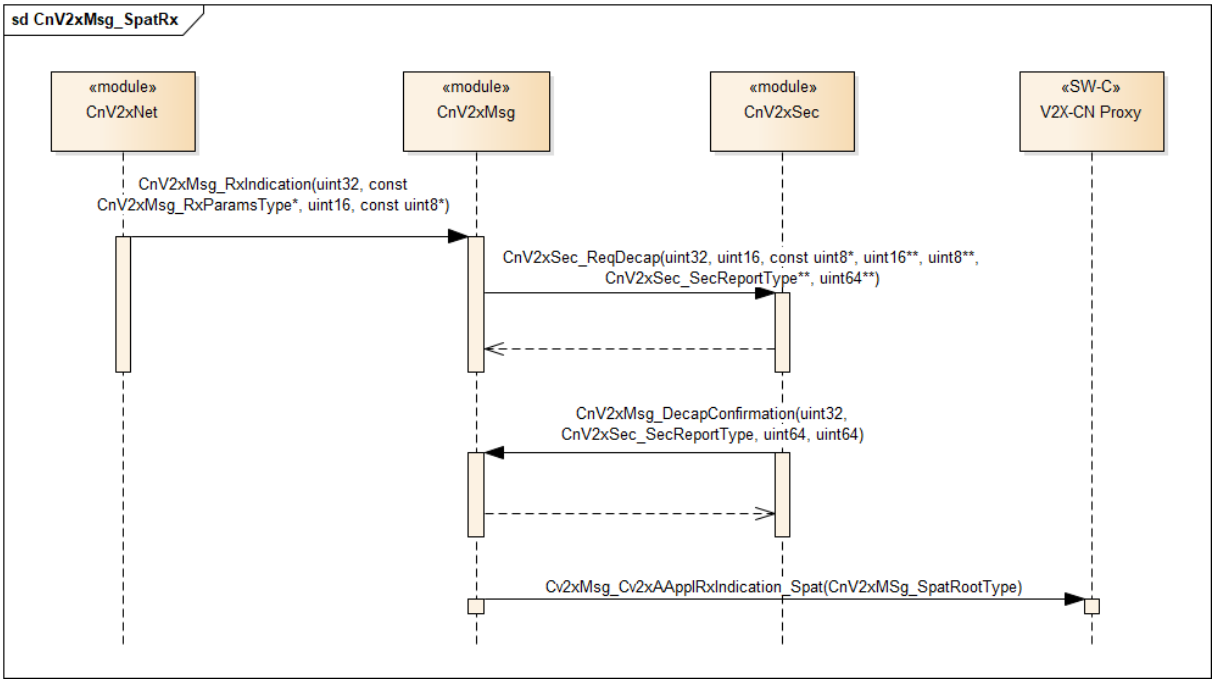


**Figure 9.7: SPAT Reception**

## 9.8 MAP Reception

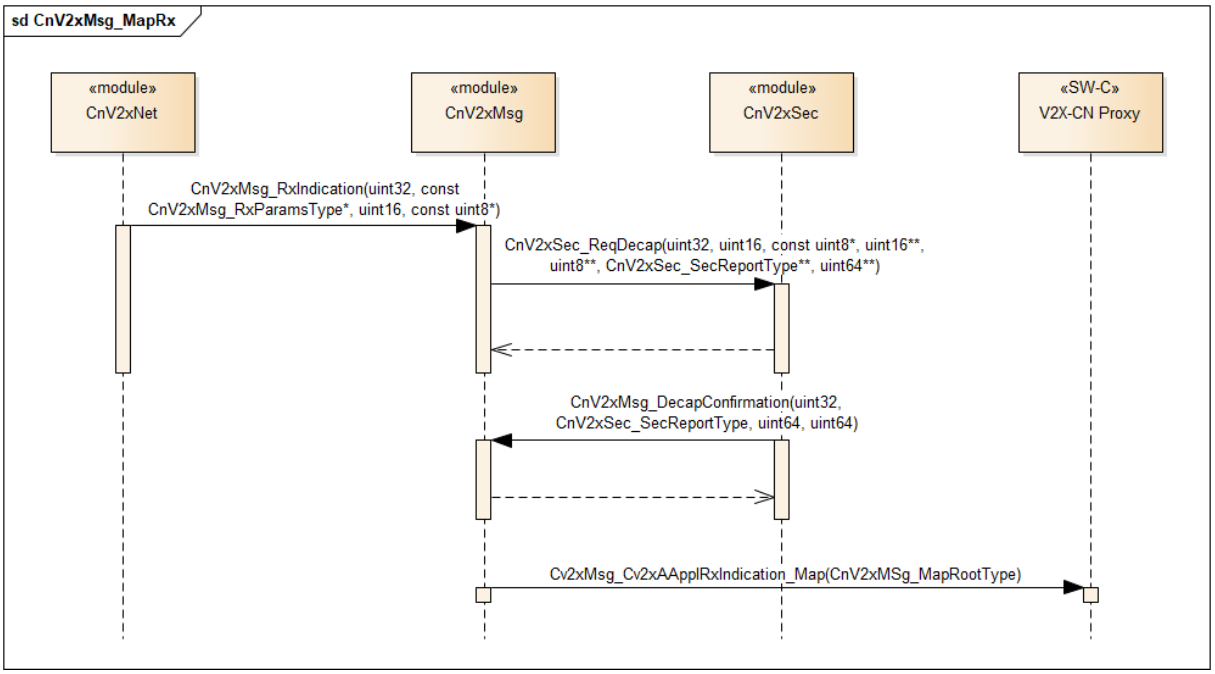


**Figure 9.8: MAP Reception**

## 9.9 Update Pseudonym

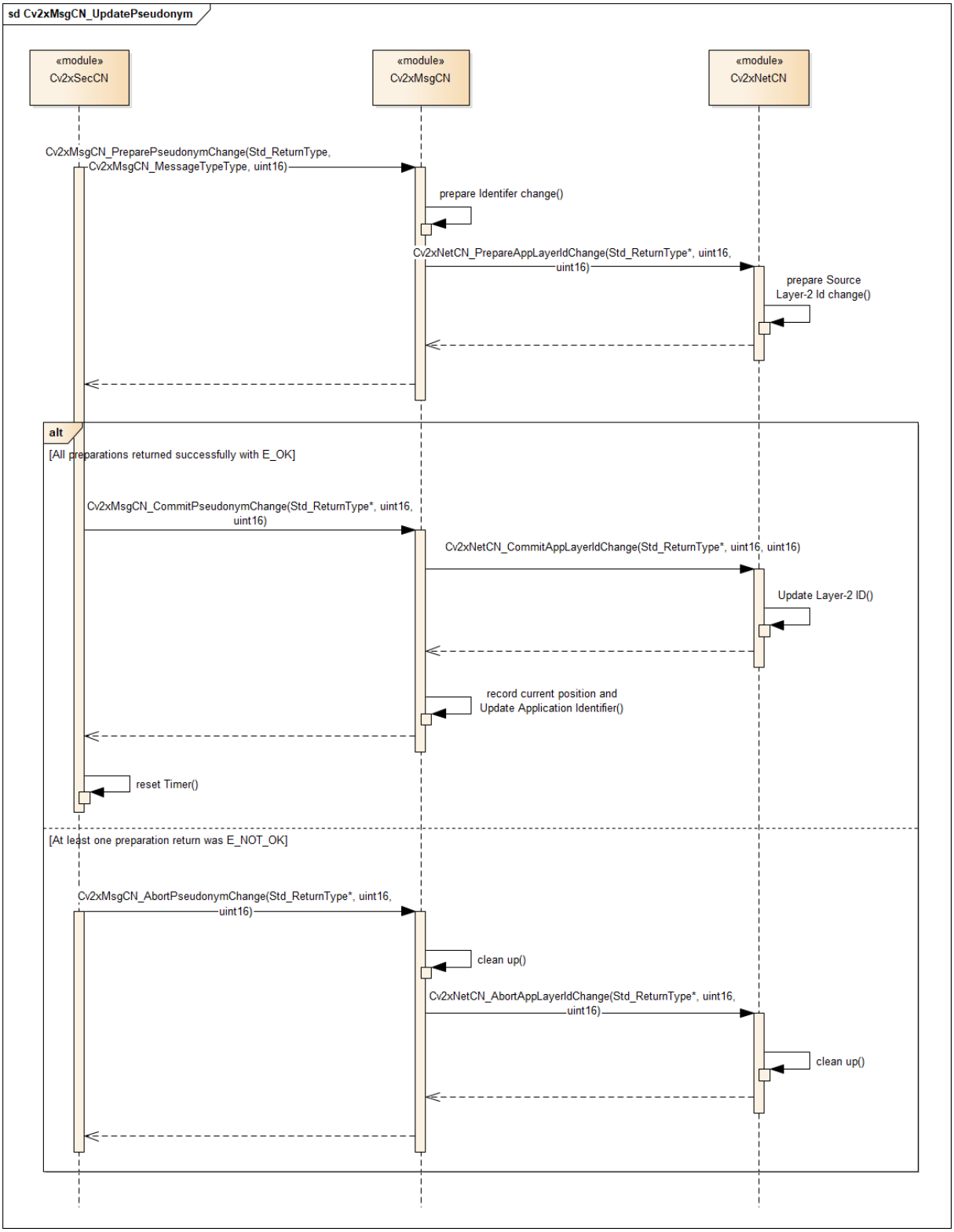


**Figure 9.9: Update Pseudonym**

## 9.10 Messages Reception via V2xDM

V2X messages reception via V2xDM please refer to [8] chapter 9.3.

# 10 Configuration specification

## 10.1 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters are described in Chapter 7 and Chapter 8.

### 10.1.1 Variants

**[SWS_CnV2xMsg_08001]** ⌈The CnV2xMsg module only supports VARIANT-PRE-COMPILE⌋ *(SRS_BSW_00345)*

### 10.1.2 CnV2xMsg

| SWS Item | [ECUC_CnV2xMsg_00001] |
|---|---|
| Module Name | CnV2xMsg |
| Description | Configuration of the CnV2xMsg module. |
| Post-Build Variant Support | false |
| Supported Config Variants | VARIANT-PRE-COMPILE |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| CnV2xMsgConfig | 1 | This container contains the configuration parameters of the BSW module CnV2xMsg.<br>**Tags:** atp.Status=draft |
| CnV2xMsgGeneral | 1 | This container contains the general configuration parameters of the AUTOSAR CnV2xMsg module.<br>**Tags:** atp.Status=draft |

### 10.1.3 CnV2xMsgGeneral

| SWS Item | [ECUC_CnV2xMsg_00002] |
|---|---|
| Container Name | CnV2xMsgGeneral |
| Parent Container | CnV2xMsg |
| Description | This container contains the general configuration parameters of the AUTOSAR CnV2x Msg module.<br>**Tags:** atp.Status=draft |
| Configuration Parameters | |

