Document Title	Specification of Vehicle-2-X Facilities
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	795

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R24-11

Document Change History

Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	• Rework of VDP service API for CAM • Removal of useless data types
2023-11-23	R23-11	AUTOSAR Release Management	• Removal of Tx confirmation and transaction ID • Removal of Verification on Demand
2022-11-24	R22-11	AUTOSAR Release Management	• New architectitecture with V2x DM
2021-11-25	R21-11	AUTOSAR Release Management	• Editorial Changes
2020-11-30	R20-11	AUTOSAR Release Management	• Corrections of typos in chap. 8 definitions • Editorial Changes
2019-11-28	R19-11	AUTOSAR Release Management	• Added support for C2CCC BSP 1.3 • Editorial Changes • Service API bugs corrections • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Added IVIM support • Added SPATEM support • Added MAPEM support

△

2017-12-08	4.3.1	AUTOSAR Release Management	• Editorial Changes
2016-11-30	4.3.0	AUTOSAR Release Management	• Initial Release

Disclaimer

Contents

1 Introduction and functional overview 7
1.1 Architectural overview . . . 7
1.2 Functional overview . . . 8
1.2.1 Cooperative Awareness (CA) . . . 8
1.2.1.1 CA basic service in the AUTOSAR architecture . . . 8
1.2.1.2 CA basic service functional architecture . . . 9
1.2.2 Decentralized Environmental Notification (DEN) . . . 9
1.2.2.1 DEN basic service in the AUTOSAR architecture . . . 9
1.2.2.2 DEN basic service functional architecture . . . 9
1.2.3 Vehicle Data Provider (VDP) . . . 10
1.2.4 Local Dynamic Map (LDM) . . . 10
1.2.5 Infrastructure to Vehicle Information (IVI) . . . 10
1.2.5.1 IVI service in the AUTOSAR architecture . . . 10
1.2.5.2 IVI service functional architecture . . . 11
1.2.6 Road and Lane Topology (RLT) service . . . 11
1.2.6.1 RLT service in the AUTOSAR architecture . . . 11
1.2.6.2 RLT service functional architecture . . . 11
1.2.7 Traffic Light Maneuver (TLM) service . . . 12
1.2.7.1 TLM service in the AUTOSAR architecture . . . 12
1.2.7.2 TLM service functional architecture . . . 12
2 Acronyms and Abbreviations 13
3 Related documentation 14
3.1 Input documents & related standards and norms . . . 14
3.2 Related specification . . . 15
4 Constraints and assumptions 16
4.1 Limitations . . . 16
4.2 Applicability to car domains . . . 16
4.3 Authorisation Tickets and Pseudonyms . . . 16
5 Dependencies to other modules 17
5.1 AUTOSAR DET (Default Error Tracer) . . . 17
5.2 AUTOSAR EcuM (Ecu State Manager) . . . 17
5.3 V2x Vehicle Data Provider . . . 17
5.4 AUTOSAR V2xDM . . . 17
5.5 AUTOSAR V2xBtp . . . 17
5.6 AUTOSAR V2xM . . . 18
6 Requirements Tracing 19
7 Functional specification 21
7.1 Startup behavior . . . 22
7.2 General Format Specification . . . 23

7.3 CA Functional Specification . . . 23
7.3.1 CA Initialization, Activation and Deactivation . . . 23
7.3.2 CAM Generation, Sending and Frequency Management . . . 24
7.3.3 CAM Generation Frequency Management for RSU ITS-Ss . . . 25
7.3.4 CAM Time Requirement . . . 26
7.3.5 CAM Format Specification . . . 26
7.3.6 Others . . . 28
7.4 DEN Functional Specification . . . 28
7.4.1 DEN Initialization . . . 28
7.4.2 DENM Transmission Management . . . 28
7.4.3 DENM Reception Management . . . 29
7.4.4 DENM Repetition . . . 29
7.4.5 DENM Keep Alive Forwarding (KAF) . . . 29
7.4.6 DENM Format Specification . . . 29
7.5 Message Reception Functional Specification . . . 32
7.6 Path History . . . 32
7.7 Error Classification . . . 32
7.7.1 Development Errors . . . 33
7.7.2 Runtime Errors . . . 33
7.7.3 Production Errors . . . 33
7.7.4 Extended Production Errors . . . 33
7.8 Security Events . . . 33
8 API specification 34
8.1 Imported types . . . 34
8.2 Type definitions . . . 34
8.2.1 V2xFac_RxParamsType . . . 34
8.2.2 V2xFac_TxParamsType . . . 36
8.3 Function definitions . . . 37
8.3.1 V2xFac_Init . . . 37
8.3.2 V2xFac_GetVersionInfo . . . 37
8.3.3 V2xFac_V2xM_PreparePseudonymChange . . . 38
8.3.4 V2xFac_V2xM_CommitPseudonymChange . . . 39
8.3.5 V2xFac_V2xM_AbortPseudonymChange . . . 40
8.3.6 V2xFac_V2xM_SetTGenCamDcc . . . 40
8.3.7 V2xFac_V2xM_SetCaBsOperation . . . 41
8.4 Callback notifications . . . 42
8.4.1 V2xFac_RxIndication . . . 42
8.5 Scheduled functions . . . 43
8.5.1 V2xFac_CaBs_MainFunction . . . 43
8.5.2 V2xFac_DenBs_MainFunction . . . 43
8.5.3 V2xFac_RxS_MainFunction . . . 44
8.6 Expected interfaces . . . 44
8.6.1 Mandatory interfaces . . . 44
8.6.2 Optional interfaces . . . 45
8.7 Service Interfaces . . . 45

8.7.1 Sender-Receiver-Interfaces . . . 45
8.7.1.1 V2xFacVdp . . . 45
8.7.1.2 V2xFacVdpBasicLFC . . . 46
8.7.2 Client-Server-Interfaces . . . 47
8.7.2.1 V2xFacDenBs . . . 47
8.7.3 Implementation Data Types . . . 50
8.7.3.1 V2xFac specific Implementation DataTypes . . . 50
8.7.3.2 Common Implementation DataTypes . . . 52
8.7.3.3 CAM specific Implementation DataTypes . . . 57
8.7.3.4 DENM specific Implementation DataTypes . . . 67
8.7.4 Ports . . . 84
8.7.4.1 V2xFac_DenBs . . . 84
8.7.4.2 V2xFac_Vdp . . . 84
8.7.4.3 V2xFac_VdpBasicLFC . . . 85

9 Sequence diagrams 86

9.1 CAM Generation and Transmission . . . 86
9.2 DENM Generation and Transmission . . . 87
9.3 Message Reception . . . 88

10 Configuration specification 89

10.1 How to read this chapter . . . 89
10.2 Containers and configuration parameters . . . 89
10.2.1 Variants . . . 89
10.2.2 V2xFac . . . 89
10.2.3 V2xFacGeneral . . . 90
10.2.4 V2xFacConfig . . . 95
10.2.5 V2xFacDmMsgConfig . . . 95
10.3 Published Information . . . 97

A Not applicable requirements 98

B History of Specification Items 99

B.1 Specification Item History of this document compared to AUTOSAR R24-11. . . . 99
B.1.1 Added Specification Items in R24-11 . . . 99
B.1.2 Changed Specification Items in R24-11 . . . 100
B.1.3 Deleted Specification Items in R24-11 . . . 100
B.2 Specification Item History of this document compared to AUTOSAR R23-11. . . . 101
B.2.1 Added Specification Items in R23-11 . . . 101
B.2.2 Changed Specification Items in R23-11 . . . 108
B.2.3 Deleted Specification Items in R23-11 . . . 108

1 Introduction and functional overview

This document specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Vehicle-2-X Facilities (V2xFac). The Vehicle-2-X Facilities layer together with the Vehicle-2-X Basic Transport (V2xBtp), the Vehicle-2-X GeoNetworking (V2xGn), Vehicle-2-X Management (V2xM), Vehicle-2-X Data Manager (V2xDM) and the communication driver layer forms the V2X stack within the AUTOSAR architecture.

The V2xFac module is designed to be hardware independent.

The V2x Facilities layer of the ETSI architecture is split into the two AUTOSAR modules V2xFac and V2xDM.

The V2xFac module is dependent on services of V2X entities in the application layer and on lower V2xBtp module, and provides services tot the V2xDM module.

1.1 Architectural overview

Positioning of the V2xFac module within the AUTOSAR BSW and the Layered Software architecture is shown in below.

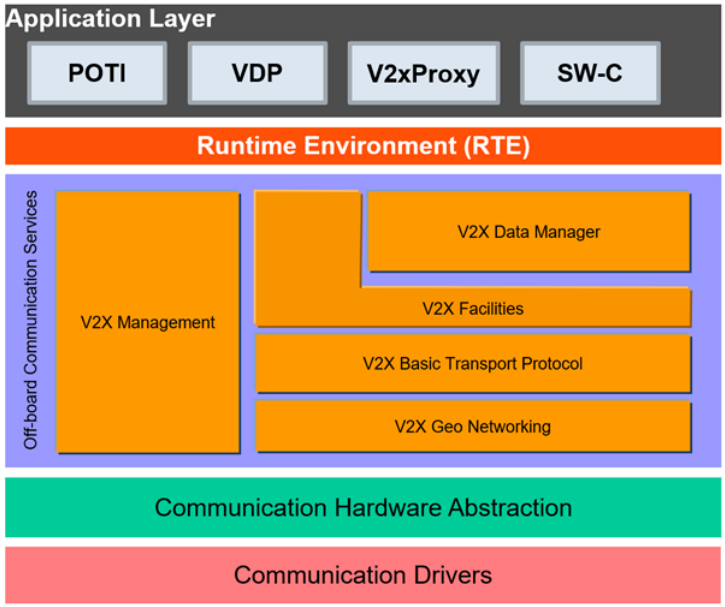

Figure 1.1: AUTOSAR BSW software architecture - V2xFac scope

The V2xFac module supports common message management for data exchange between V2X applications.

It provides the basic services (BS) Cooperative Awareness (CA) and Decentralized Environmental Notification (DEN) for transmission. In reception, it analyses the header of the received messages and provides the payload of the supported messages to the V2xDM module.

1.2 Functional overview

The V2xFac module implements the basic services CA and DEN for transmission. For reception, the header analysis is supported for the following services:

- CA
- DEN
- IVI
- RLT
- TLM.

Messages are provided to the V2XDM based on configuration.

Each type of message can be systematically discarded based on the configuration.

By default, CAM, DENM, IVIM, SPATEM and MAPEM messages can be received and their ITS header inspected before being provided to the V2xDM for disassembling. Alternatively, by configuration based on BTP port and ITS AID, other messages can be provided to the V2xDM module instead of being discarded.

1.2.1 Cooperative Awareness (CA)

1.2.1.1 CA basic service in the AUTOSAR architecture

The CA basic service is a facilities layer entity that operates the CAM protocol.

It provides two services: sending and receiving of CAMs.

The CA basic service generates and sends CAMs to other ITS-Ss or it receives CAMs from ITS-Ss and provides their payload to the V2xDM module (see [1] chapter 4).

The CA basic service uses the services provided by the protocol entities of the lower layers of the V2X stack to disseminate the CAM.

Upon receiving a CAM, the CA basic service makes the content of the CAM available to the V2xDM module.

Selected sets of data of the received CAMs are given to the upper Application layer by the V2xDM module via the standardized AUTOSAR service interface.

It may interface with the AUTOSAR application layer in order to collect relevant information for CAM generation (Vehicle Data Provider - VDP).

1.2.1.2 CA basic service functional architecture

"The CA basic service is part of the Application Support domain of the Facilities Layer according to ETSI TS 102 894-1 [2] shows the functional block diagram with the functional blocks of the CA basic service and interfaces to other facilities and layers."

For sending and receiving CAMs, the CA basic service part of the V2xFac shall provide the following sub-functions

- Encode CAM
- Decode CAM headers
- CAM transmission management
- CAM reception management

For details see [1] chapter 5.2.

1.2.2 Decentralized Environmental Notification (DEN)

1.2.2.1 DEN basic service in the AUTOSAR architecture

The DEN basic service is a facilities layer entity that operates the DENM protocol. It provides services to entities at the AUTOSAR application layer and to the V2xDM module.(refer to [3] chapter 4.2)

The DEN basic service generates and sends DENMs to other ITS-Ss or it receives DENMs from other ITS-Ss and provides their payload to the V2xDM module (see [3] chapter 5 and 6).

Upon receiving a DENM, the DEN basic service checks the headerof the DENM before providing it to the V2xDM module which makes selected content available to the V2X Applications.

1.2.2.2 DEN basic service functional architecture

For sending and receiving DENMs, the DEN basic service shall provide the following sub-functions

- Encode DEN
- Decode DEN header
- DEN transmission management
- DEN reception management
- Keep-Alive forwarding

For Details see [3] chapter 5.3. Position and Time management (POTI)

The POTI, as specified in ETSI TS 102 890-2 [4], provides the position of the ITS-S and time information.

Within the AUTOSAR architecture POTI service is a V2X Application within the Application layer and is not part of V2xFac.

For details See [3] chapter 5.1.

1.2.3 Vehicle Data Provider (VDP)

The VDP is connected with the vehicle network and provides the vehicle status information.

Within the AUTOSAR architecture VDP service is a V2X Application within the Application layer and is not part of V2xFac.

The VDP provides an interface to the lower layer (V2X Services).

The facilities basic services CA and DEN get vehicle relevant data from this interface.

The V2xM gets e.g. position and time information from this interface.

1.2.4 Local Dynamic Map (LDM)

The LDM as outlined in [5] is a database in the ITS-S, which may be updated with received CAM or DENM data.

V2x applications may retrieve information from the LDM for further processing.

Within the AUTOSAR architecture LDM service is a V2X Application within the Application layer and is not part of the V2xFac module.

For details see [5] chapter 5.1.

1.2.5 Infrastructure to Vehicle Information (IVI)

1.2.5.1 IVI service in the AUTOSAR architecture

The IVI service is a facilities layer entity that provides receiving of IVIMs.

The IVI service receives IVIMs from Infrastructure ITS-Ss and checks their header before providing their payload to the V2xDM module (see [6] chapter 7).

Upon receiving an IVIM payload, the V2xDM module makes selected content of the IVIM available to the V2X Applications.

Since these messages do not require any separate processing, their reception in AUTOSAR is implemented via a generic service (see 8.5.3).

1.2.5.2 IVI service functional architecture

The IVI service is part of the Application Support domain of the Facilities Layer according to ETSI TS 103 301 [6] which shows the functional block diagram with the functional blocks of the IVI service and interfaces to other facilities and layers.

For receiving IVIMs, the IVI service part of the V2xFac shall provide the following sub-functions:

- Decode IVIM header
- IVIM reception management

1.2.6 Road and Lane Topology (RLT) service

1.2.6.1 RLT service in the AUTOSAR architecture

The RLT service is a facilities layer entity that provides receiving of MAPEMs.

The RLT service receives MAPEMs from Infrastructure ITS-Ss and checks their header before providing their payload to the V2xDM module (see [6] chapter 6).

Upon receiving a MAPEM payload, the V2XDM module makes selected content of the MAPEM available to the V2X Applications.

Since these messages do not require any separate processing, their reception in AUTOSAR is implemented via a generic service (see 8.5.3).

1.2.6.2 RLT service functional architecture

The RLT service is part of the Application Support domain of the Facilities Layer according to ETSI TS 103 301 [6] shows the functional block diagram with the functional blocks of the RLT services and interfaces to other facilities and layers.

For receiving MAPEMs, the RLT service part of the V2xFac shall provide the following sub-functions:

- Decode MAPEM header
- MAPEM reception management

1.2.7 Traffic Light Maneuver (TLM) service

1.2.7.1 TLM service in the AUTOSAR architecture

The TLM service is a facilities layer entity that provides receiving of SPATEMs.

The TLM service receives SPATEMs from Infrastructure ITS-Ss and checks their header before providing their payload to the V2xDM module (see [6] chapter 5).

Upon receiving a SPATEM payload, the V2xDM module makes selected content of the SPATEM available to the V2X Applications.

Since these messages do not require any separate processing, their reception in AUTOSAR is implemented via a generic service (see 8.5.3).

1.2.7.2 TLM service functional architecture

The TLM service is part of the Application Support domain of the Facilities Layer according to ETSI TS 103 301 [6] which shows the functional blocks of the TLM services and interfaces to other facilities and layers.

For receiving SPATEMs, the TLM service part of the V2xFac shall provide the following sub-functions

Decode SPATEM header

SPATEM reception management

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the V2X Facilities module that are not included in the AUTOSAR glossary [7].

Abbreviation / Acronym:	Description:
DEM	Diagnostic Event Manager
DET	Default Error Tracer
API	Application Programming Interface
BS	Basic Service
BSW	Basic Software
BTP	Basic Transport Protocol
CA	Cooperative Awareness
CAM	Cooperative Awareness Message
DCC	Decentralized Congestion Control
DE	Data Element
DEN	Decentralized Environmental Notification
DENM	Decentralized Environmental Notification Messages
DF	Data Frame
EcuM	Electronic Control Unit Manager
ETSI	European Telecommunications Standards Institute
IF	Interface
ITS	Intelligent Transport System
ITS-S	ITS-Station
KAF	DENM Keep Alive Forwarding
LDM	Local Dynamic Map
POTI	Position and Time management
RSU	Road Side Unit
VDP	Vehicle Data Provider
V2X	Either vehicle to vehicle (V2V), or vehicle to infrastructure (V2I) and/or infrastructure to vehicle (I2V)
V2xM	Vehicle-2-X Management
V2xFac	Vehicle-2-X Facilities
V2xBtp	Vehicle-2-X Basic Transport
V2xGn	Vehicle-2-X Geo Networking
IVI	Infrastructure to Vehicle Information
IVIM	Infrastructure to Vehicle Information Message
RLT	Road and Lane Topology
MAPEM	MAP Extended Message
TLM	Traffic Light Maneuver
SPATEM	Signal Phase And Timing Extended Message

Table 2.1: Acronyms and abbreviations used in the scope of this Document

3 Related documentation

3.1 Input documents & related standards and norms

[1] EN 302 637-2 V1.4.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service

[2] TS 102 894-1 V1.1.1: Intelligent Transport Systems (ITS); Users and applications requirements; Part 1: Facility layer structure, functional requirements and specifications

[3] EN 302 637-3 V1.3.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service

[4] EN 302 890-2 v0.0.3: Intelligent Transport System (ITS); Facilities layer function; Part 2: Position and Time management (PoTi); Release 2

[5] EN 302 895 Ver. 1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)

[6] TS 103 301 V1.2.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services

[7] Glossary
AUTOSAR_FO_TR_Glossary

[8] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[9] Specification of Default Error Tracer
AUTOSAR_CP_SWS_DefaultErrorTracer

[10] Specification of ECU State Manager
AUTOSAR_CP_SWS_ECUStateManager

[11] EN 302 636-5-1 V2.1.1: Vehicular Communication; Geonetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocols

[12] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[13] Requirements on Vehicle-2-X Communication
AUTOSAR_CP_RS_V2XCommunication

[14] TS 102 894-2 V1.3.1: Intelligent Transport Systems (ITS); Users and applications requirements; Applications and facilities layer common data dictionary

[15] Car 2 Car Communication Consortium; Basic System Profile release 1.3

[16] SAE J2945/1_201603: On-Board System Requirements for V2V Safety Communications

[17] EN 302 636-4-1 V1.3.1: Vehicular Communication; Geonetworking; Part 4 Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [8], which is also valid for V2xFac.

Thus, the specification SWS BSW General shall be considered as additional and required specification for V2xFac.

4 Constraints and assumptions

4.1 Limitations

- The V2X modules follow the guidance regarding the Day-1 scenarios defined by Car-2-Car-Consortium and C-Roads platform.
- The V2xFac module is only relevant for European Day1 V2X implementation.

4.2 Applicability to car domains

This specification is applicable to all car domains.

4.3 Authorisation Tickets and Pseudonyms

The Authorisation Ticket (AT) is referred to as Pseudonym in this document.

5 Dependencies to other modules

This section describes the relations of the V2xFac module to other modules within the AUTOSAR basic software architecture. It outlines the modules that are required or optional for the realization of the V2xFac module and the V2xFac services that these modules use.

5.1 AUTOSAR DET (Default Error Tracer)

In development mode, the V2xFac module reports errors through the Det_ReportError function of the DET Module [9].

5.2 AUTOSAR EcuM (Ecu State Manager)

The EcuM [10] initializes the V2xFac module by calling V2xFac_Init specified in 8.3.1.

5.3 V2x Vehicle Data Provider

The V2xFac module retrieves vehicle relevant data from the VDP application by using the Sender-Receiver-Interface V2xFacVdp (see [SWS_V2xFac_00094]).

5.4 AUTOSAR V2xDM

The V2xFac module provides the Client-Server-Interface V2xFacDenBs for using the DEN basic service. The operations TriggerEvent, UpdateEvent or TerminateEvent are provided.

After having checked the validity of their header, the V2xFac module delivers the payload of the received messages to the V2xDM module. The V2xFac module assumes an interface for message reception (see [SWS_V2xFac_00092]) to be provided by the Vehicle-2-X Data Manager (V2xDM) module.

5.5 AUTOSAR V2xBtp

The V2xFac module assumes a transmit request primitive (V2xBtp_Transmit [11], see [SWS_V2xFac_00092]) to be provided by the V2xBtp module.

5.6 AUTOSAR V2xM

The V2xFac module assumes a request primitive (see [SWS_V2xFac_00092]) to be provided by the Vehicle-2-X Management (V2xM) module.

6 Requirements Tracing

The following tables reference the requirements specified in [12] and [13] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_V2xFac_00106]
[SRS_BSW_00345]	BSW Modules shall support pre-compile configuration	[SWS_V2xFac_00238]
[SRS_V2X_00010]	The implementation of the V2X system shall follow additional guidance given by C2C-CC requirements	[SWS_V2xFac_00004] [SWS_V2xFac_00008] [SWS_V2xFac_00010] [SWS_V2xFac_00014] [SWS_V2xFac_00019] [SWS_V2xFac_00116] [SWS_V2xFac_20168] [SWS_V2xFac_20185] [SWS_V2xFac_20215] [SWS_V2xFac_20256] [SWS_V2xFac_20257] [SWS_V2xFac_20297] [SWS_V2xFac_20313]
[SRS_V2X_00190]	The V2X system shall handle vehicle states in a consistent manner	[SWS_V2xFac_20444] [SWS_V2xFac_20445]
[SRS_V2X_00214]	The V2X system shall allow applications to deactivate transmission of CAMs	[SWS_V2xFac_00006]
[SRS_V2X_00232]	The V2X system shall cooperate with tolling zone stations in vicinity	[SWS_V2xFac_00307]
[SRS_V2X_00259]	The V2X system shall manage the life time of all DENM packets	[SWS_V2xFac_20259]
[SRS_V2X_00291]	The V2X system shall only send messages with valid postion and time	[SWS_V2xFac_20215] [SWS_V2xFac_20291]
[SRS_V2X_00301]	The V2X system’s Facility Layer shall handle DENM repetition	[SWS_V2xFac_00029]
[SRS_V2X_00318]	The V2X system’s Facility Layer shall generate traces and path histories	[SWS_V2xFac_20318]
[SRS_V2X_00693]	The V2X system shall provide functionality for generating traces and path histories	[SWS_V2xFac_20285] [SWS_V2xFac_20286] [SWS_V2xFac_20287] [SWS_V2xFac_20288] [SWS_V2xFac_20289] [SWS_V2xFac_20302] [SWS_V2xFac_20303] [SWS_V2xFac_20304] [SWS_V2xFac_20305] [SWS_V2xFac_20306] [SWS_V2xFac_20307] [SWS_V2xFac_20308]
[SRS_V2X_00711]	The V2X system’s CA basic service shall be compliant to ETSI Specification of Cooperative Awareness Basic Service	[SWS_V2xFac_00001] [SWS_V2xFac_00004] [SWS_V2xFac_00008] [SWS_V2xFac_00009] [SWS_V2xFac_00010] [SWS_V2xFac_00011] [SWS_V2xFac_00014] [SWS_V2xFac_00015] [SWS_V2xFac_00016] [SWS_V2xFac_00019] [SWS_V2xFac_00116] [SWS_V2xFac_00231] [SWS_V2xFac_00247] [SWS_V2xFac_00248] [SWS_V2xFac_00294] [SWS_V2xFac_00295] [SWS_V2xFac_00296] [SWS_V2xFac_00301] [SWS_V2xFac_00303] [SWS_V2xFac_00306] [SWS_V2xFac_20292] [SWS_V2xFac_20297]
[SRS_V2X_00741]	The V2X system’s DEN basic service shall be compliant to ETSI Specifications of Decentralized Environmental Notification Basic Service	[SWS_V2xFac_00001] [SWS_V2xFac_00004] [SWS_V2xFac_00025] [SWS_V2xFac_00027] [SWS_V2xFac_00232] [SWS_V2xFac_00247] [SWS_V2xFac_00248] [SWS_V2xFac_00301] [SWS_V2xFac_00303] [SWS_V2xFac_00306]

▽

△

Requirement	Description	Satisfied by
[SRS_V2X_10001]	The V2X system's Facility layer shall support receiving IVI messages	[SWS_V2xFac_00247] [SWS_V2xFac_00248] [SWS_V2xFac_00301] [SWS_V2xFac_00303] [SWS_V2xFac_00305] [SWS_V2xFac_00306]
[SRS_V2X_10003]	The V2X system's Facility layer shall support receiving MAPEM messages	[SWS_V2xFac_00247] [SWS_V2xFac_00248] [SWS_V2xFac_00301] [SWS_V2xFac_00303] [SWS_V2xFac_00306]
[SRS_V2X_10004]	The V2X system's Facility layer shall support receiving SPAT extended messages	[SWS_V2xFac_00247] [SWS_V2xFac_00248] [SWS_V2xFac_00301] [SWS_V2xFac_00303] [SWS_V2xFac_00306]
[SRS_V2X_26001]	The V2X system shall provide selected information from a V2X message to the application layer and/ or to the vehicle network.	[SWS_V2xFac_00300]

Table 6.1: Requirements Tracing

Note:

Requirement IDs within this document have an encoding to state where each requirement has its origin:

- SWS items starting with a leading 0 (SWS_V2xFac_0xxxx) are module specific and not inherited.
- SWS items starting with a leading 2 (SWS_V2xFac_2xxxx) are inherited from C2C-CC Basic System Profile

7 Functional specification

The V2xFac module operates the transmission part of the basic services Cooperative Awareness (CA) and Decentralized Environmental Notification (DEN).

In reception, it checks the validity of the header of the received messages before providing their payload to the V2xDM module.

[SWS_V2xFac_00231]

Upstream requirements: SRS_V2X_00711

⌈The V2xFac module shall implement the transmission part of CA Basic Service as specified in [1] unless specified otherwise in this document⌋

[SWS_V2xFac_00232]

Upstream requirements: SRS_V2X_00741

⌈The V2xFac module shall implement the transmission part of DEN Basic Service as specified in [3] unless specified otherwise in this document⌋

[SWS_V2xFac_00300]

Upstream requirements: SRS_V2X_26001

⌈The V2xFac module shall check the validity of the header of the received Facilities messages based on its configuration which shall define the association of Facilities message ID in the ITS header and BTP port.

Incoming messages are passed to the V2xDm if configured. The V2xFac is responsible to ensure that a message is discarded if the receiving port does not match the expected message ID given via configuration.

Dedicated services for certain messages (such as DenBs) might define additional validity checks for the incoming messages. The V2xFac is responsible to ensure that a message is discarded if it is categorized as invalid by its dedicated service.

Additional verifications on ITS header fields such as for instance on Station ID in the scope of misbehavior detection or the consistence with supported protocol versions are also possible and left to implementors decision⌋

[SWS_V2xFac_00301]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00741, SRS_V2X_10001, SRS_V2X_10003, SRS_V2X_10004

⌈The V2xFac module shall check the validity of the header of received CAM, DENM, IVIM, MAPEM and SPATEM as specified in [1], [3] and [6] unless specified otherwise in this document.

Facilities messages received on standardized ports as specificied in [SWS_V2xFac_00247] shall respect the one on one mapping between ports and Message IDs or be discarded.⌋

[SWS_V2xFac_00247]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00741, SRS_V2X_10001, SRS_V2X_-10003, SRS_V2X_10004

⌈

Standardised BTP port number	Service	Message ID in ITS header	Rx/Tx
2001	CA	CAM	Rx and Tx
2002	DEN	DENM	Rx and Tx
2003	RLT	MAPEM	Rx only
2004	TLM	SPATEM	Rx only
2006	IVI	IVIM	Rx only

The V2xFac module shall only support these messages in association to these BTP ports.

⌋

[SWS_V2xFac_20444]

Upstream requirements: SRS_V2X_00190

⌈For a stationary vehicle, the system shall report the last estimated heading value.⌋

[SWS_V2xFac_20445]

Upstream requirements: SRS_V2X_00190

⌈At system shutdown, the system shall store the last heading value and the corresponding gear position (forward, neutral or backward).

At system start-up, the system shall report the heading value based on this stored heading value and the current gear position, until the vehicle is no longer stationary.⌋

7.1 Startup behavior

[SWS_V2xFac_00001]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00741

⌈The function V2xFac_Init (refer to chapter 8.3.2) of the V2xFac shall initialize

the internal states of the V2xFac module.⌋

Note: The function V2xFac_Init shall not be called before the Vehicle-2-X Management (V2xM) is initialized by the Electronic Control Unit Manager (EcuM).

[SWS_V2xFac_00004]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00741, SRS_V2X_00010

⌈The function V2xFac_Init shall initialize the basic services CA and DEN for transmission and the reception service for all message types as per configuration.⌋

7.2 General Format Specification

[SWS_V2xFac_20313]

Upstream requirements: SRS_V2X_00010

⌈The data elements which constitute the content of the CAM and DENM shall be compliant to [14].⌋

[SWS_V2xFac_00248]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00741, SRS_V2X_10001, SRS_V2X_10003, SRS_V2X_10004

⌈The header format of received messages shall be compliant to [14].⌋

7.3 CA Functional Specification

For details see [1] chapter 6.1.

7.3.1 CA Initialization, Activation and Deactivation

[SWS_V2xFac_00116]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00010

⌈The path history shall be cleared when the sending functionality is enabled via the V2xFac_V2xM_SetCaBsOperation API.⌋

[SWS_V2xFac_00006]

Upstream requirements: SRS_V2X_00214

⌈CA basic service initialization shall enable the transmission of CAM messages.⌋

[SWS_V2xFac_00008]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00010

⌈The function V2xFac_Init shall initialize the parameter T_GenCam_DCC [1] needed for the frequency management for CAMs according to T_GenCamMax [1].

For details see [1] chapter 5.3.5 and [15].⌋

[SWS_V2xFac_00009]

Upstream requirements: SRS_V2X_00711

⌈The function V2xFac_Init shall initialize the parameter T_GenCam [1] to the default value T_GenCamMax.

For details see [1] chapter 6.1.3⌋

[SWS_V2xFac_00010]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00010

⌈The function V2xFac_Init shall initialize the parameter N_GenCam [1] to the default value 0.⌋

[SWS_V2xFac_00011]

Upstream requirements: SRS_V2X_00711

⌈The function V2xFac_Init shall initialize the parameter T_CheckCamGen [1] to the default value equal to the configuration parameter T_GenCamMin [1].

For details see [1] chapter 6.1.3⌋

7.3.2 CAM Generation, Sending and Frequency Management

For CAM reception See 7.5.

[SWS_V2xFac_00014]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00010

⌈The CA basic service shall periodically generate CAMs controlled by a CAM frequency management (For details see [1] chapter 6.1.3 and [14].)⌋

[SWS_V2xFac_00015]

Upstream requirements: SRS_V2X_00711

⌈The generated CAMs shall be transmitted by the V2xBtp using the API function V2xBtp_Transmit() (see [SWS_V2xFac_00092]).⌋

[SWS_V2xFac_00016]

Upstream requirements: SRS_V2X_00711

⌈The CA basic service shall receive CAMs via the callback function `V2xFac_RxIndication()`.⌋

[SWS_V2xFac_00294]

Upstream requirements: SRS_V2X_00711

⌈The MAX_DANGLE [15] representing the delta angle (in degrees) between two generation rules checks shall use a value of 4°.⌋

[SWS_V2xFac_00295]

Upstream requirements: SRS_V2X_00711

⌈The MAX_DDISTANCE [15] representing the delta distance (in meters) between two generation rules checks shall use a value of 4 meters.⌋

[SWS_V2xFac_00296]

Upstream requirements: SRS_V2X_00711

⌈The MAX_DSPEED [15] representing the delta speed between two generation rules checks shall use a value of 0,5 m/s.⌋

[SWS_V2xFac_20297]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00010

⌈The adjustable N_GenCam parameter (see [1]) specified in the CAM Generation Frequency Management shall be set to 3 for the V2xFac module.⌋

[SWS_V2xFac_20291]

Upstream requirements: SRS_V2X_00291

⌈The V2xFac module shall transmit CAM messages as long as position and time information are available.⌋

7.3.3 CAM Generation Frequency Management for RSU ITS-Ss

Generation of CA messages for road side units (RSU-ITS) is currently not supported by AUTOSAR.

7.3.4 CAM Time Requirement

[SWS_V2xFac_00019]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00010

⌈The CAM generation shall follow time requirements according to [1] chapter 6.1.5 and [15].⌋

[SWS_V2xFac_20168]

Upstream requirements: SRS_V2X_00010

⌈The V2xFac module shall check the timestamp in the security envelope compared to the reception time and accept only CAMs in the last time of 2 seconds and other messages within the last time of 10 minutes.

Due to the tolerance of the ITS station times and allowed clock deviation in [15], the V2xFac module shall accept messages coming from the future compared to ego vehicle clock:

- up to a maximum of 40 ms for vehicles (20 ms estimated deviation from ego vehicle + 20 ms deviation for transmitting vehicle).
- up to a maximum of 220 ms for RSUs (20 ms estimated deviation from ego vehicle + 200 ms deviation for transmitting RSU).

⌋

7.3.5 CAM Format Specification

For details about CAM data format refer to to the following ETSI documents:

See [1] chapter 7

See [1] Annex A: ASN.1 specification of CAM

See [1] Annex B: Description for data elements and data frames

See [14] Annex A, Annex B

[SWS_V2xFac_20285]

Upstream requirements: SRS_V2X_00693

⌈The path history field inside the CAM low frequency (LF) container shall contain a PathHistory data element covering a distance of at least 200 m (K_PHDISTANCE_M parameter in [16], Appendix A.5).

An exception to the minimum covered distance by PathHistory shall be only made if either of the following conditions is fulfilled:

- the vehicle has not yet physically covered the distance with its current pseudonym (e.g., after vehicle startup or right after pseudonym change when driving)
- the maximum number of PathPoints is used while the overall length covered by the PathHistory still does not reach 200m.

Only in the above two cases the vehicle may send PathHistory information covering a distance below the 200 m lower limit.⌋

[SWS_V2xFac_20286]

Upstream requirements: SRS_V2X_00693

⌈The PathHistory in CAMs shall cover at most 500 m.⌋

[SWS_V2xFac_20287]

Upstream requirements: SRS_V2X_00693

⌈The V2xFac module shall send PathDeltaTime in every PathPoint of the PathHistory. Therefore, the PathHistory shall describe a time-ordered list (newest point first) of actually travelled geographical locations, including current ego position.⌋

[SWS_V2xFac_20288]

Upstream requirements: SRS_V2X_00693

⌈In cases where the vehicle does not move, i.e. PathPoint position information does not change, the PathDeltaTime of the first PathPoint shall still be updated with every CAM.⌋

[SWS_V2xFac_20289]

Upstream requirements: SRS_V2X_00693

⌈When the V2xFac module is stationary for a duration longer than the maximum value of PathDeltaTime (specified in [14]) the PathDeltaTime of the first PathPoint in the CAM shall be fixed to the maximum value.⌋

[SWS_V2xFac_20292]

Upstream requirements: SRS_V2X_00711

⌈The traffic class value for CAM messages shall be set to 2.⌋

[SWS_V2xFac_20256]

Upstream requirements: SRS_V2X_00010

⌈The V2xFac module shall use a Single Hop Broadcast (SHB) header on all CAM packets it sends. Therefore, the value of the transportType parameter shall be set to 0x50⌋

7.3.6 Others

[SWS_V2xFac_00307]

Upstream requirements: SRS_V2X_00232

⌈Upon receiving a CAM message, the V2xFac module checks the station type of the emitter of this CAM message.