| SWS Item | [ECUC_CnV2xMsg_00003] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgBsmBsMainFunction | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | This parameter defines the schedule period of CnV2xMsg_BsmBs_Main Function.Unit:[s]<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | ]0 .. 1[ | | |
| Default value | 0.1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00007] | | |
|---|---|---|---|
| Parameter Name | Cnv2xMsgCRsiSMainFunction | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | This parameter defines the schedule period of CnV2xMsg_RsiS_MainFunction.Unit:[s]<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | ]0 .. INF[ | | |
| Default value | 0.1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00004] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgDevErrorDetect | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | Switches the Default Error Tracer (Det) detection and notification ON or OFF. - true: enabled (ON) - false: disabled (OFF)<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00010] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgMapSMainFunction | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | This parameter defines the schedule period of CnV2xMsg_MapS_MainFunction.Unit:[s]<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | ]0 .. INF[ | | |
| Default value | 0.1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00006] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgMgtMainFunction | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | This parameter defines the schedule period of CnV2xMsg_Mgt_MainFunction.Unit:[s]<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | ]0 .. 1[ | | |
| Default value | 0.1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00008] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgRsmSMainFunction | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | This parameter defines the schedule period of CnV2xMsg_RsmS_Main Function.Unit:[s]<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | ]0 .. INF[ | | |
| Default value | 0.1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00009] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgSpatSMainFunction | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | This parameter defines the schedule period of CnV2xMsg_SpatS_Main Function.Unit:[s]<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | ]0 .. INF[ | | |
| Default value | 0.1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00011] | |
|---|---|---|
| Parameter Name | CnV2xMsgVehicleClass | |
| Parent Container | CnV2xMsgGeneral | |
| Description | This configuration value defines the Vehicle Class information, Road Side Unit not supported by AUTOSAR.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucEnumerationParamDef | |
| Range | CNV2XMSG_VC_BUS | 50<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_EM_AMBULANCE | 65<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_EM_ENGINEERING | 68<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_EM_FIRETRUCK_HEAVY | 63<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_EM_FIRETRUCK_LIGHT | 62<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_EM_NURSING | 64<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_EM_POLICE_HEAVY | 67<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_EM_POLICE_LIGHT | 66<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_GOODS_LIGHT | 20<br>**Tags:** atp.Status=draft |
| | CNV2XMSG_VC_GOODS_SEMITRAILER | 25<br>**Tags:** atp.Status=draft |

△

| | | | |
|---|---|---|---|
| | CNV2XMSG_VC_PASSENGER | 10<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_VC_SPECIAL | 1<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_VC_UNKNOWN | 0<br>**Tags:** atp.Status=draft | |
| **Default value** | CNV2XMSG_VC_UNKNOWN | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | **[ECUC_CnV2xMsg_00005]** | | |
| **Parameter Name** | CnV2xMsgVersionInfoApi | | |
| **Parent Container** | CnV2xMsgGeneral | | |
| **Description** | Enable/disables the API for reading the version information of the CnV2xMsg Module. - true: enabled (ON) - false: disabled (OFF)<br>**Tags:** atp.Status=draft | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | **[ECUC_CnV2xMsg_00017]** | | |
| **Parameter Name** | CnV2xMsgvMaxCurveRadius | | |
| **Parent Container** | CnV2xMsgGeneral | | |
| **Description** | The maximum Curve Radius<br>**Tags:** atp.Status=draft | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 2500 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **SWS Item** | **[ECUC_CnV2xMsg_00013]** |
| **Parameter Name** | CnV2xMsgvMaxPHistDistance |
| **Parent Container** | CnV2xMsgGeneral |

▽

△

| | | | |
|---|---|---|---|
| **Description** | The Maximum distance between the first and last path history point along the vehicle path), Unit:[m]<br>**Tags:** atp.Status=draft | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 300 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | **[ECUC_CnV2xMsg_00015]** | | |
| **Parameter Name** | CnV2xMsgvMaxPHistPoints | | |
| **Parent Container** | CnV2xMsgGeneral | | |
| **Description** | Maximum number of path history points in a BSM packet<br>**Tags:** atp.Status=draft | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 15 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | **[ECUC_CnV2xMsg_00016]** | | |
| **Parameter Name** | CnV2xMsgvMinCurveRadius | | |
| **Parent Container** | CnV2xMsgGeneral | | |
| **Description** | The minimum Curve Radius<br>**Tags:** atp.Status=draft | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 100 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **SWS Item** | **[ECUC_CnV2xMsg_00012]** |
| **Parameter Name** | CnV2xMsgvMinPHistDistance |
| **Parent Container** | CnV2xMsgGeneral |