As an exception to the normal message handling, if the station type in the CAM message indicates an RSU, before discarding the message, the V2xFac module shall provide the position and the radius of the protected zone to the V2xM module through the V2xM_SetTollingZoneInformation API.

If several CAM messages from the same RSU (same station Id) are received in sequence, the CAM messages shall be discarded without calling again the V2xM_SetTollingZoneInformation API.⌋

7.4 DEN Functional Specification

As defined in ETSI documents (See [3] chapter 5.2) the DEN basic service is a facilities layer entity that implements the DEN protocol. It interfaces with ITS-S applications in order to receive the application request for DENM transmission and to provide the received DENM content to the ITS-S applications.

7.4.1 DEN Initialization

[SWS_V2xFac_00025]

Upstream requirements: SRS_V2X_00741

⌈The function V2xFac_Init shall initialize an empty originating ITS-S message table.

For details see [3] chapter 8.2.1.6⌋

7.4.2 DENM Transmission Management

[SWS_V2xFac_00027]

Upstream requirements: SRS_V2X_00741

⌈The DEN basic service is triggered by the V2x-Application via its service operations TriggerEvent, UpdateEvent or TerminateEvent from the service interface `V2xFacDenBs`.

The function parameter "EventID" given by the above mentioned operations shall be mapped by the DEN basic service to the actionID generated for DENMs.

For details see [3] chapter 5.3 and 8.2⌋

7.4.3 DENM Reception Management

See 7.5.

7.4.4 DENM Repetition

[SWS_V2xFac_00029]

Upstream requirements: SRS_V2X_00301

⌈In between two consequent DENM updates, a DENM may be repeated by the DEN basic service.

For details see [3] chapter 6.1.2.3⌋

7.4.5 DENM Keep Alive Forwarding (KAF)

KAF functionality for the DEN basic service as defined by ETSI is not supported.

See [3] chapter 5.3 and 8.3

7.4.6 DENM Format Specification

For details about DENM data format refer to to the following ETSI documents:

See [3] chapter 7,

See [3] Annex A: ASN.1 specification of DENM

See [3] Annex B: Description for data elements and data frames

See [14] Annex A, Annex B

[SWS_V2xFac_20302]

Upstream requirements: SRS_V2X_00693

⌈The path history field inside the DEN messages shall contain Trace data elements covering a distance of at least 600 m (K_PHDISTANCE_M parameter in [16], Appendix A.5).

An exception to the minimum covered distance by Traces shall be only made if either of the following conditions is fulfilled:

- the vehicle has not yet physically covered the distance with its current pseudonym (e.g., after vehicle startup or right after pseudonym change when driving)
- the maximum number of PathPoints is used while the overall length covered by the PathHistory still does not reach 200m.

Only in the above two cases the vehicle may send Traces information covering a distance below the 600 m lower limit.⌋

[SWS_V2xFac_20303]

Upstream requirements: SRS_V2X_00693

⌈The Traces in the DENMs shall cover at most 1000 m.⌋

[SWS_V2xFac_20304]

Upstream requirements: SRS_V2X_00693

⌈The V2xFac module shall use the DENM traces as follow: The PathDeltaTime shall be sent in every PathPoint in the first DENM traces element. Therefore, the first element of the traces shall describe a time-ordered list (newest point first) of actually travelled geographical locations leading to the event position, including event position.⌋

[SWS_V2xFac_20305]

Upstream requirements: SRS_V2X_00693

⌈The PathDeltaTime data elements of the PathPoints in the first DENM traces element shall only be updated if the DENM is updated.⌋

[SWS_V2xFac_20306]

Upstream requirements: SRS_V2X_00693

⌈In cases where the event detecting vehicle does not move, i.e. PathPoint position information does not change, the PathDeltaTime of the first PathPoint of the first DENM traces element shall still be updated with every DEN_Update.⌋

NOTE: This is only the case for stationary events where the detecting vehicle is identical to the event, e.g. a stationary vehicle warning. For dynamic events, e.g. dangerous situations, or events, where the event is not identical to the vehicle, e.g. adverse weather warning, this is not the case.

[SWS_V2xFac_20307]

Upstream requirements: SRS_V2X_00693

⌈When the V2xFac module is stationary for a duration longer than the maximum value of PathDeltaTime (specified in [14]) the PathDeltaTime of the first PathPoint in the DENM shall be set to this maximum value and a new PathPoint shall be created.⌋

[SWS_V2xFac_20308]

Upstream requirements: SRS_V2X_00693

⌈Additional PathHistory elements may be present in the DENM traces. However, unlike the first element, these shall describe alternative routes to the event location. These routes may or may not be available at the time of detecting the event. In the alternative routes, the PathPoints shall be position-ordered (i.e. shortest-path routes) and they shall not include the PathDeltaTime.⌋

[SWS_V2xFac_20318]

Upstream requirements: SRS_V2X_00318

⌈The traces and path histories used by the V2xFac module shall be generated using the Design Method One as specified in [16], Appendix A.5.

The V2xFac module shall use the generation method with the following settings:

- *K_PHALLOWABLEERROR_M = 0.47 m*, where *PH_ActualError < K_PHALLOWABLEERROR_M*
- Maximum distance between concise path points, *K_PH_CHORDLENGTHTHRESHOLD = 22.5 m/s*
- *K_PH_MAXESTIMATEDRADIUS = REarthMeridian*
- *K_PHSMALLDELTAPHI_R = 1 degree*
- *REarthMeridian = 6378.137 km*

(according to IUGG - International Union of Geodesy and Geophysics), used for great-circle or orthodromic distance calculation:

$$PH\_ActualChordLength = REarthMeridian * \cos^{-1}[\cos(lat1)\cos(lat2)\cos(long1 - long2) + \sin(lat1)\sin(lat2)]$$

⌋

[SWS_V2xFac_20257]

Upstream requirements: SRS_V2X_00010

⌈The V2xFac module shall use GeoBroadcast (GBC) headers on all DENM packets it sends. Therefore, the value of the transportType parameter shall be set to 0x40⌋

[SWS_V2xFac_20259]

Upstream requirements: SRS_V2X_00259

⌈The V2xFac module shall set the maxPacketLifetime parameter of the packets transport parameters TxParams of all GBC packets to the minimum of ValidityDuration, RepetitionInterval and itsGnMaxPacketLifetime, with ValidityDuration and RepetitionInterval values as requested by the application and itsGnMaxPacketLifetime value as specified in [17], Annex H.⌋

7.5 Message Reception Functional Specification

[SWS_V2xFac_00303]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00741, SRS_V2X_10001, SRS_V2X_10003, SRS_V2X_10004

⌈Upon receiving a message, the V2xFac module checks the validity and the consistency of the ITS header of the received messages, and, if the header is consistent and valid, makes the payload of the message available to the V2xDM module.

The payload of the received messages shall be sent to the V2xDM module via the callback function V2xDM_V2xStackRxIndication().⌋

7.6 Path History

[SWS_V2xFac_20185]

Upstream requirements: SRS_V2X_00010

⌈Facilities layer shall clear the own station's path history cache (used to fill into new messages) when the security entity changes its pseudonym identity.⌋

[SWS_V2xFac_20215]

Upstream requirements: SRS_V2X_00010, SRS_V2X_00291

⌈Traces and path history data shall only be generated when position confidence and ITS time information are available⌋

7.7 Error Classification

Section "Error Handling" of the document [8] "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it

constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

7.7.1 Development Errors

[SWS_V2xFac_00106]

Upstream requirements: SRS_BSW_00323

⌈In case development error detection is enabled for the V2xFac module, the V2xFac module shall check API parameters for validity and report detected errors to the DET.⌋

[SWS_V2xFac_00031] Definiton of development errors in module V2xFac ⌈

Type of error	*Related error code*	*Error value*
API service called with wrong parameter	V2XFAC_E_PARAM	0x01
API service called with invalid pointer	V2XFAC_E_PARAM_POINTER	0x02
V2xFac initialization failed	V2XFAC_E_INIT_FAILED	0x03
API function called before the V2xFac module has been fully initialized	V2XFAC_E_UNINIT	0x04

⌋

7.7.2 Runtime Errors

There are no runtime errors.

7.7.3 Production Errors

There are no production errors.

7.7.4 Extended Production Errors

There are no extended production errors.

7.8 Security Events

The module does not report security events.

8 API specification

8.1 Imported types

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

[SWS_V2xFac_00032] Definition of imported datatypes of module V2xFac ⌈

Module	*Header File*	*Imported Type*
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType
V2x_GeneralTypes	Rte_V2xM_Type.h	V2xM_PositionAndTimeType
	V2x_GeneralTypes.h	V2x_GnAddressType
	V2x_GeneralTypes.h	V2x_GnAreaShapeType
	V2x_GeneralTypes.h	V2x_GnDestinationAreaType
	V2x_GeneralTypes.h	V2x_GnDestinationType
	V2x_GeneralTypes.h	V2x_GnLongPositionVectorType
	V2x_GeneralTypes.h	V2x_GnPacketTransportType
	V2x_GeneralTypes.h	V2x_GnUpperProtocolType
	V2x_GeneralTypes.h	V2x_PseudonymType
	V2x_GeneralTypes.h	V2x_SecProfileType
	V2x_GeneralTypes.h	V2x_SecReportType
	V2x_GeneralTypes.h	V2x_TrafficClassIdType
V2xBtp	V2xBtp.h	V2xBtp_TxParamsType

⌋

8.2 Type definitions

8.2.1 V2xFac_RxParamsType

[SWS_V2xFac_00034] Definition of datatype V2xFac_RxParamsType ⌈

Name	V2xFac_RxParamsType	
Kind	Structure	
Elements	destinationPort	
	Type	uint16
	Comment	Identifies the protocol entity at the ITS facilities layer at the destination of a BTP packet.
	destinationAddress	
	Type	V2x_GnAddressType

▽

△

	Comment	Destination address for GeoUnicast packet
	destinationArea	
	Type	V2x_GnDestinationAreaType
	Comment	Destination area for GeoBroadcast/GeoAnycast packet.
	destinationType	
	Type	V2x_GnDestinationType
	Comment	Select which destination type (destinationAddress or destinationArea is used for this packet).
	sourcePositionVector	
	Type	V2x_GnLongPositionVectorType
	Comment	Geographical position for the source of the received GeoNetworking packet.
	securityReport	
	Type	V2x_SecReportType
	Comment	Result information from the security operations for decryption and verification. This parameter is supplied by the V2xM module and forwarded up to the ITS Facilities layer passing through the Geo Networking and BTP layers.
	certificateId	
	Type	uint64
	Comment	Identification of source certificate, for example the certificate hash. This parameter is supplied by the V2xM and forwarded up to the ITS Facilities layer passing through the GeoNetworking and BTP layers.
	sspBits	
	Type	Array of uint8
	Size	4
	Comment	Sender permissions
	sspLength	
	Type	uint8
	Comment	Sender permissions length
	trafficClass	
	Type	V2x_TrafficClassIdType
	Comment	Traffic class, with which the GeoNetworking packet was generated by the source.
	remPacketLifetime	
	Type	uint16
	Comment	Remaining lifetime of the packet in [s].
	itsAid	
	Type	uint32
	Comment	The numerical value of the ITS-AID (Application Identifier).
Description	Wraps GeoNetworking parameters from V2xBtp	
Available via	V2xFac.h	

⌋

8.2.2 V2xFac_TxParamsType

temporary addition

[SWS_V2xFac_00069] Definition of datatype V2xFac_DenmMessageRootType ⌈

Name	V2xFac_DenmMessageRootType	
Kind	Structure	
Elements	itsPduHeader	
	Type	V2xFac_ItsPduHeaderType
	Comment	Structure of the ItsPduHeader
	denm	
	Type	V2xFac_DenMsgType
	Comment	Structure of the DEN data
Description	DENM root message as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
Available via	Rte_V2xFac_Type.h	

⌋

temporary addition

[SWS_V2xFac_00036] Definition of datatype V2xFac_ItsPduHeaderType ⌈

Name	V2xFac_ItsPduHeaderType	
Kind	Structure	
Elements	protocolVersion	
	Type	uint8
	Comment	Version of ITS message and/or communication protocol
	messageId	
	Type	uint8
	Comment	Type of the ITS message.
	stationId	
	Type	uint32
	Comment	Identifier of originating ITS-S
Description	DF_ItsPduHeader as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Available via	Rte_V2xFac_Type.h	

⌋

8.3 Function definitions

8.3.1 V2xFac_Init

[SWS_V2xFac_00082] Definition of API function V2xFac_Init ⌈

Service Name	V2xFac_Init	
Syntax	`void V2xFac_Init ( void* CfgPtr )`	
Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CfgPtr	Points to a null pointer.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Initializes the V2xFac module.	
Available via	V2xFac.h	

⌋

8.3.2 V2xFac_GetVersionInfo

[SWS_V2xFac_00084] Definition of API function V2xFac_GetVersionInfo ⌈

Service Name	V2xFac_GetVersionInfo	
Syntax	`void V2xFac_GetVersionInfo ( Std_VersionInfoType* VersionInfoPtr )`	
Service ID [hex]	0x02	
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.	
Available via	V2xFac.h	

⌋

[SWS_V2xFac_00085] ⌈If V2xFacDevErrorDetect is enabled:
If the VersionInfoPtr pointer parameter is invalid (e.g. NULL), the error-code V2XFAC_E_PARAM_POINTER shall be reported to the DET module.⌋

8.3.3 V2xFac_V2xM_PreparePseudonymChange

[SWS_V2xFac_00086] Definition of API function V2xFac_V2xM_Prepare PseudonymChange ⌈

Service Name	V2xFac_V2xM_PreparePseudonymChange	
Syntax	`Std_ReturnType V2xFac_V2xM_PreparePseudonymChange ( const V2x_PseudonymType* PseudonymPtr )`	
Service ID [hex]	0x03	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	PseudonymPtr	The Pseudonym provided by V2xM
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: operation successful `E_NOT_OK`: pseudonym change rejected
Description	By this API primitive the V2xFac module gets an indication that the given Pseudonym and hereby the StationId is about to be changed	
Available via	V2xFac_V2xM.h	

⌋

[SWS_V2xFac_00136] ⌈The function V2xFac_V2xM_PreparePseudonymChange shall prepare the setting of the pseudonym specific part of the StationId being used for packet transmission.⌋

[SWS_V2xFac_00137] ⌈If development error detection is enabled: the function shall check that the service V2xFac_Init was previously called. If the check fails, the function shall raise the development error V2XFAC_E_UNINIT.⌋

[SWS_V2xFac_00138] ⌈If development error detection is enabled: the function shall check the parameter PseudonymPtr for being valid. If the check fails, the function shall raise the development error V2XFAC_E_PARAM_POINTER.⌋

8.3.4 V2xFac_V2xM_CommitPseudonymChange

[SWS_V2xFac_00140] Definition of API function V2xFac_V2xM_CommitPseudonymChange ⌈

Service Name	V2xFac_V2xM_CommitPseudonymChange	
Syntax	`Std_ReturnType V2xFac_V2xM_CommitPseudonymChange ( void )`	
Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: operation successful `E_NOT_OK`: operation failed
Description	This function is called by the V2xM when all modules are OK with the pseudonym change and the change is to be committed.	
Available via	V2xFac_V2xM.h	

⌋

[SWS_V2xFac_00141] ⌈The function V2xFac_V2xM_CommitPseudonymChange shall set the pseudonym specific part of the GeoNetworking Address being used for packet transmission and clean the path history. V2xFac shall store the access of the GeoNetworking Address for subsequent API calls.⌋

[SWS_V2xFac_00142] ⌈If development error detection is enabled: the function shall check that the service V2xFac_Init was previously called. If the check fails, the function shall raise the development error V2XFAC_E_UNINIT.⌋

Note: The function requires previous preparation of the pseudonym via an API call to V2xFac_V2xM_PreparePseudonymChange.

8.3.5 V2xFac_V2xM_AbortPseudonymChange

[SWS_V2xFac_00144] Definition of API function V2xFac_V2xM_AbortPseudonymChange ⌈

Service Name	V2xFac_V2xM_AbortPseudonymChange	
Syntax	`Std_ReturnType V2xFac_V2xM_AbortPseudonymChange ( void )`	
Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: operation successful `E_NOT_OK`: operation failed
Description	This function is called by the V2xM when not all modules are OK with the pseudonym change and the change is to be rolled back.	
Available via	V2xFac_V2xM.h	

⌋

[SWS_V2xFac_00145] ⌈The function V2xFac_V2xM_AbortPseudonymChange shall roll back the prepared pseudonym change.⌋

[SWS_V2xFac_00146] ⌈If development error detection is enabled: the function shall check that the service V2xFac_Init was previously called. If the check fails, the function shall raise the development error V2XFAC_E_UNINIT.⌋

Note: The function requires previous preparation of the pseudonym via an API call to V2xFac_V2xM_PreparePseudonymChange.

8.3.6 V2xFac_V2xM_SetTGenCamDcc

[SWS_V2xFac_00148] Definition of API function V2xFac_V2xM_SetTGenCamDcc ⌈

Service Name	V2xFac_V2xM_SetTGenCamDcc
Syntax	`void V2xFac_V2xM_SetTGenCamDcc ( uint16 TGenCamDcc )`
Service ID [hex]	0x06
Sync/Async	Synchronous

▽

△

Reentrancy	Non Reentrant	
Parameters (in)	TGenCamDcc	The TGenCamDcc in [ms], provided by V2xM
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	By this API primitive the V2xFac module gets an indication of the current TGenCamDcc value.	
Available via	V2xFac_V2xM.h	

⌋

[SWS_V2xFac_00149] ⌈The function V2xFac_V2xM_SetTGenCamDcc shall set the TGenCamDcc for subsequent API calls.⌋

[SWS_V2xFac_00150] ⌈If development error detection is enabled: the function shall check that the service V2xFac_Init was previously called. If the check fails, the function shall raise the development error V2XFAC_E_UNINIT.⌋

8.3.7 V2xFac_V2xM_SetCaBsOperation

[SWS_V2xFac_00152] Definition of API function V2xFac_V2xM_SetCaBsOperation ⌈

Service Name	V2xFac_V2xM_SetCaBsOperation	
Syntax	`void V2xFac_V2xM_SetCaBsOperation ( boolean OperationState )`	
Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OperationState	FALSE: CaBs disabled TRUE: CaBs enbaled
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	By this API primitive the V2xFac module gets an indication of the current operation state of the CA Basic Service.	
Available via	V2xFac_V2xM.h	

⌋

[SWS_V2xFac_00153] ⌈The function V2xFac_V2xM_SetCaBsOperation shall enable or disable the CA Basic Service.⌋

[SWS_V2xFac_00154] ⌈If development error detection is enabled: the function shall check that the service V2xFac_Init was previously called. If the check fails, the function shall raise the development error V2XFAC_E_UNINIT.⌋

8.4 Callback notifications

This is a list of functions provided for other modules.

8.4.1 V2xFac_RxIndication

[SWS_V2xFac_00088] Definition of callback function V2xFac_RxIndication ⌈

Service Name	V2xFac_RxIndication	

Syntax	`void V2xFac_RxIndication (`
	`  const V2xFac_RxParamsType* ReceiveParams,`
	`  uint16 Length,`
	`  const uint8* DataPtr`
	`)`	

Service ID [hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ReceiveParams	Wraps RxIndication parameters
	Length	Length of the data pointed by DataPtr.
	DataPtr	Payload of the received BTP packet.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This API primitive is called by the V2xBtp module providing the data and the GeoNetworking parameters of a received BTP packet to V2xFac module.	
Available via	V2xFac.h	

⌋

[SWS_V2xFac_00158] ⌈If development error detection is enabled: the function shall check that the service V2xFac_Init was previously called. If the check fails, the function shall raise the development error V2XFAC_E_UNINIT.⌋

[SWS_V2xFac_00159] ⌈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 V2XFAC_E_PARAM_POINTER.⌋

[SWS_V2xFac_00160] ⌈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 V2XFAC_E_PARAM_POINTER.⌋

8.5 Scheduled functions

8.5.1 V2xFac_CaBs_MainFunction

[SWS_V2xFac_00090] Definition of scheduled function V2xFac_CaBs_MainFunction ⌈

Service Name	V2xFac_CaBs_MainFunction
Syntax	`void V2xFac_CaBs_MainFunction ( void )`
Service ID [hex]	0x0a
Description	This is the main processing function of the CA basic service
Available via	SchM_V2xFac.h

⌋

8.5.2 V2xFac_DenBs_MainFunction

[SWS_V2xFac_00091] Definition of scheduled function V2xFac_DenBs_MainFunction ⌈

Service Name	V2xFac_DenBs_MainFunction
Syntax	`void V2xFac_DenBs_MainFunction ( void )`
Service ID [hex]	0x0b
Description	This is the main processing function of the DEN basic service.
Available via	SchM_V2xFac.h

⌋

8.5.3 V2xFac_RxS_MainFunction

[SWS_V2xFac_00305] Definition of scheduled function V2xFac_RxS_MainFunction

Upstream requirements: SRS_V2X_10001

⌈

Service Name	V2xFac_RxS_MainFunction
Syntax	`void V2xFac_RxS_MainFunction ( void )`
Service ID [hex]	0x0c
Description	This is the main processing function of the Facilities message reception service.
Available via	SchM_V2xFac.h

⌋

[SWS_V2xFac_00306]

Upstream requirements: SRS_V2X_00711, SRS_V2X_00741, SRS_V2X_10001, SRS_V2X_10003, SRS_V2X_10004

⌈The function shall process the Facilities message reception service as described in chapter 7.5.⌋

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.

[SWS_V2xFac_00092] Definition of mandatory interfaces required by module V2x Fac ⌈

API Function	***Header File***	***Description***
V2xBtp_Transmit	V2xBtp.h	This API is called by the V2xFac module to request sending a BTP-PDU to the peer BTP entity.
V2xDM_V2xStackRxIndication	V2xDM.h	Function is called by the underlying V2x stack to provide a V2x message to the V2x Data Manager

▽

△

API Function	Header File	Description
V2xM_CalcDistance	V2xM.h	Calculates the distance between two geographical points on earth with the assumption that they are on elevation 0.
V2xM_CalcHeadingInTolerance	V2xM.h	Calculates if difference of heading values are within a tolerance value
V2xM_GetPositionAndTime	V2xM.h	Provides the instantaneous position information.
V2xM_GetRefTimePtr	V2xM.h	Provides a pointer to the time reference of the V2X-Stack.
V2xM_SetTollingZoneInformation	V2xM.h	Set available tolling zone information. This is done from V2xFac that receives this information via CAM messages.

⌋

8.6.2 Optional interfaces

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

[SWS_V2xFac_00093] Definition of optional interfaces requested by module V2x Fac ⌈

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

⌋

8.7 Service Interfaces

8.7.1 Sender-Receiver-Interfaces

8.7.1.1 V2xFacVdp

[SWS_V2xFac_00094] ⌈The V2xFac requires an interface `V2xFacVdp()` as defined below to get data from the VDP application.⌋

[SWS_V2xFac_00095] Definition of SenderReceiverInterface V2xFacVdp ⌈

Name	V2xFacVdp	
Comment	Interface to receive data from VDP application. Set the boolean to TRUE if the BasicVehicleLowFrequencyContainer shall be sent with the next CAM triggered by the call of this service interface.	
IsService	false	
Variation	–	
Data Elements	vdpData	
	Type	V2xFac_BasicVehicleContainerHighFrequencyType
	Variation	–
	vdpSendBasicLFC	
	Type	boolean
	Variation	–

⌋

[SWS_V2xFac_00321] Reception of a New High Frequency Container ⌈When receiving a new High Frequency Container to generate a CAM, the V2X_Facilities shall check the data element `vdpSendBasicLFC` to verify if a Low Frequency container shall be sent as well with the next transmission.⌋

[SWS_V2xFac_00322] Use last received Low Frequency Container on VDP request only ⌈The contents of a CAM message is entirely under the responsibility of VDP. Hence, V2X_Fac shall only include the last received Low Frequency Container if VDP requests it. Keeping the consistency between Low Frequency and High Frequency container is under the responsibility of VDP.⌋

8.7.1.2 V2xFacVdpBasicLFC

[SWS_V2xFac_91609] Definition of SenderReceiverInterface V2xFacVdpBasicLFC ⌈

Name	V2xFacVdpBasicLFC	
Comment	Interface to provide the data structure of the V2xFac_BasicVehicleContainerLowFrequencyType from the VDP application.	
IsService	false	
Variation	–	
Data Elements	vdpBasicLowFrequencyContainer	
	Type	V2xFac_BasicVehicleContainerLowFrequencyType
	Variation	–

⌋

[SWS_V2xFac_00323] Low Frequency Container reception ⌈This interface provide the low frequency data for the CAM message. It needs to be updated initially and afterwards when a parameter is changing within the data structure. It is important to provide this data structure BEFORE the CAM message is triggered with the call to the service interface `V2xFacVdp()`⌋

8.7.2 Client-Server-Interfaces

8.7.2.1 V2xFacDenBs

The V2xFac module provides the Client-Server service Interface V2xFacDenBs to the application layer. The service Interface V2xFacDenBs shall implement the following operations.

- TriggerEvent
- UpdateEvent
- TerminateEvent

[SWS_V2xFac_00098] ⌈The V2X_Facilities shall provide an interface V2xFacDenBs as defined below to provide tha capability of event handling (triggering, updating and terminating DENMs).⌋

[SWS_V2xFac_00099] Definition of ClientServerInterface V2xFacDenBs ⌈

Name	V2xFacDenBs		
Comment	Service of V2xFac module basic service DEN		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	2	E_ACTION_ID_ NONEXISTENT	ActionID provided for Update/Termination does not exist
	3	E_DENM_ UNCONSTRUCTABLE	DENM couldn't be constructed
	4	E_DENM_TIME_OUT	DENM hasn't been sent before timeout of DENM has been reached

Operation	TerminateEvent	
Comment	Requests termination of an existing DENM (see [11] chapter 4 and 5.4.1.4)	
Mapped to API	–	
Variation	–	
Parameters	EventData	
	Type	V2xFac_DenMsgType
	Direction	IN

▽

△

	Comment	Pre-filled DENM message structure, including the ActionID from TriggerEvent
	Variation	–
	RepetitionDuration	
	Type	uint32
	Direction	IN
	Comment	Duration of the DENM repetition in units of milliseconds
	Variation	–
	RepetitionInterval	
	Type	uint16
	Direction	IN
	Comment	Interval of DENM repetition in units of milliseconds
	Variation	–
	DestinationArea	
	Type	V2xFac_GnDestinationAreaType
	Direction	IN
	Comment	Destination area for DENM dissemination as specified in ETSI EN 302 931.
	Variation	–
	TrafficClass	
	Type	V2xFac_TrafficClassIdType
	Direction	IN
	Comment	GN traffic class of the DENM as defined in ETSI EN 302 636-4-1
	Variation	–
	ActionID	
	Type	V2xFac_ActionIdType
	Direction	OUT
	Comment	The DEN basic service returns the actionID or other applicable identifier created by the DEN basic service to the requesting ITS-S application
	Variation	–
Possible Errors	E_OK E_NOT_OK E_ACTION_ID_NONEXISTENT E_DENM_UNCONSTRUCTABLE E_DENM_TIME_OUT	

Operation	TriggerEvent	
Comment	Requests creation of a new DENM (see [11] chapter 4 and 5.4.1.2)	
Mapped to API	–	
Variation	–	
Parameters	EventData	
	Type	V2xFac_DenMsgType
	Direction	IN
	Comment	Pre-filled DENM message structure
	Variation	–
	RepetitionDuration	
	Type	uint32
	Direction	IN
	Comment	Duration of the DENM repetition in units of milliseconds

▽

△

	Variation	–
	RepetitionInterval	
	Type	uint16
	Direction	IN
	Comment	Interval of DENM repetition in units of milliseconds
	Variation	–
	DestinationArea	
	Type	V2xFac_GnDestinationAreaType
	Direction	IN
	Comment	Destination area for DENM dissemination as specified in ETSI EN 302 931.
	Variation	–
	TrafficClass	
	Type	V2xFac_TrafficClassIdType
	Direction	IN
	Comment	GN traffic class of the DENM as defined in ETSI EN 302 636-4-1
	Variation	–
	ActionID	
	Type	V2xFac_ActionIdType
	Direction	OUT
	Comment	The DEN basic service returns the actionID or other applicable identifier created by the DEN basic service to the requesting ITS-S application
	Variation	–
Possible Errors	E_OK E_NOT_OK E_DENM_UNCONSTRUCTABLE E_DENM_TIME_OUT	

Operation	UpdateEvent	
Comment	Requests update of an existing DENM (see [11] chapter 4 and 5.4.1.3)	
Mapped to API	–	
Variation	–	
Parameters	EventData	
	Type	V2xFac_DenMsgType
	Direction	IN
	Comment	Pre-filled DENM message structure, including the ActionID from TriggerEvent
	Variation	–
	RepetitionDuration	
	Type	uint32
	Direction	IN
	Comment	Duration of the DENM repetition in units of milliseconds
	Variation	–
	RepetitionInterval	
	Type	uint16
	Direction	IN
	Comment	Interval of DENM repetition in units of milliseconds
	Variation	–
	DestinationArea	

▽

△

	Type	V2xFac_GnDestinationAreaType
	Direction	IN
	Comment	Destination area for DENM dissemination as specified in ETSI EN 302 931.
	Variation	–
	TrafficClass	
	Type	V2xFac_TrafficClassIdType
	Direction	IN
	Comment	GN traffic class of the DENM as defined in ETSI EN 302 636-4-1
	Variation	–
	ActionID	
	Type	V2xFac_ActionIdType
	Direction	OUT
	Comment	The DEN basic service returns the actionID or other applicable identifier created by the DEN basic service to the requesting ITS-S application
	Variation	–
Possible Errors	E_OK E_NOT_OK E_ACTION_ID_NONEXISTENT E_DENM_UNCONSTRUCTABLE E_DENM_TIME_OUT	

⌋

8.7.3 Implementation Data Types

8.7.3.1 V2xFac specific Implementation DataTypes

[SWS_V2xFac_00162] Definition of ImplementationDataType V2xFac_TrafficClassIdType ⌈

Name	V2xFac_TrafficClassIdType
Kind	Type
Derived from	uint8
Description	Traffic class for sending DENMs
Variation	–
Available via	Rte_V2xFac_Type.h

⌋

[SWS_V2xFac_00163] Definition of ImplementationDataType V2xFac_GnDestinationAreaType ⌈

Name	V2xFac_GnDestinationAreaType	
Kind	Structure	
Elements	latitude	
	Type	sint32
	Comment	Latitude [1/10 microdegree]
	longitude	
	Type	sint32
	Comment	Longitude [1/10 microdegree]
	distanceA	
	Type	uint16
	Comment	Distance a of the geometric shape [meters]
	distanceB	
	Type	uint16
	Comment	Distance b of the geometric shape [meters]
	angle	
	Type	uint16
	Comment	Angle of the geometric shape [degrees from North]
	shape	
	Type	V2xFac_GnAreaShapeType
	Comment	Shape type of the geometric area
Description	Destination area for DENM dissemination as specified in ETSI EN 302 931.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00164] Definition of ImplementationDataType V2xFac_GnAreaShapeType ⌈

Name	V2xFac_GnAreaShapeType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_GNAREASHAPE_CIRCLE	0x00	Circle
	V2XFAC_GNAREASHAPE_RECT	0x01	Rectangle
	V2XFAC_GNAREASHAPE_ELLIPSE	0x02	Ellipsis
Description	Enumeration of a GeoNetworking Area Shape		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

8.7.3.2 Common Implementation DataTypes

[SWS_V2xFac_00224] Definition of ImplementationDataType V2xFac_DeltaReferencePositionType ⌈

Name	V2xFac_DeltaReferencePositionType	
Kind	Structure	
Elements	deltaLatitude	
	Type	sint32
	Comment	Defines offset latitude with regards to a referred latitude value.
	deltaLongitude	
	Type	sint32
	Comment	Defines an offset longitude with regards to a referred longitude value.
	deltaAltitude	
	Type	sint16
	Comment	Defines an offset altitude with regards to a referred altitude value.
Description	DF_DeltaReferencePosition as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00037] Definition of ImplementationDataType V2xFac_AltitudeType ⌈

Name	V2xFac_AltitudeType	
Kind	Structure	
Elements	altitudeValue	
	Type	sint32
	Comment	Altitude in a WGS84 co-ordinate system
	altitudeConfidence	
	Type	V2xFac_AltitudeConfidenceType
	Comment	Absolute accuracy of a reported altitude value
Description	DF_Altitude as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00165] Definition of ImplementationDataType V2xFac_AltitudeConfidenceType ⌈

Name	V2xFac_AltitudeConfidenceType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_ALTITUDECONFIDENCE_ALT_000_01	0x00	the altitude accuracy is equal to or less than 0.01 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_02	0x01	the altitude accuracy is equal to or less than 0.02 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_05	0x02	the altitude accuracy is equal to or less than 0.05 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_10	0x03	the altitude accuracy is equal to or less than 0.1 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_20	0x04	the altitude accuracy is equal to or less than 0.2 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_50	0x05	the altitude accuracy is equal to or less than 0.5 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_001_00	0x06	the altitude accuracy is equal to or less than 1 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_002_00	0x07	the altitude accuracy is equal to or less than 2 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_005_00	0x08	the altitude accuracy is equal to or less than 5 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_010_00	0x09	the altitude accuracy is equal to or less than 10 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_020_00	0x0a	the altitude accuracy is equal to or less than 20 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_050_00	0x0b	the altitude accuracy is equal to or less than 50 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_100_00	0x0c	the altitude accuracy is equal to or less than 100 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_200_00	0x0d	the altitude accuracy is equal to or less than 200 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_OUTOFRANGE	0x0e	the altitude accuracy is out of range, i.e. greater than 200 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_UNAVAILABLE	0x0f	the altitude accuracy information is unavailable
Description	Enumeration of DE_AltitudeConfidence as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00038] Definition of ImplementationDataType V2xFac_PosConfidenceEllipseType ⌈

Name	V2xFac_PosConfidenceEllipseType	
Kind	Structure	
Elements	semiMajorConfidence	
	Type	uint16
	Comment	Half of length of the major axis
	semiMinorConfidence	
	Type	uint16
	Comment	Half of length of the minor axis
	semiMajorOrientation	
	Type	uint16
	Comment	Orientation direction of the ellipse major axis
Description	DF_PosConfidenceEllipse as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00039] Definition of ImplementationDataType V2xFac_HeadingType ⌈

Name	V2xFac_HeadingType	
Kind	Structure	
Elements	headingValue	
	Type	uint16
	Comment	Orientation of a heading with regards to the WGS84 north
	headingConfidence	
	Type	uint8
	Comment	Absolute accuracy of a reported heading value
Description	DF_Heading as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00040] Definition of ImplementationDataType V2xFac_SpeedType ⌈

Name	V2xFac_SpeedType	
Kind	Structure	
Elements	speedValue	
	Type	uint16

▽

△

	Comment	Speed value
	speedConfidence	
	Type	uint8
	Comment	The absolute accuracy of a speed value
Description	DF_Speed as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00047] Definition of ImplementationDataType V2xFac_ReferencePositionType ⌈

Name	V2xFac_ReferencePositionType	
Kind	Structure	
Elements	latitude	
	Type	sint32
	Comment	Latitude of the geographical point
	longitude	
	Type	sint32
	Comment	Longitude of the geographical point
	posConfidenceEllipse	
	Type	V2xFac_PosConfidenceEllipseType
	Comment	Accuracy of the geographical position
	altitude	
	Type	V2xFac_AltitudeType
	Comment	Altitude and altitude accuracy of the geographical point
Description	DF_ReferencePosition as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00225] Definition of ImplementationDataType V2xFac_ActionId Type ⌈