▽

△

| | | | |
|---|---|---|---|
| **Description** | The Minimum distance between the first and last path history point along the vehicle path), Unit:[m]<br>**Tags:** atp.Status=draft | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 200 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | **[ECUC_CnV2xMsg_00014]** | | |
| **Parameter Name** | CnV2xMsgvPathPerpendicularDist | | |
| **Parent Container** | CnV2xMsgGeneral | | |
| **Description** | The perpendicular distance between any point on the vehicle path and the straight line connecting two adjacent path history points, unit:[m]<br>**Tags:** atp.Status=draft | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 200 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | **[ECUC_CnV2xMsg_00018]** | | |
| **Parameter Name** | CnV2xMsgvPPredRadiusError | | |
| **Parent Container** | CnV2xMsgGeneral | | |
| **Description** | The error from the actual radius, Unit:[%]<br>**Tags:** atp.Status=draft | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 100 | | |
| **Default value** | 2 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00019] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgvPPredTransitionTime | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | The transition time from a constant radius of curvature (R1) to a new constant radius of curvature (R2), unit: [s]<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | [-INF .. INF] | | |
| Default value | 4 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| SWS Item | [ECUC_CnV2xMsg_00020] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgvStationarySpeedThresh | | |
| Parent Container | CnV2xMsgGeneral | | |
| Description | The threshold of vehicle speed, unit:[m/s]<br>**Tags:** atp.Status=draft | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 18446744073709551615 | | |
| Default value | 1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.1.4 CnV2xMsgConfig

| SWS Item | [ECUC_CnV2xMsg_00022] |
|---|---|
| Container Name | CnV2xMsgConfig |
| Parent Container | CnV2xMsg |
| Description | This container contains the configuration parameters of the BSW module CnV2xMsg.<br>**Tags:** atp.Status=draft |
| Configuration Parameters | |

| SWS Item | [ECUC_CnV2xMsg_00021] | | |
|---|---|---|---|
| Parameter Name | CnV2xMsgV2xDmServiceConfig | | |
| Parent Container | CnV2xMsgConfig | | |
| Description | Enable/disables the messages reception service via V2xDm. - true: enabled (ON) - false: disabled (OFF) | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | – | |
| | Post-build time | – | |
| Scope / Dependency | scope: local | | |

| Included Containers | | |
|---|---|---|
| Container Name | Multiplicity | Scope / Dependency |
| CnV2xMsgDmMsgConfig | 1..* | This container contains the configuration of all messages that are passed on to the V2x Data Manager.<br>**Tags:** atp.Status=draft |

## 10.1.5 CnV2xMsgDmMsgConfig

| SWS Item | [ECUC_CnV2xMsg_00023] | | |
|---|---|---|---|
| Container Name | CnV2xMsgDmMsgConfig | | |
| Parent Container | CnV2xMsgConfig | | |
| Description | This container contains the configuration of all messages that are passed on to the V2x Data Manager.<br>**Tags:** atp.Status=draft | | |
| Post-Build Variant Multiplicity | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | All Variants |
| | Link time | – | |
| | Post-build time | – | |
| Configuration Parameters | | | |

| SWS Item | [ECUC_CnV2xMsg_00025] | |
|---|---|---|
| Parameter Name | CnV2xMsgDmAid | |
| Parent Container | CnV2xMsgDmMsgConfig | |
| Description | When message is processed by the V2X Data Manager (CnV2xMsgV2xDmService Config is enabled), this configuration is used to indicate the type of message.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..* | |
| Type | EcucEnumerationParamDef | |
| Range | CNV2XMSG_AID_DYNAMIC_RSI | 3622<br>**Tags:** atp.Status=draft |

▽

△

| | | | |
|---|---|---|---|
| | CNV2XMSG_AID_EMERGENCY_EVENTTRIGGERED_BSM | 3617<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_EMERGENCY_REGULAR_BSM | 113<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_MAP | 3618<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_NONEMERGENCY_EVENTTRIGGERED_BSM | 112<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_NONEMERGENCY_REGULAR_BSM | 111<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_RSM | 3623<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_SEMIDYNAMIC_RSI | 3621<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_SPAT | 3619<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_STATIC_RSI | 3620<br>**Tags:** atp.Status=draft | |
| | CNV2XMSG_AID_V2X_TERMINAL_AFTERMARKET | 3617<br>**Tags:** atp.Status=draft | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

# A Not applicable requirements