Name	V2xFac_ActionIdType	
Kind	Structure	
Elements	originatingStationID	
	Type	uint32
	Comment	Identifier for an ITS-S
	sequenceNumber	
	Type	uint16

▽

△

	Comment	sequenceNumber
Description	DF_ActionID as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00059] Definition of ImplementationDataType V2xFac_PathHistoryType ⌈

Name	V2xFac_PathHistoryType	
Kind	Structure	
Elements	count	
	Type	uint8
	Comment	Number of valid elements within array.
	values	
	Type	Array of V2xFac_PathPointType
	Size	23
	Comment	–
Description	DF_PathHistory as defined in ETSI TS 102 894-2. Size of the Array shall be 23 as defined in ETSI EN 302 637-2.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00074] Definition of ImplementationDataType V2xFac_CauseCodeType ⌈

Name	V2xFac_CauseCodeType	
Kind	Structure	
Elements	causeCode	
	Type	uint8
	Comment	Encoded value of a traffic event type
	subCauseCode	
	Type	uint8
	Comment	Type of sub cause of a detected event
Description	DF_CauseCode as defined inETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

8.7.3.3 CAM specific Implementation DataTypes

[SWS_V2xFac_00177] Definition of ImplementationDataType V2xFac_AccelerationControlType ⌈

Name	V2xFac_AccelerationControlType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	brakePedalEngaged	0x40	Bit 6: Driver is stepping on the brake pedal
	bit	gasPedalEngaged	0x20	Bit 5: Driver is stepping on the gas pedal
	bit	emergencyBrakeEngaged	0x10	Bit 4: Emergency brake system is engaged
	bit	collisionWarningEngaged	0x08	Bit 3: Collision warning system is engaged
	bit	accEngaged	0x04	Bit 2: ACC is engaged
	bit	cruiseControlEngaged	0x02	Bit 1: Cruise control is engaged
	bit	speedLimiterEngaged	0x01	Bit 0 (LSB): Speed limiter is engaged
Description	BitString DE_AccelerationControl as defined in ETSI TS 102 894-2.			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00174] Definition of ImplementationDataType V2xFac_BasicVehicleContainerHighFrequencyPresenceType ⌈

Name	V2xFac_BasicVehicleContainerHighFrequencyPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	accelerationControl	0x40	Bit 6: Optional child present
	bit	lanePosition	0x20	Bit 5: Optional child present
	bit	steeringWheelAngle	0x10	Bit 4: Optional child present
	bit	lateralAcceleration	0x08	Bit 3: Optional child present
	bit	verticalAcceleration	0x04	Bit 2: Optional child present
	bit	performanceClass	0x02	Bit 1: Optional child present
	bit	cenDsrcTollingZone	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_BasicVehicleContainerHighFrequencyType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00173] Definition of ImplementationDataType V2xFac_BasicVehicleContainerHighFrequencyType ⌈

Name	V2xFac_BasicVehicleContainerHighFrequencyType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_BasicVehicleContainerHighFrequencyPresenceType
	Comment	Mark optional childs present or not
	heading	
	Type	V2xFac_HeadingType
	Comment	Heading and heading accuracy of the vehicle movement
	speed	
	Type	V2xFac_SpeedType
	Comment	Driving speed and speed accuracy of the originating ITS-S
	driveDirection	
	Type	V2xFac_DriveDirectionType
	Comment	Vehicle drive direction
	vehicleLength	
	Type	V2xFac_VehicleLengthType
	Comment	Vehicle length and accuracy of the vehicle that originates the CAM
	vehicleWidth	
	Type	uint8
	Comment	Width of a vehicle, including side mirrors
	longitudinalAcceleration	
	Type	V2xFac_LongitudinalAccelerationType
	Comment	Vehicle longitudinal acceleration and accuracy
	curvature	
	Type	V2xFac_CurvatureType
	Comment	Actual trajectory curvature and accuracy
	curvatureCalculationMode	
	Type	V2xFac_CurvatureCalculationModeType
	Comment	Flag indicating whether vehicle yaw-rate is used
	yawRate	
	Type	V2xFac_YawRateType
	Comment	YawRate and accuracy
	accelerationControl	
	Type	V2xFac_AccelerationControlType
	Comment	Current status of the vehicle mechanisms controlling the longitudinal movement
	lanePosition	
	Type	sint8
	Comment	Lane position of the vehicle
	steeringWheelAngle	
	Type	V2xFac_SteeringWheelAngleType
	Comment	Steering wheel angle and accuracy
	lateralAcceleration	

▽

△

	Type	V2xFac_LateralAccelerationType
	Comment	Vehicle lateral acceleration and accuracy
	verticalAcceleration	
	Type	V2xFac_VerticalAccelerationType
	Comment	Vertical Acceleration of the originating ITS-S
	performanceClass	
	Type	uint8
	Comment	Characterizes the maximum age of the CAM data elements
Description	BasicVehicleContainerHighFrequency as defined in ETSI EN 302 637-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00187] Definition of ImplementationDataType V2xFac_BasicVehicleContainerLowFrequencyType ⌈

Name	V2xFac_BasicVehicleContainerLowFrequencyType	
Kind	Structure	
Elements	vehicleRole	
	Type	V2xFac_VehicleRoleType
	Comment	Vehicle role
	exteriorLights	
	Type	V2xFac_ExteriorLightsType
	Comment	Exterior Lights
	pathHistory	
	Type	V2xFac_PathHistoryType
	Comment	Path History
Description	BasicVehicleLowFrequencyContainer as defined in ETSI EN 302 637-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00184] Definition of ImplementationDataType V2xFac_CurvartureConfidenceType ⌈

Name	V2xFac_CurvartureConfidenceType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_CURVATURE-CONFIDENCE_ONE_PER_METER_0_00002	0x00	The accuracy is less than or equal to 0,00002 m-1

▽

△

	V2XFAC_CURVATURE-CONFIDENCE_ONE_PER_METER_0_0001	0x01	The accuracy is less than or equal to 0,0001 m-1
	V2XFAC_CURVATURE-CONFIDENCE_ONE_PER_METER_0_0005	0x02	The accuracy is less than or equal to 0,0005 m-1
	V2XFAC_CURVATURE-CONFIDENCE_ONE_PER_METER_0_002	0x03	The accuracy is less than or equal to 0,002 m-1
	V2XFAC_CURVATURE-CONFIDENCE_ONE_PER_METER_0_01	0x04	The accuracy is less than or equal to 0,01 m-1
	V2XFAC_CURVATURE-CONFIDENCE_ONE_PER_METER_0_1	0x05	The accuracy is less than or equal to 0,1 m-1
	V2XFAC_CURVATURE-CONFIDENCE_OUT_OF_RANGE	0x06	The accuracy is out of range, i.e. greater than 0,1 m-1
	V2XFAC_CURVATURE-CONFIDENCE_UNAVAILABLE	0x07	The information is not available
Description	Enumeration of DE_CurvatureConfidence as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00176] Definition of ImplementationDataType V2xFac_CurvatureCalculationModeType ⌈

Name	V2xFac_CurvatureCalculationModeType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_CURVATURECALCMODE_YAWRATE_USED	0x00	Calc mode Yawrate used
	V2XFAC_CURVATURECALCMODE_YAWRATE_NOT_USED	0x01	Calc mode Yawrate not used
	V2XFAC_CURVATURECALCMODE_UNAVAILABLE	0x02	Calc mode unavailable
Description	Enumeration of DE_CurvatureCalculationMode as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00052] Definition of ImplementationDataType V2xFac_CurvatureType ⌈

Name	V2xFac_CurvatureType	
Kind	Structure	
Elements	curvatureValue	
	Type	sint16
	Comment	Describes the inverse of a detected vehicle turning curve radius
	curvatureConfidence	
	Type	V2xFac_CurvartureConfidenceType
	Comment	Describes the absolute accuracy range of a reported curvature value
Description	DF_Curvature as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00175] Definition of ImplementationDataType V2xFac_DriveDirectionType ⌈

Name	V2xFac_DriveDirectionType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_DRIVINGDIRECTION_FORWARD	0x00	Driving direction forward
	V2XFAC_DRIVINGDIRECTION_BACKWARD	0x01	Driving direction backward
	V2XFAC_DRIVINGDIRECTION_UNAVAILABLE	0x02	Driving direction unavailable
Description	Enumeration of DE_DrivingDirection as defined in ETSI EN 302 637-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00189] Definition of ImplementationDataType V2xFac_ExteriorLightsType ⌈

Name	V2xFac_ExteriorLightsType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	lowBeamHeadlightsOn	0x80	Bit 7: low beam headlights on
	bit	highBeamHeadlightsOn	0x40	Bit 6: high beam headlights on

▽

△

	bit	leftTurnSignalOn	0x20	Bit 5: left turn signal on
	bit	rightTurnSignalOn	0x10	Bit 4: right turn signal on
	bit	daytimeRunningLightsOn	0x08	Bit 3: daytime running lights on
	bit	reverseLightOn	0x04	Bit 2: reverse light on
	bit	fogLightOn	0x02	Bit 1: fog light on
	bit	parkingLightsOn	0x01	Bit 0: parking lights on
Description	BitString DE_ExteriorLights as defined in ETSI TS 102 894-2.			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00055] Definition of ImplementationDataType V2xFac_LateralAccelerationType ⌈

Name	V2xFac_LateralAccelerationType	
Kind	Structure	
Elements	lateralAccelerationValue	
	Type	sint16
	Comment	Vehicle acceleration at lateral direction
	lateralAccelerationConfidence	
	Type	uint8
	Comment	The absolute accuracy of a reported vehicle acceleration
Description	DF_LateralAcceleration as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00051] Definition of ImplementationDataType V2xFac_LongitudinalAccelerationType ⌈

Name	V2xFac_LongitudinalAccelerationType	
Kind	Structure	
Elements	longitudinalAccelerationValue	
	Type	sint16
	Comment	Vehicle acceleration at longitudinal direction
	longitudinalAccelerationConfidence	
	Type	uint8
	Comment	The absolute accuracy of a reported vehicle acceleration
Description	DF_LongitudinalAcceleration as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00190] Definition of ImplementationDataType V2xFac_PathPoint PresenceType ⌈

Name	V2xFac_PathPointPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	pathDeltaTime	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_PathPointType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00060] Definition of ImplementationDataType V2xFac_PathPoint Type ⌈

Name	V2xFac_PathPointType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_PathPointPresenceType
	Comment	Mark optional children present or not
	pathPosition	
	Type	V2xFac_DeltaReferencePositionType
	Comment	Defines a geographical point position as offset position to a reference geographical point.
	pathDeltaTime	
	Type	uint16
	Comment	Presents the time difference when two consecutive PathPoint values are measured.
Description	DF_PathPoint as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00054] Definition of ImplementationDataType V2xFac_Steering WheelAngleType ⌈

Name	V2xFac_SteeringWheelAngleType	
Kind	Structure	
Elements	steeringWheelAngleValue	
	Type	uint16
	Comment	Steering wheel angle of the vehicle at certain point in time.
	steeringWheelAngleConfidence	

▽

△

	Type	uint8
	Comment	Absolute accuracy for a reported steering wheel angle value.
Description	DF_SteeringWheelAngle as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00239] Definition of ImplementationDataType V2xFac_VehicleLengthConfidenceIndicationType ⌈

Name	V2xFac_VehicleLengthConfidenceIndicationType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_VEHICLELENGTHCONFIDENCEINDICATION_NOTRAILERPRESENT	0x00	no trailer present
	V2XFAC_VEHICLELENGTHCONFIDENCEINDICATION_TRAILERPRESENTWITHKNOWNLENGTH	0x01	trailer present with known length
	V2XFAC_VEHICLELENGTHCONFIDENCEINDICATION_TRAILERPRESENTWITHUNKNOWNLENGTH	0x02	trailer present with unknown length
	V2XFAC_VEHICLELENGTHCONFIDENCEINDICATION_TRAILERPRESENCEISUNKNOWN	0x03	trailer presence is unknown
	V2XFAC_VEHICLELENGTHCONFIDENCEINDICATION_UNAVAILABLE	0x04	information is not known
Description	Enumeration of DE_VehicleLengthConfidenceIndication as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00050] Definition of ImplementationDataType V2xFac_VehicleLengthType ⌈

Name	V2xFac_VehicleLengthType	
Kind	Structure	
Elements	vehicleLengthValue	
	Type	uint16

▽

△

	Comment	Length of a vehicle
	vehicleLengthConfidenceIndication	
	Type	V2xFac_VehicleLengthConfidenceIndicationType
	Comment	Indication of whether trailer is detected to be present and whether the length of the trailer is known.
Description	DF_VehicleLength as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00188] Definition of ImplementationDataType V2xFac_VehicleRoleType ⌈

Name	V2xFac_VehicleRoleType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_VEHICLEROLE_DEFAULT	0x00	default vehicle role as indicated by the vehicle type
	V2XFAC_VEHICLEROLE_PUBLIC_TRANSPORT	0x01	vehicle is used to operate public transport service
	V2XFAC_VEHICLEROLE_SPECIAL_TRANSPORT	0x02	vehicle is used for special transport purpose, e.g. oversized trucks
	V2XFAC_VEHICLEROLE_DANGEROUS_GOODS	0x03	vehicle is used for dangerous goods transportation
	V2XFAC_VEHICLEROLE_ROAD_WORK	0x04	vehicle is used to realize roadwork or road maintenance mission
	V2XFAC_VEHICLEROLE_RESCUE	0x05	vehicle is used for rescue purpose in case of an accident, e.g. as a towing service
	V2XFAC_VEHICLEROLE_EMERGENCY	0x06	vehicle is used for emergency mission, e.g. ambulance, fire brigade
	V2XFAC_VEHICLEROLE_SAFETY_CAR	0x07	vehicle is used for public safety, e.g. patrol
	V2XFAC_VEHICLEROLE_AGRICULTURAL	0x08	vehicle is used for agriculture, e.g. farm tractor
	V2XFAC_VEHICLEROLE_COMMERCIAL	0x09	vehicle is used for transportation of commercial goods
	V2XFAC_VEHICLEROLE_MILITARY	0x0a	vehicle is used for military purpose
	V2XFAC_VEHICLEROLE_ROAD_OPERATOR	0x0b	vehicle is used in road operator missions
	V2XFAC_VEHICLEROLE_TAXI	0x0c	vehicle is used to provide an authorized taxi service
	V2XFAC_VEHICLEROLE_RESERVED_1	0x0d	reserved for future usage
	V2XFAC_VEHICLEROLE_RESERVED_2	0x0e	reserved for future usage

▽

△

	V2XFAC_VEHICLEROLE_RESERVED_3	0x0f	reserved for future usage
Description	Enumeration of DE_VehicleRole as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00056] Definition of ImplementationDataType V2xFac_VerticalAccelerationType ⌈

Name	V2xFac_VerticalAccelerationType	
Kind	Structure	
Elements	verticalAccelerationValue	
	Type	sint16
	Comment	Vehicle acceleration at vertival direction
	verticalAccelerationConfidence	
	Type	uint8
	Comment	The absolute accuracy of a reported vehicle acceleration
Description	DF_VerticalAcceleration as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00245] Definition of ImplementationDataType V2xFac_YawRateConfidenceType ⌈

Name	V2xFac_YawRateConfidenceType		
Kind	Type		
Derived from	uint8		
Range	YAWRATECONFIDENCE_DEGSEC_000_01	0x00	0 if the accuracy is equal to or less than 0,01 degree/second
	YAWRATECONFIDENCE_DEGSEC_000_05	0x01	1 if the accuracy is equal to or less than 0,05 degrees/second
	YAWRATECONFIDENCE_DEGSEC_000_10	0x02	2 if the accuracy is equal to or less than 0,1 degree/second
	YAWRATECONFIDENCE_DEGSEC_001_00	0x03	3 if the accuracy is equal to or less than 1 degree/second
	YAWRATECONFIDENCE_DEGSEC_005_00	0x04	4 if the accuracy is equal to or less than 5 degrees/second
	YAWRATECONFIDENCE_DEGSEC_010_00	0x05	5 if the accuracy is equal to or less than 10 degrees/second
	YAWRATECONFIDENCE_DEGSEC_100_00	0x06	6 if the accuracy is equal to or less than 100 degrees/second

▽

△

	YAWRATECONFIDENCE_OUTOFRANGE	0x07	7 if the accuracy is out of range, i.e. greater than 100 degrees/ second
	YAWRATECONFIDENCE_UNAVAILABLE	0x08	8 if the accuracy information is unavailable
Description	Enumeration of DE_YawRateConfidence as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00053] Definition of ImplementationDataType V2xFac_YawRate Type ⌈

Name	V2xFac_YawRateType	
Kind	Structure	
Elements	yawRateValue	
	Type	sint16
	Comment	Vehicle rotation around z-axis
	yawRateConfidence	
	Type	V2xFac_YawRateConfidenceType
	Comment	Absolute accuracy range for reported yaw rate value
Description	DF_YawRate as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

8.7.3.4 DENM specific Implementation DataTypes

[SWS_V2xFac_00304] ⌈Dangerous Goods and Road works containers are not supported by V2xFac module in transmission.⌋

Note: The reception is supported but handled by the V2x Data Manager Module. The API must be reworked for the next release to better reflect this.

[SWS_V2xFac_00070] Definition of ImplementationDataType V2xFac_DenMsgType ⌈

Name	V2xFac_DenMsgType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_DenMsgPresenceType
	Comment	Mark optional childs present or not
	management	
	Type	V2xFac_ManagementContainerType
	Comment	management container
	situation	
	Type	V2xFac_SituationContainerType
	Comment	situation container
	location	
	Type	V2xFac_LocationContainerType
	Comment	location container
	alacarte	
	Type	V2xFac_AlacarteContainerType
	Comment	alacarte container
Description	DecentralizedEnvironmentalNotificationMessage as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00199] Definition of ImplementationDataType V2xFac_DenMsgPresenceType ⌈

Name	V2xFac_DenMsgPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	situation	0x04	Bit 2: Optional child present
	bit	location	0x02	Bit 1: Optional child present
	bit	alacarte	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_DenMsgType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00071] Definition of ImplementationDataType V2xFac_ManagementContainerType ⌈

Name	V2xFac_ManagementContainerType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_ManagementContainerPresenceType
	Comment	Mark optional childs present or not
	actionId	
	Type	V2xFac_ActionIdType
	Comment	Action identifier
	detectionTime	
	Type	uint64
	Comment	Time at which the event is detected
	referenceTime	
	Type	uint64
	Comment	Refers to the time at which a new DENM, an update DENM or a cancellation DENM is generated
	termination	
	Type	V2xFac_TerminationType
	Comment	Indicates if the type of generated DENM is a cancellation DENM or a negation DENM.
	eventPosition	
	Type	V2xFac_ReferencePositionType
	Comment	Geographical position of the detected event
	relevanceDistance	
	Type	V2xFac_RelevanceDistanceType
	Comment	The distance in which event information is relevant for the receiving ITS-S
	relevanceTrafficDirection	
	Type	V2xFac_RelevanceTrafficDirectionType
	Comment	Traffic direction that is relevant to information indicated in a message
	validityDuration	
	Type	uint32
	Comment	estimation of how long the event may persist
	transmissionInterval	
	Type	uint16
	Comment	Time interval between two consecutive message transmissions
	stationType	
	Type	uint8
	Comment	Station type information of the originating ITS-S
Description	ManagementContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00240] Definition of ImplementationDataType V2xFac_TerminationType ⌈

Name	V2xFac_TerminationType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_TERMINATION_ISCANCELLATION	0x00	Cancellation
	V2XFAC_TERMINATION_ISNEGATION	0x01	–
Description	Enumeration of Termination as defined inETSI EN 302 637-3.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00200] Definition of ImplementationDataType V2xFac_RelevanceDistanceType ⌈

Name	V2xFac_RelevanceDistanceType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_50_M	0x00	less than 50 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_100_M	0x01	less than 100 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_200_M	0x02	less than 200 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_500_M	0x03	less than 500 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_1000_M	0x04	less than 1000 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_5_KM	0x05	less than 5 km
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_10_KM	0x06	less than 10 km
	V2XFAC_RELEVANCEDISTANCE_OVER_10_KM	0x07	over 10 km
Description	Enumeration of DE_RelevanceDistance as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00201] Definition of ImplementationDataType V2xFac_RelevanceTrafficDirectionType ⌈

Name	V2xFac_RelevanceTrafficDirectionType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_RELEVANCE-TRAFFICDIRECTION_ALL_TRAFFIC_DIRECTIONS	0x00	all traffic directions
	V2XFAC_RELEVANCE-TRAFFICDIRECTION_UPSTREAM_TRAFFIC	0x01	upstream traffic
	V2XFAC_RELEVANCE-TRAFFICDIRECTION_DOWNSTREAM_TRAFFIC	0x02	downstream traffic
	V2XFAC_RELEVANCE-TRAFFICDIRECTION_OPPOSITE_TRAFFIC	0x03	opposite traffic
Description	Enumeration of DE_RelevanceTrafficDirection as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00202] Definition of ImplementationDataType V2xFac_ManagementContainerPresenceType ⌈

Name	V2xFac_ManagementContainerPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	termination	0x08	Bit 3: Optional child present
	bit	relevanceDistance	0x04	Bit 2: Optional child present
	bit	relevanceTrafficDirection	0x02	Bit 1: Optional child present
	bit	transmissionInterval	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_ManagementContainerType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00073] Definition of ImplementationDataType V2xFac_SituationContainerType ⌈

Name	V2xFac_SituationContainerType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_SituationContainerPresenceType

▽

△

	Comment	Mark optional childs present or not
	informationQuality	
	Type	uint8
	Comment	Quality level of the information provided by the ITS-S application
	eventType	
	Type	V2xFac_CauseCodeType
	Comment	Encoded value of a traffic event type
	linkedCause	
	Type	V2xFac_CauseCodeType
	Comment	Encoded value of a traffic event type
	eventHistory	
	Type	V2xFac_EventHistoryType
	Comment	EventHistory
Description	SituationContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00203] Definition of ImplementationDataType V2xFac_SituationContainerPresenceType ⌈

Name	V2xFac_SituationContainerPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	linkedCause	0x02	Bit 1: Optional child present
	bit	eventHistory	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_SituationContainerType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00075] Definition of ImplementationDataType V2xFac_EventHistoryType ⌈

Name	V2xFac_EventHistoryType	
Kind	Structure	
Elements	count	
	Type	uint8
	Comment	Number of valid elements within array.
	values	

▽

△

	Type	Array of V2xFac_EventPointType
	Size	23
	Comment	–
Description	DF_EventHistory as defined in ETSI TS 102 894-2.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00076] Definition of ImplementationDataType V2xFac_EventPoint Type ⌈

Name	V2xFac_EventPointType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_EventPointPresenceType
	Comment	Mark optional childs present or not
	eventPosition	
	Type	V2xFac_DeltaReferencePositionType
	Comment	Offset position of a detected event point.
	eventDeltaTime	
	Type	uint16
	Comment	Time travelled by the detecting ITS-S since the previous detected event point.
	informationQuality	
	Type	uint8
	Comment	Information quality of the detection for this event point.
Description	DF_EventPoint as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00204] Definition of ImplementationDataType V2xFac_EventPoint PresenceType ⌈

Name	V2xFac_EventPointPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	eventDeltaTime	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_EventPointType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00077] Definition of ImplementationDataType V2xFac_LocationContainerType ⌈

Name	V2xFac_LocationContainerType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_LocationContainerPresenceType
	Comment	Mark optional childs present or not
	eventSpeed	
	Type	V2xFac_SpeedType
	Comment	Moving speed of a detected event
	eventPositionHeading	
	Type	V2xFac_HeadingType
	Comment	The heading direction of the event
	traces	
	Type	V2xFac_TracesType
	Comment	One or more paths
	roadType	
	Type	V2xFac_RoadTypeType
	Comment	Type of a road segment.
Description	LocationContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00241] Definition of ImplementationDataType V2xFac_RoadTypeType ⌈

Name	V2xFac_RoadTypeType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_ROADTYPE_URBAN_NOSTRUCTURALSEPARATIONTOOPPOSITELANES	0x00	Urban road without structural separation to opposite lanes.
	V2XFAC_ROADTYPE_URBAN_WITHSTRUCTURALSEPARATIONTOOPPOSITELANES	0x01	Urban road with structural separation to opposite lanes.
	V2XFAC_ROADTYPE_NONURBAN_NOSTRUCTURALSEPARATIONTOOPPOSITELANES	0x02	Non-urban road without structural separation to opposite lanes.
	V2XFAC_ROADTYPE_ONURBAN_WITHSTRUCTURALSEPARATIONTOOPPOSITELANES	0x03	Non-urban road with structural separation to opposite lanes.
Description	Enumeration of DE_RoadType as defined in ETSI TS 102 894-2.		

▽

△

Variation	–
Available via	Rte_V2xFac_Type.h

⌋

[SWS_V2xFac_00205] Definition of ImplementationDataType V2xFac_TracesType ⌈

Name	V2xFac_TracesType	
Kind	Structure	
Elements	count	
	Type	uint8
	Comment	Number of valid elements within array.
	values	
	Type	Array of V2xFac_PathHistoryType
	Size	7
	Comment	–
Description	DF_Traces as defined in ETSI TS 102 894-2. Size of the Array shall be 7.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00206] Definition of ImplementationDataType V2xFac_LocationContainerPresenceType ⌈

Name	V2xFac_LocationContainerPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	eventSpeed	0x04	Bit 2: Optional child present
	bit	eventPositionHeading	0x02	Bit 1: Optional child present
	bit	roadType	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_LocationContainerType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00078] Definition of ImplementationDataType V2xFac_AlacarteContainerType ⌈

Name	V2xFac_AlacarteContainerType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_AlacarteContainerPresenceType
	Comment	Mark optional childs present or not
	lanePosition	
	Type	sint8
	Comment	The lane position of the event position
	impactReduction	
	Type	V2xFac_ImpactReductionContainerType
	Comment	–
	externalTemperature	
	Type	sint8
	Comment	Indicates the ambient temperature at the event position
	positioningSolution	
	Type	V2xFac_PositioningSolutionTypeType
	Comment	Indicates the positioning technology being used to estimate a geographical position
	stationaryVehicle	
	Type	V2xFac_StationaryVehicleContainerType
	Comment	–
Description	AlacarteContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00207] Definition of ImplementationDataType V2xFac_PositioningSolutionTypeType ⌈

Name	V2xFac_PositioningSolutionTypeType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_POSITIONING-SOLUTIONTYPE_NO_POSITIONING_SOLUTION	0x00	No GNSS
	V2XFAC_POSITIONING-SOLUTIONTYPE_SGNSS	0x01	Global Navigation Satellite System
	V2XFAC_POSITIONING-SOLUTIONTYPE_DGNSS	0x02	Differential GNSS
	V2XFAC_POSITIONING-SOLUTIONTYPE_SGNSSPLUSDR	0x03	GNSS and dead reckoning

▽

△

	V2XFAC_POSITIONING-SOLUTIONTYPE_ DGNSSPLUSDR	0x04	Differential GNSS and dead reckoning
	V2XFAC_POSITIONING-SOLUTIONTYPE_ DR	0x05	dead reckoning
Description	Enumeration of DE_PositioningSolutionType as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00208] Definition of ImplementationDataType V2xFac_AlacarteContainerPresenceType ⌈

Name	V2xFac_AlacarteContainerPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	lanePosition	0x20	Bit 5: Optional child present
	bit	impactReduction	0x10	Bit 4: Optional child present
	bit	externalTemperature	0x08	Bit 3: Optional child present
	bit	roadWorks	0x04	Bit 2: Optional child present
	bit	positioningSolution	0x02	Bit 1: Optional child present
	bit	stationaryVehicle	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_AlacarteContainerType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00079] Definition of ImplementationDataType V2xFac_ImpactReductionContainerType ⌈

Name	V2xFac_ImpactReductionContainerType	
Kind	Structure	
Elements	heightLonCarrLeft	
	Type	uint8
	Comment	Height of left longitudinal carrier of the vehicle from base to top
	heightLonCarrRight	
	Type	uint8
	Comment	Height of right longitudinal carrier of the vehicle from base to top
	posLonCarrLeft	
	Type	uint8

▽

△

	Comment	Distance from the centre of vehicle front bumper to the front of the left longitudinal carrier of vehicle
	posLonCarrRight	
	Type	uint8
	Comment	Distance from the centre of vehicle front bumper to the front of the right longitudinal carrier of vehicle
	positionOfPillars	
	Type	V2xFac_PositionOfPillarsType
	Comment	Indicates the perpendicular inter-distance of neighbouring pillar
	posCentMass	
	Type	uint8
	Comment	Indicates the perpendicular distance from the centre of mass of an empty load vehicle
	wheelBaseVehicle	
	Type	uint8
	Comment	Perpendicular distance between front and rear axle of the wheel base of vehicle
	turningRadius	
	Type	uint8
	Comment	The smallest circular turn (i.e. U-turn) that the vehicle is capable of making
	posFrontAx	
	Type	uint8
	Comment	Perpendicular distance between the vehicle front line of the bounding box and the front wheel axle in 10 centimetres
	positionOfOccupants	
	Type	V2xFac_PositionOfOccupantsType
	Comment	indicates whether a in vehicle seat is occupied at the moment when the impactReduction is generated
	vehicleMass	
	Type	uint16
	Comment	Mass of an empty loaded vehicle in multiple of 100 kg
	requestResponseIndication	
	Type	V2xFac_RequestResponseIndicationType
	Comment	This DE includes whether an ITS message is transmitted as request from ITS-S or a response transmitted from ITS-S after receiving request from other ITS-Ss
Description	ImpactReductionContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00209] Definition of ImplementationDataType V2xFac_PositionOfPillarsType ⌈

Name	V2xFac_PositionOfPillarsType	
Kind	Structure	
Elements	count	
	Type	uint8
	Comment	Number of valid elements within array.
	values	
	Type	Array of uint8
	Size	3
	Comment	–
Description	DF_PositionOfPillars as defined in ETSI TS 102 894-2. Size of the Array shall be 3.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00210] Definition of ImplementationDataType V2xFac_PositionOfOccupantsType ⌈

Name	V2xFac_PositionOfOccupantsType			
Kind	Bitfield			
Derived from	uint32			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	row1LeftOccupied	0x80000	Bit 19: row 1 left occupied
	bit	row1RightOccupied	0x40000	Bit 18: row 1 right occupied
	bit	row1MidOccupied	0x20000	Bit 17: row 1 mid occupied
	bit	row1NotDetectable	0x10000	Bit 16: row 1 not detectable
	bit	row1NotPresent	0x8000	Bit 15: row 1 not present
	bit	row2LeftOccupied	0x4000	Bit 14: row 2 left occupied
	bit	row2RightOccupied	0x2000	Bit 13: row 2 right occupied
	bit	row2MidOccupied	0x1000	Bit 12: row 2 mid occupied
	bit	row2NotDetectable	0x800	Bit 11: row 2 not detectable
	bit	row2NotPresent	0x400	Bit 10: row 2 not present
	bit	row3LeftOccupied	0x200	Bit 9: row 3 left occupied
	bit	row3RightOccupied	0x100	Bit 8: row 3 right occupied
	bit	row3MidOccupied	0x80	Bit 7: row 3 mid occupied
	bit	row3NotDetectable	0x40	Bit 6: row 3 not detectable
	bit	row3NotPresent	0x20	Bit 5: row 3 not present
	bit	row4LeftOccupied	0x10	Bit 4: row 4 left occupied
	bit	row4RightOccupied	0x08	Bit 3: row 4 right occupied
	bit	row4MidOccupied	0x04	Bit 2: row 4 mid occupied
	bit	row4NotDetectable	0x02	Bit 1: row 4 not detectable
	bit	row4NotPresent	0x01	Bit 0 (LSB): row 4 not present
Description	BitString DE_PositionOfOccupants as defined in ETSI TS 102 894-2.			

▽

△

Variation	–
Available via	Rte_V2xFac_Type.h

⌋

[SWS_V2xFac_00242] Definition of ImplementationDataType V2xFac_RequestResponseIndicationType ⌈

Name	V2xFac_RequestResponseIndicationType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_REQUESTRESPONSEINDICATION_REQUEST	0x00	Request
	V2XFAC_REQUESTRESPONSEINDICATION_RESPONSE	0x01	Response
Description	Enumeration of DE_RequestResponseIndication as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00081] Definition of ImplementationDataType V2xFac_StationaryVehicleContainerType ⌈

Name	V2xFac_StationaryVehicleContainerType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_StationaryVehicleContainerPresenceType
	Comment	Mark optional childs present or not
	stationarySince	
	Type	V2xFac_StationarySinceType
	Comment	Duration in minutes of a vehicle being stationary
	stationaryCause	
	Type	V2xFac_CauseCodeType
	Comment	Additional information to describe causes of the stationary vehicle
	numberOfOccupants	
	Type	uint8
	Comment	Number of occupants in a vehicle
	vehicleIdentification	
	Type	V2xFac_VehicleIdentificationType
	Comment	Provides information related to the identification of a vehicle
	energyStorageType	
	Type	V2xFac_EnergyStorageType
	Comment	Type of energy being used and stored

▽

△

Description	StationaryVehicleContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.
Variation	–
Available via	Rte_V2xFac_Type.h

⌋

[SWS_V2xFac_00216] Definition of ImplementationDataType V2xFac_StationarySinceType ⌈

Name	V2xFac_StationarySinceType		
Kind	Type		
Derived from	uint8		
Range	V2XFAC_STATIONARYSINCE_LESS_THAN_1_MINUTE	0x00	less than 1 minute
	V2XFAC_STATIONARYSINCE_LESS_THAN_2_MINUTES	0x01	less than 2 minutes
	V2XFAC_STATIONARYSINCE_LESS_THAN_15_MINUTES	0x02	less than 15 minutes
	V2XFAC_STATIONARYSINCE_EQUAL_OR_GREATER_15_MINUTES	0x03	equal or greater 15 minutes
Description	Enumeration of DE_StationarySince as defined in ETSI TS 102 894-2.		
Variation	–		
Available via	Rte_V2xFac_Type.h		

⌋

[SWS_V2xFac_00217] Definition of ImplementationDataType V2xFac_EnergyStorageType ⌈

Name	V2xFac_EnergyStorageType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	hydrogenStorage	0x40	Bit 6: hydrogen storage
	bit	electricEnergyStorage	0x20	Bit 5: electric energy storage
	bit	liquidPropaneGas	0x10	Bit 4: liquid propane gas
	bit	compressedNaturalGas	0x08	Bit 3: compressed natural gas
	bit	diesel	0x04	Bit 2: diesel
	bit	gasoline	0x02	Bit 1: gasoline
	bit	ammonia	0x01	Bit 0 (LSB): ammonia
Description	BitString DE_EnergyStorage as defined in ETSI TS 102 894-2.			
Variation	–			

▽

△

Available via	Rte_V2xFac_Type.h

⌋

[SWS_V2xFac_00218] Definition of ImplementationDataType V2xFac_StationaryVehicleContainerPresenceType ⌈

Name	V2xFac_StationaryVehicleContainerPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	stationarySince	0x20	Bit 5: Optional child present
	bit	stationaryCause	0x10	Bit 4: Optional child present
	bit	carryingDangerousGoods	0x08	Bit 3: Optional child present
	bit	numberOfOccupants	0x04	Bit 2: Optional child present
	bit	vehicleIdentification	0x02	Bit 1: Optional child present
	bit	energyStorageType	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_StationaryVehicleContainerType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00230] Definition of ImplementationDataType V2xFac_VehicleIdentificationType ⌈

Name	V2xFac_VehicleIdentificationType	
Kind	Structure	
Elements	presence	
	Type	V2xFac_VehicleIdentificationPresenceType
	Comment	Mark optional childs present or not
	wmiNumber	
	Type	V2xFac_WmiNumberType
	Comment	World Manufacturer Identifier (WMI)
	vds	
	Type	V2xFac_VdsType
	Comment	Vehicle Descriptor Section (VDS)
Description	DF_VehicleIdentification as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00223] Definition of ImplementationDataType V2xFac_VehicleIdentificationPresenceType ⌈

Name	V2xFac_VehicleIdentificationPresenceType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	wmiNumber	0x02	Bit 1: Optional child present
	bit	vds	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for V2xFac_VehicleIdentificationType			
Variation	–			
Available via	Rte_V2xFac_Type.h			

⌋

[SWS_V2xFac_00243] Definition of ImplementationDataType V2xFac_WmiNumberType ⌈

Name	V2xFac_WmiNumberType	
Kind	Structure	
Elements	count	
	Type	uint8
	Comment	Number of valid elements within array.
	values	
	Type	Array of uint8
	Size	3
	Comment	–
Description	DE_WMInumber as defined in ETSI TS 102 894-2. Size of the Array shall be 3.	
Variation	–	
Available via	Rte_V2xFac_Type.h	

⌋

[SWS_V2xFac_00244] Definition of ImplementationDataType V2xFac_VdsType ⌈

Name	V2xFac_VdsType	
Kind	Structure	
Elements	count	
	Type	uint8
	Comment	Number of valid elements within array.
	values	
	Type	Array of uint8
	Size	6
	Comment	–
Description	DE_VDS as defined in ETSI TS 102 894-2. Size of the Array shall be 6.	
Variation	–	

▽

△

Available via	Rte_V2xFac_Type.h

⌋

8.7.4 Ports

8.7.4.1 V2xFac_DenBs

[SWS_V2xFac_00102] Definition of Port V2xFac_DenBs provided by module V2x Fac ⌈

Name	V2xFac_DenBs		
Kind	ProvidedPort	***Interface***	V2xFacDenBs
Description	Service port for DEN specific service requests		
Variation	–		

⌋

8.7.4.2 V2xFac_Vdp

[SWS_V2xFac_00105] Definition of Port V2xFac_Vdp required by module V2xFac ⌈

Name	V2xFac_Vdp		
Kind	RequiredPort	***Interface***	V2xFacVdp
Description	Port for retrieving data from VDP application		
Variation	–		

⌋

8.7.4.3 V2xFac_VdpBasicLFC

[SWS_V2xFac_91610] Definition of Port V2xFac_VdpBasicLFC required by module V2xFac ⌈

Name	V2xFac_VdpBasicLFC		
Kind	RequiredPort	***Interface***	V2xFacVdpBasicLFC
Description	Port for retrieving data from the VDP application to provide low frequency container.		
Variation	–		

⌋

9 Sequence diagrams

9.1 CAM Generation and Transmission

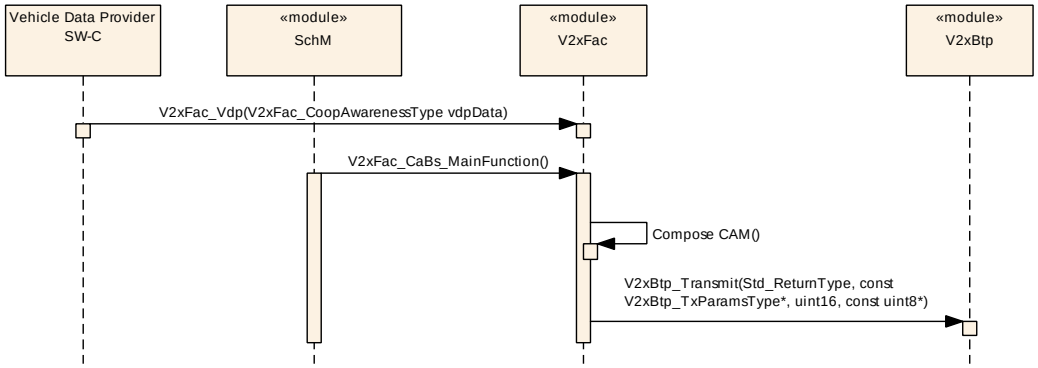

Figure 9.1: CAM Generation and Transmission

9.2 DENM Generation and Transmission

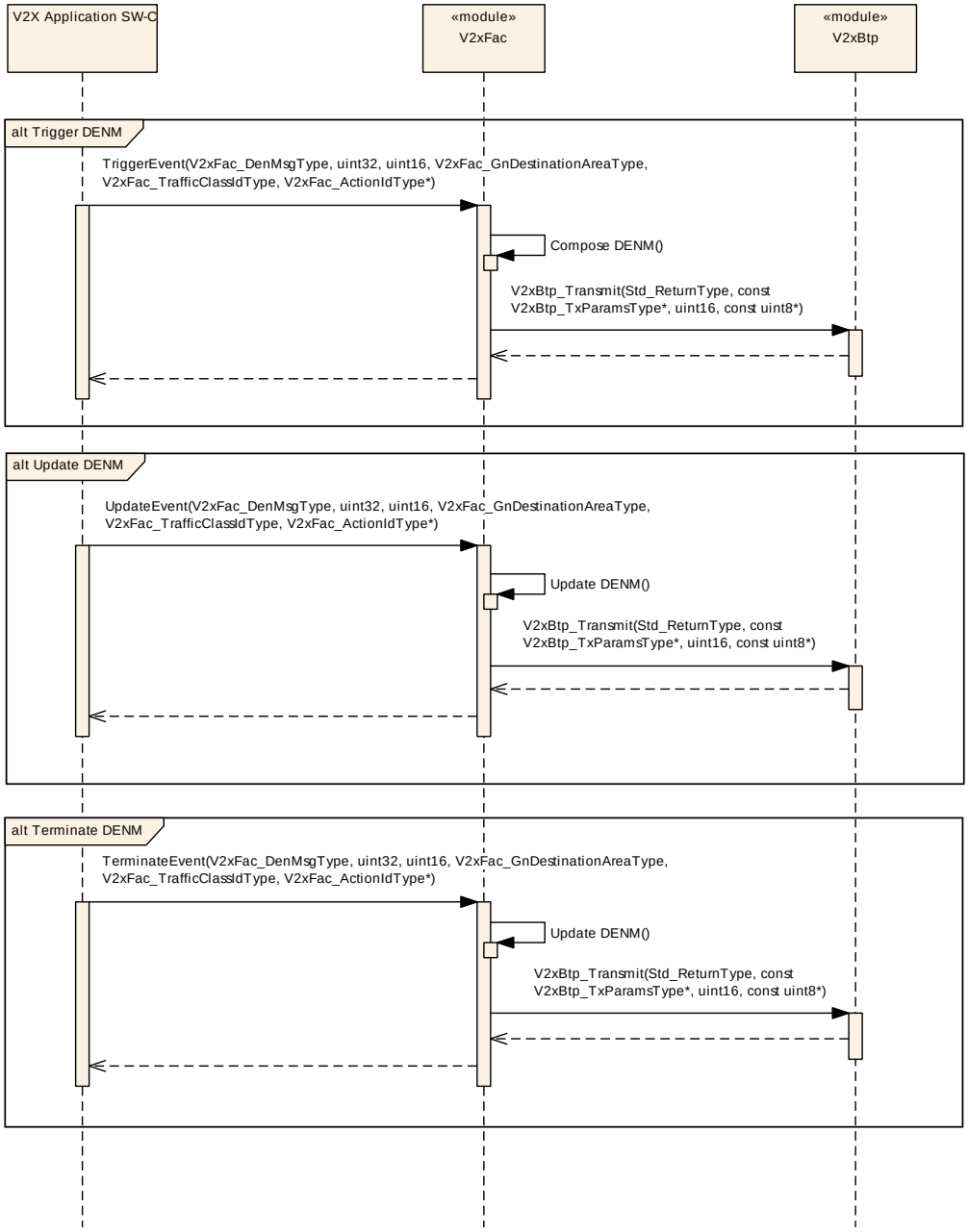

Figure 9.2: DENM Generation and Transmission

9.3 Message Reception

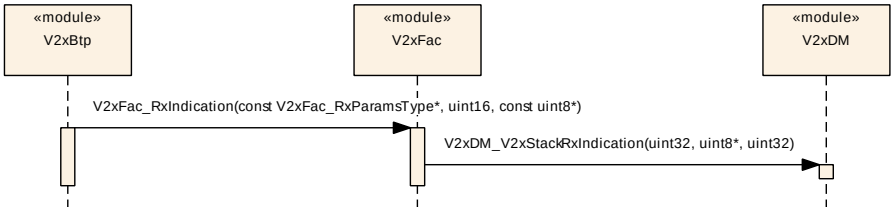

Figure 9.3: Message Reception

10 Configuration specification

Chapter 10.2 specifies the structure (containers) and the parameters of the module V2xFac.

Chapter 10.3 specifies published information of the module V2xFac.

10.1 How to read this chapter

For details refer to the chapter 10.1 “Introduction to configuration specification” in [8].

10.2 Containers and configuration parameters

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

10.2.1 Variants

[SWS_V2xFac_00238]

Upstream requirements: SRS_BSW_00345

⌈The V2xFac module only supports VARIANT-PRE-COMPILE.⌋

10.2.2 V2xFac

[ECUC_V2xFac_00001] Definition of EcucModuleDef V2xFac ⌈

Module Name	V2xFac
Description	Configuration of the V2xFac module.
Post-Build Variant Support	false
Supported Config Variants	VARIANT-PRE-COMPILE

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
V2xFacConfig	1	This container contains the configuration parameters and sub containers of the AUTOSAR V2xFac module.
V2xFacGeneral	1	This container contains the general configuration parameters of the Vehicle-2-X Facilities

⌋

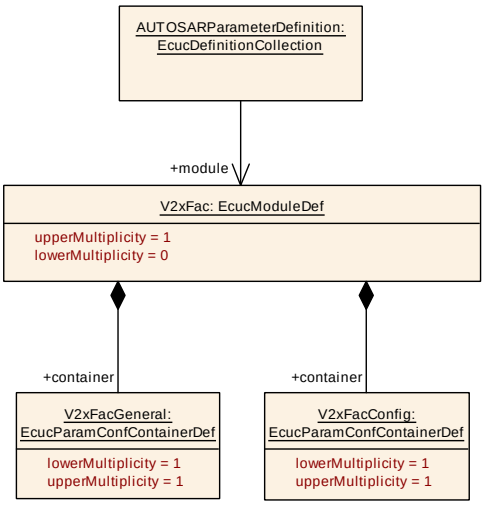

Figure 10.1: ECU Configuration V2xFacilities

10.2.3 V2xFacGeneral

[ECUC_V2xFac_00002] Definition of EcucParamConfContainerDef V2xFacGeneral ⌈

Container Name	V2xFacGeneral
Parent Container	V2xFac
Description	This container contains the general configuration parameters of the Vehicle-2-X Facilities
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
V2xFacCaBsMainFunctionPeriod	1	[ECUC_V2xFac_00006]
V2xFacDenBsMainFunctionPeriod	1	[ECUC_V2xFac_00005]
V2xFacDevErrorDetect	1	[ECUC_V2xFac_00004]
V2xFacRxSMainFunctionPeriod	1	[ECUC_V2xFac_00015]
V2xFacStationType	1	[ECUC_V2xFac_00007]
V2xFacVersionInfoApi	1	[ECUC_V2xFac_00003]

No Included Containers

⌋

[ECUC_V2xFac_00006] Definition of EcucFloatParamDef V2xFacCaBsMainFunctionPeriod ⌈

Parameter Name	V2xFacCaBsMainFunctionPeriod		
Parent Container	V2xFacGeneral		
Description	This parameter defines the schedule period of V2xFac_CaBs_MainFunction.Unit: [s]		
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		

⌋

[ECUC_V2xFac_00005] Definition of EcucFloatParamDef V2xFacDenBsMainFunctionPeriod ⌈

Parameter Name	V2xFacDenBsMainFunctionPeriod		
Parent Container	V2xFacGeneral		
Description	This parameter defines the schedule period of V2xFac_DenBs_MainFunction.Unit: [s]		
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		

⌋

[ECUC_V2xFac_00004] Definition of EcucBooleanParamDef V2xFacDevErrorDetect ⌈

Parameter Name	V2xFacDevErrorDetect
Parent Container	V2xFacGeneral
Description	Switches the Default Error Tracer (Det) detection and notification ON or OFF. • 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		

⌋

[ECUC_V2xFac_00015] Definition of EcucFloatParamDef V2xFacRxSMainFunctionPeriod ⌈

Parameter Name	V2xFacRxSMainFunctionPeriod		
Parent Container	V2xFacGeneral		
Description	This parameter defines the schedule period of V2xFac_RxS_MainFunction.Unit: [s]		
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		

⌋

[ECUC_V2xFac_00007] Definition of EcucEnumerationParamDef V2xFacStationType ⌈

Parameter Name	V2xFacStationType	
Parent Container	V2xFacGeneral	
Description	This configuration value defines the station type information of the originating ITS-S, RoadSideUnit (15) not supported by AUTOSAR.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	V2XFAC_ST_BUS	–
	V2XFAC_ST_CYCLIST	–
	V2XFAC_ST_HEAVYTRUCK	–
	V2XFAC_ST_LIGHTTRUCK	–
	V2XFAC_ST_MOPED	–
	V2XFAC_ST_MOTORCYCLE	–
	V2XFAC_ST_PASSENGERCAR	–
	V2XFAC_ST_PEDESTRIAN	–
	V2XFAC_ST_ SPECIALVEHICLES	–
	V2XFAC_ST_TRAILER	–

▽

△

	V2XFAC_ST_TRAM	–	
	V2XFAC_ST_UNKNOWN	–	
Default value	V2XFAC_ST_UNKNOWN		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_V2xFac_00003] Definition of EcucBooleanParamDef V2xFacVersionInfo Api ⌈

Parameter Name	V2xFacVersionInfoApi		
Parent Container	V2xFacGeneral		
Description	Enable/disables the API for reading the version information of the V2xFac Module. • 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		

⌋

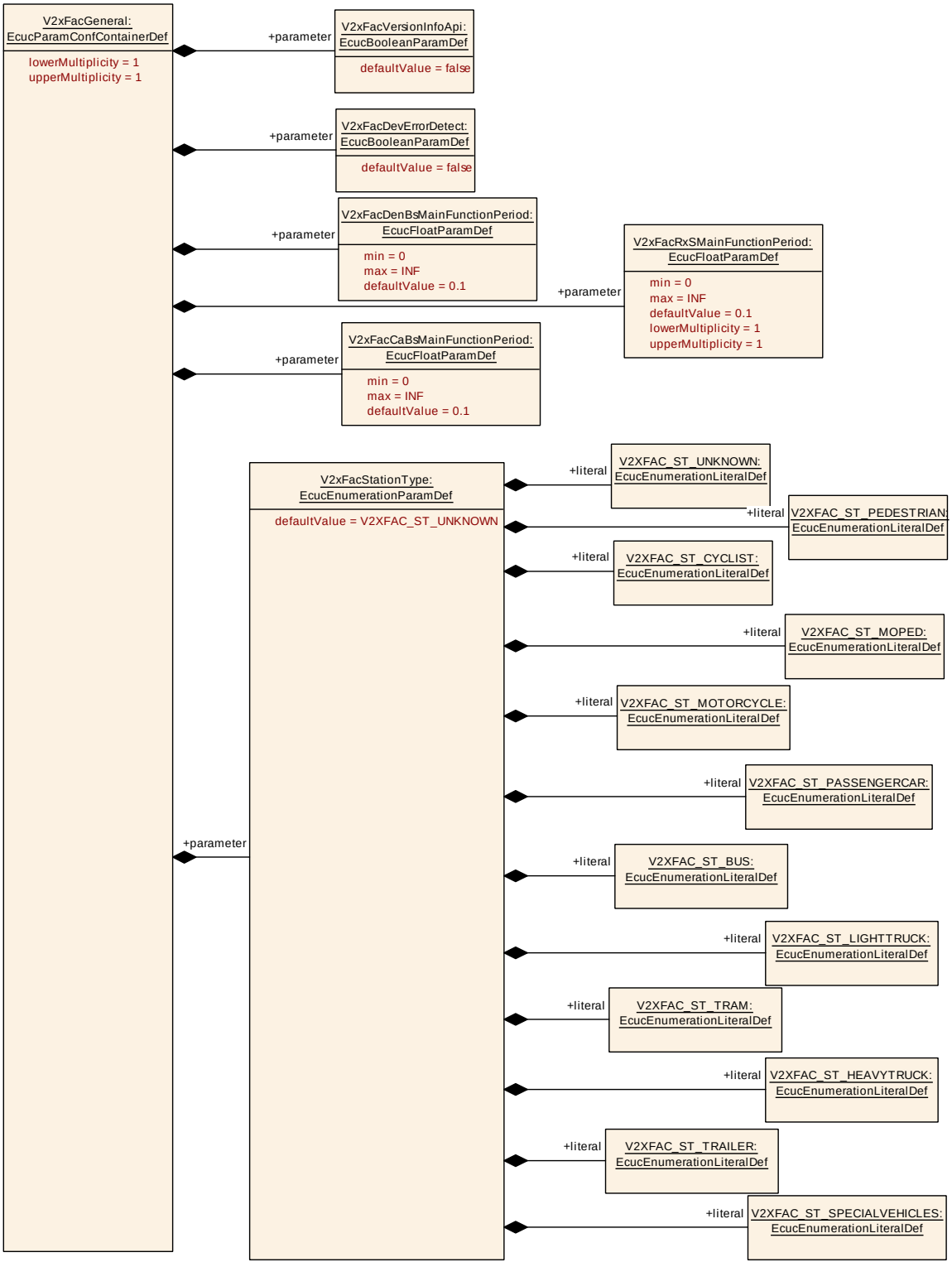

Figure 10.2: ECU Configuration V2xFacGeneral

10.2.4 V2xFacConfig

[ECUC_V2xFac_00011] Definition of EcucParamConfContainerDef V2xFacConfig ⌈

Container Name	V2xFacConfig
Parent Container	V2xFac
Description	This container contains the configuration parameters and sub containers of the AUTOSAR V2xFac module.
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
V2xFacDmMsgConfig	1..*	This container contains the configuration of all messages that shall be passed on to the V2x Data Manager.

⌋

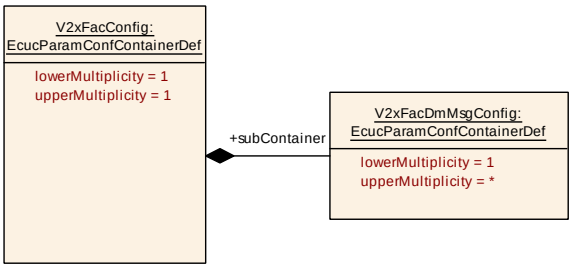

Figure 10.3: ECU Configuration V2xFacConfig

10.2.5 V2xFacDmMsgConfig

[ECUC_V2xFac_00012] Definition of EcucParamConfContainerDef V2xFacDmMsgConfig ⌈

Container Name	V2xFacDmMsgConfig		
Parent Container	V2xFacConfig		
Description	This container contains the configuration of all messages that shall be passed on to the V2x Data Manager.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
V2xFacDmPortId	1	[ECUC_V2xFac_00013]
V2xFacV2xMessageId	1	[ECUC_V2xFac_00016]
V2xFacDmMsgRef	1	[ECUC_V2xFac_00014]

No Included Containers

⌋

[ECUC_V2xFac_00013] Definition of EcucIntegerParamDef V2xFacDmPortId ⌈

Parameter Name	V2xFacDmPortId		
Parent Container	V2xFacDmMsgConfig		
Description	This parameter contains the destination port identifier that references the V2x message.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_V2xFac_00016] Definition of EcucIntegerParamDef V2xFacV2xMessageId ⌈

Parameter Name	V2xFacV2xMessageId		
Parent Container	V2xFacDmMsgConfig		
Description	This parameter contains the message identifier related to the V2X message that can be received on the associated BTP port as specified in ETSI ITS 102 894-2. Note: Nothing precludes to use several message configurations to receive the same type of message on several BTP ports or several messages on the same BTP port, even if this is not intended by the standard.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_V2xFac_00014] Definition of EcucReferenceDef V2xFacDmMsgRef ⌈

Parameter Name	V2xFacDmMsgRef		
Parent Container	V2xFacDmMsgConfig		
Description	Symbolic name reference to [V2xDMMessage]		
Multiplicity	1		
Type	Symbolic name reference to V2xDMMessage		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

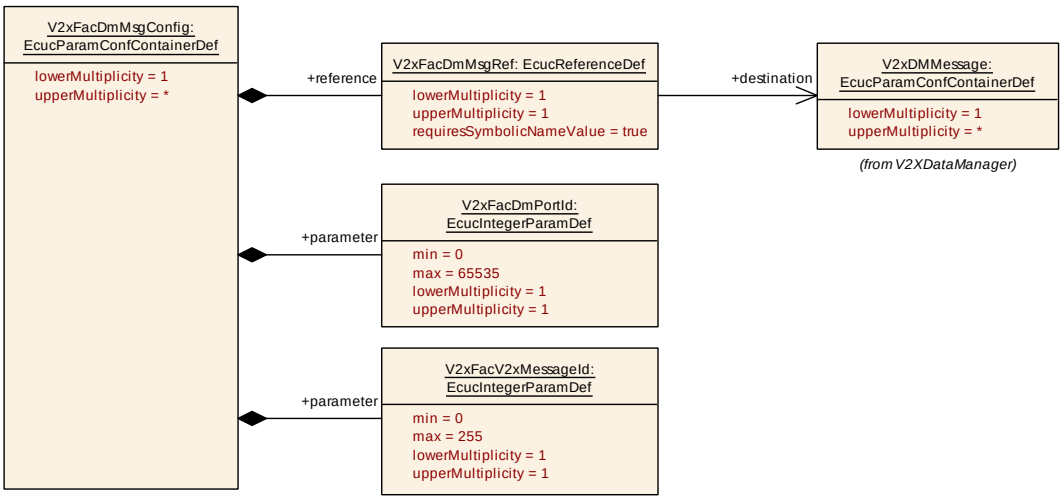

Figure 10.4: ECU Configuration V2XFacDmMsgConfig

10.3 Published Information

For details refer to the chapter 10.3 “Published Information” in [8].

A Not applicable requirements

N/A

B History of Specification Items

Please note that the lists in this chapter also include specification items that have been removed from the specification in a later version. These specification items do not appear as hyperlinks in the document.

B.1 Specification Item History of this document compared to AUTOSAR R24-11.

B.1.1 Added Specification Items in R24-11

Number	Heading
[SWS_V2xFac_-00050]	Definition of ImplementationDataType V2xFac_VehicleLengthType
[SWS_V2xFac_-00051]	Definition of ImplementationDataType V2xFac_LongitudinalAccelerationType
[SWS_V2xFac_-00052]	Definition of ImplementationDataType V2xFac_CurvatureType
[SWS_V2xFac_-00053]	Definition of ImplementationDataType V2xFac_YawRateType
[SWS_V2xFac_-00054]	Definition of ImplementationDataType V2xFac_SteeringWheelAngleType
[SWS_V2xFac_-00055]	Definition of ImplementationDataType V2xFac_LateralAccelerationType
[SWS_V2xFac_-00056]	Definition of ImplementationDataType V2xFac_VerticalAccelerationType
[SWS_V2xFac_-00060]	Definition of ImplementationDataType V2xFac_PathPointType
[SWS_V2xFac_-00173]	Definition of ImplementationDataType V2xFac_BasicVehicleContainerHighFrequencyType
[SWS_V2xFac_-00174]	Definition of ImplementationDataType V2xFac_BasicVehicleContainerHighFrequencyPresenceType
[SWS_V2xFac_-00175]	Definition of ImplementationDataType V2xFac_DriveDirectionType
[SWS_V2xFac_-00176]	Definition of ImplementationDataType V2xFac_CurvatureCalculationModeType
[SWS_V2xFac_-00177]	Definition of ImplementationDataType V2xFac_AccelerationControlType
[SWS_V2xFac_-00184]	Definition of ImplementationDataType V2xFac_CurvartureConfidenceType
[SWS_V2xFac_-00187]	Definition of ImplementationDataType V2xFac_BasicVehicleContainerLowFrequencyType

△

Number	Heading
[SWS_V2xFac_-00188]	Definition of ImplementationDataType V2xFac_VehicleRoleType
[SWS_V2xFac_-00189]	Definition of ImplementationDataType V2xFac_ExteriorLightsType
[SWS_V2xFac_-00190]	Definition of ImplementationDataType V2xFac_PathPointPresenceType
[SWS_V2xFac_-00239]	Definition of ImplementationDataType V2xFac_VehicleLengthConfidence IndicationType
[SWS_V2xFac_-00245]	Definition of ImplementationDataType V2xFac_YawRateConfidenceType
[SWS_V2xFac_-00321]	Reception of a New High Frequency Container
[SWS_V2xFac_-00322]	Use last received Low Frequency Container on VDP request only
[SWS_V2xFac_-00323]	Low Frequency Container reception
[SWS_V2xFac_-91609]	Definition of SenderReceiverInterface V2xFacVdpBasicLFC
[SWS_V2xFac_-91610]	Definition of Port V2xFac_VdpBasicLFC required by module V2xFac

Table B.1: Added Specification Items in R24-11

B.1.2 Changed Specification Items in R24-11

Number	Heading
[SWS_V2xFac_-00095]	Definition of SenderReceiverInterface V2xFacVdp

Table B.2: Changed Specification Items in R24-11

B.1.3 Deleted Specification Items in R24-11

Number	Heading
[SWS_V2xFac_-00166]	Definition of ImplementationDataType V2xFac_ClosedLanesPresenceType
[SWS_V2xFac_-00167]	Definition of ImplementationDataType V2xFac_HardShoulderStatusType

▽

△

Number	Heading
[SWS_V2xFac_-00168]	Definition of ImplementationDataType V2xFac_DrivingLaneStatusType
[SWS_V2xFac_-00226]	Definition of ImplementationDataType V2xFac_ClosedLanesType

Table B.3: Deleted Specification Items in R24-11

B.2 Specification Item History of this document compared to AUTOSAR R23-11.

B.2.1 Added Specification Items in R23-11

Number	Heading
[SWS_V2xFac_-00001]	
[SWS_V2xFac_-00004]	
[SWS_V2xFac_-00006]	
[SWS_V2xFac_-00008]	
[SWS_V2xFac_-00009]	
[SWS_V2xFac_-00010]	
[SWS_V2xFac_-00011]	
[SWS_V2xFac_-00014]	
[SWS_V2xFac_-00015]	
[SWS_V2xFac_-00016]	
[SWS_V2xFac_-00019]	
[SWS_V2xFac_-00025]	
[SWS_V2xFac_-00027]	

▽

△

Number	Heading
[SWS_V2xFac_-00029]	
[SWS_V2xFac_-00031]	Definiton of development errors in module V2xFac
[SWS_V2xFac_-00032]	Definition of imported datatypes of module V2xFac
[SWS_V2xFac_-00034]	Definition of datatype V2xFac_RxParamsType
[SWS_V2xFac_-00036]	Definition of ImplementationDataType V2xFac_ItsPduHeaderType
[SWS_V2xFac_-00037]	Definition of ImplementationDataType V2xFac_AltitudeType
[SWS_V2xFac_-00038]	Definition of ImplementationDataType V2xFac_PosConfidenceEllipseType
[SWS_V2xFac_-00039]	Definition of ImplementationDataType V2xFac_HeadingType
[SWS_V2xFac_-00040]	Definition of ImplementationDataType V2xFac_SpeedType
[SWS_V2xFac_-00047]	Definition of ImplementationDataType V2xFac_ReferencePositionType
[SWS_V2xFac_-00059]	Definition of ImplementationDataType V2xFac_PathHistoryType
[SWS_V2xFac_-00069]	Definition of ImplementationDataType V2xFac_DenmMessageRootType
[SWS_V2xFac_-00070]	Definition of ImplementationDataType V2xFac_DenMsgType
[SWS_V2xFac_-00071]	Definition of ImplementationDataType V2xFac_ManagementContainerType
[SWS_V2xFac_-00073]	Definition of ImplementationDataType V2xFac_SituationContainerType
[SWS_V2xFac_-00074]	Definition of ImplementationDataType V2xFac_CauseCodeType
[SWS_V2xFac_-00075]	Definition of ImplementationDataType V2xFac_EventHistoryType
[SWS_V2xFac_-00076]	Definition of ImplementationDataType V2xFac_EventPointType
[SWS_V2xFac_-00077]	Definition of ImplementationDataType V2xFac_LocationContainerType
[SWS_V2xFac_-00078]	Definition of ImplementationDataType V2xFac_AlacarteContainerType
[SWS_V2xFac_-00079]	Definition of ImplementationDataType V2xFac_ImpactReductionContainer Type
[SWS_V2xFac_-00081]	Definition of ImplementationDataType V2xFac_StationaryVehicleContainer Type

▽

△

Number	Heading
[SWS_V2xFac_-00082]	Definition of API function V2xFac_Init
[SWS_V2xFac_-00084]	Definition of API function V2xFac_GetVersionInfo
[SWS_V2xFac_-00085]	
[SWS_V2xFac_-00086]	Definition of API function V2xFac_V2xM_PreparePseudonymChange
[SWS_V2xFac_-00088]	Definition of callback function V2xFac_RxIndication
[SWS_V2xFac_-00090]	Definition of scheduled function V2xFac_CaBs_MainFunction
[SWS_V2xFac_-00091]	Definition of scheduled function V2xFac_DenBs_MainFunction
[SWS_V2xFac_-00092]	Definition of mandatory interfaces in module V2xFac
[SWS_V2xFac_-00093]	Definition of optional interfaces in module V2xFac
[SWS_V2xFac_-00094]	
[SWS_V2xFac_-00095]	Definition of SenderReceiverInterface V2xFacVdp
[SWS_V2xFac_-00098]	
[SWS_V2xFac_-00099]	Definition of ClientServerInterface V2xFacDenBs
[SWS_V2xFac_-00102]	Definition of Port V2xFac_DenBs provided by module V2xFac
[SWS_V2xFac_-00105]	Definition of Port V2xFac_Vdp required by module V2xFac
[SWS_V2xFac_-00106]	
[SWS_V2xFac_-00116]	
[SWS_V2xFac_-00136]	
[SWS_V2xFac_-00137]	
[SWS_V2xFac_-00138]	
[SWS_V2xFac_-00140]	Definition of API function V2xFac_V2xM_CommitPseudonymChange
[SWS_V2xFac_-00141]	

▽

△

Number	Heading
[SWS_V2xFac_-00142]	
[SWS_V2xFac_-00144]	Definition of API function V2xFac_V2xM_AbortPseudonymChange
[SWS_V2xFac_-00145]	
[SWS_V2xFac_-00146]	
[SWS_V2xFac_-00148]	Definition of API function V2xFac_V2xM_SetTGenCamDcc
[SWS_V2xFac_-00149]	
[SWS_V2xFac_-00150]	
[SWS_V2xFac_-00152]	Definition of API function V2xFac_V2xM_SetCaBsOperation
[SWS_V2xFac_-00153]	
[SWS_V2xFac_-00154]	
[SWS_V2xFac_-00158]	
[SWS_V2xFac_-00159]	
[SWS_V2xFac_-00160]	
[SWS_V2xFac_-00162]	Definition of ImplementationDataType V2xFac_TrafficClassIdType
[SWS_V2xFac_-00163]	Definition of ImplementationDataType V2xFac_GnDestinationAreaType
[SWS_V2xFac_-00164]	Definition of ImplementationDataType V2xFac_GnAreaShapeType
[SWS_V2xFac_-00165]	Definition of ImplementationDataType V2xFac_AltitudeConfidenceType
[SWS_V2xFac_-00166]	Definition of ImplementationDataType V2xFac_ClosedLanesPresenceType
[SWS_V2xFac_-00167]	Definition of ImplementationDataType V2xFac_HardShoulderStatusType
[SWS_V2xFac_-00168]	Definition of ImplementationDataType V2xFac_DrivingLaneStatusType
[SWS_V2xFac_-00199]	Definition of ImplementationDataType V2xFac_DenMsgPresenceType
[SWS_V2xFac_-00200]	Definition of ImplementationDataType V2xFac_RelevanceDistanceType

▽

△

Number	Heading
[SWS_V2xFac_-00201]	Definition of ImplementationDataType V2xFac_RelevanceTrafficDirection Type
[SWS_V2xFac_-00202]	Definition of ImplementationDataType V2xFac_ManagementContainer PresenceType
[SWS_V2xFac_-00203]	Definition of ImplementationDataType V2xFac_SituationContainerPresence Type
[SWS_V2xFac_-00204]	Definition of ImplementationDataType V2xFac_EventPointPresenceType
[SWS_V2xFac_-00205]	Definition of ImplementationDataType V2xFac_TracesType
[SWS_V2xFac_-00206]	Definition of ImplementationDataType V2xFac_LocationContainerPresence Type
[SWS_V2xFac_-00207]	Definition of ImplementationDataType V2xFac_PositioningSolutionTypeType
[SWS_V2xFac_-00208]	Definition of ImplementationDataType V2xFac_AlacarteContainerPresence Type
[SWS_V2xFac_-00209]	Definition of ImplementationDataType V2xFac_PositionOfPillarsType
[SWS_V2xFac_-00210]	Definition of ImplementationDataType V2xFac_PositionOfOccupantsType
[SWS_V2xFac_-00216]	Definition of ImplementationDataType V2xFac_StationarySinceType
[SWS_V2xFac_-00217]	Definition of ImplementationDataType V2xFac_EnergyStorageType
[SWS_V2xFac_-00218]	Definition of ImplementationDataType V2xFac_StationaryVehicleContainer PresenceType
[SWS_V2xFac_-00223]	Definition of ImplementationDataType V2xFac_VehicleIdentificationPresence Type
[SWS_V2xFac_-00224]	Definition of ImplementationDataType V2xFac_DeltaReferencePositionType
[SWS_V2xFac_-00225]	Definition of ImplementationDataType V2xFac_ActionIdType
[SWS_V2xFac_-00226]	Definition of ImplementationDataType V2xFac_ClosedLanesType
[SWS_V2xFac_-00230]	Definition of ImplementationDataType V2xFac_VehicleIdentificationType
[SWS_V2xFac_-00231]	
[SWS_V2xFac_-00232]	
[SWS_V2xFac_-00238]	

▽

△

Number	Heading
[SWS_V2xFac_-00240]	Definition of ImplementationDataType V2xFac_TerminationType
[SWS_V2xFac_-00241]	Definition of ImplementationDataType V2xFac_RoadTypeType
[SWS_V2xFac_-00242]	Definition of ImplementationDataType V2xFac_RequestResponseIndicationType
[SWS_V2xFac_-00243]	Definition of ImplementationDataType V2xFac_WmiNumberType
[SWS_V2xFac_-00244]	Definition of ImplementationDataType V2xFac_VdsType
[SWS_V2xFac_-00247]	
[SWS_V2xFac_-00248]	
[SWS_V2xFac_-00294]	
[SWS_V2xFac_-00295]	
[SWS_V2xFac_-00296]	
[SWS_V2xFac_-00300]	
[SWS_V2xFac_-00301]	
[SWS_V2xFac_-00303]	
[SWS_V2xFac_-00304]	
[SWS_V2xFac_-00305]	Definition of scheduled function V2xFac_RxS_MainFunction
[SWS_V2xFac_-00306]	
[SWS_V2xFac_-00307]	
[SWS_V2xFac_-20168]	
[SWS_V2xFac_-20185]	
[SWS_V2xFac_-20215]	
[SWS_V2xFac_-20256]	
[SWS_V2xFac_-20257]	

▽

△

Number	Heading
[SWS_V2xFac_-20259]	
[SWS_V2xFac_-20285]	
[SWS_V2xFac_-20286]	
[SWS_V2xFac_-20287]	
[SWS_V2xFac_-20288]	
[SWS_V2xFac_-20289]	
[SWS_V2xFac_-20291]	
[SWS_V2xFac_-20292]	
[SWS_V2xFac_-20297]	
[SWS_V2xFac_-20302]	
[SWS_V2xFac_-20303]	
[SWS_V2xFac_-20304]	
[SWS_V2xFac_-20305]	
[SWS_V2xFac_-20306]	
[SWS_V2xFac_-20307]	
[SWS_V2xFac_-20308]	
[SWS_V2xFac_-20313]	
[SWS_V2xFac_-20318]	
[SWS_V2xFac_-20444]	
[SWS_V2xFac_-20445]	

Table B.4: Added Specification Items in R23-11

B.2.2 Changed Specification Items in R23-11

none

B.2.3 Deleted Specification Items in R23-11

none