Document Title	Specification of a Request Manager for SAE J1939
Document Category	Software specification (SWS)
Document UID	611
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Improved description of `J1939Rm_TxConfirmation` • Removed `J1939Rm_GetVersionInfo` and `J1939RmVersionInfoApi` • Adapted to current template and improved EcuC tables, removed references to RS SAE J1939
2025-11-27	R25-11	AUTOSAR Release Management	• Improved uptracing to RS BSW General
2024-11-27	R24-11	AUTOSAR Release Management	• Clarified internal feedback of RQST(2) and ACKM • Fixed included header for CDD user callouts
2023-11-23	R23-11	AUTOSAR Release Management	• Fixed include file for configurable callouts • Fixed reference to `J1939Rm_CheckReceivedComIPdu`, fixed name of `J1939RmRequest2QueueSize`, fixed description of `destAddress` of `J1939Rm_CancelRequestTimeout` • Added information about automatic handle IDs to configuration • `J1939Rm_AckCode` and `J1939Rm_ExtIdType` changed to `uint8`
2022-11-24	R22-11	AUTOSAR Release Management	• No content changes

△

2021-11-25	R21-11	AUTOSAR Release Management	• Converted to LaTeX • Fixed UserId parameter range • Extended acronym and related documents tables • Improved linking of terms
2020-11-30	R20-11	AUTOSAR Release Management	• Improved structure of error sections • Replaced error descriptions with generated tables
2019-11-28	R19-11	AUTOSAR Release Management	• Cleaned up EcuC diagrams • Improved service port tables • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Changed header file structure • Improved name of J1939Rm_ComRxIpduCallout • Harmonized J1939RM_E_UNINIT • Routing of RQST/RQST2/ACKM
2017-12-08	4.3.1	AUTOSAR Release Management	• Clarified availability of J1939Rm_ComRxIpduCallout • Added internal feedback of ACKM • Clarification of extIdInfo parameter and underlying standard • Improved parameter checks
2016-11-30	4.3.0	AUTOSAR Release Management	• Request2 support • Improved handling of meta data • Reliable TxConfirmation replaces timeout • Separate configuration of different users
2015-07-31	4.2.2	AUTOSAR Release Management	• Fixed names and signatures of service ports • Support for explicit broadcast of ACKM • Introduction of further error classes
2014-10-31	4.2.1	AUTOSAR Release Management	• Improved interaction with COM • Harmonized with SWS BSW General

▽

△

2014-03-31	4.1.3	AUTOSAR Release Management	• Clarified availability of callbacks • Standardized callback header names • Fixed UserType Enum
2013-10-31	4.1.2	AUTOSAR Release Management	• Additional development error for function parameter checks • Clarification of Request timeout and state handling • Separate configuration of received and transmitted PGNs • Removed change documentation
2013-03-15	4.1.1	AUTOSAR Administration	• Initial Release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction and Functional Overview 9
1.1 Request Management according to SAE J1939 . . . 9
1.2 SAE J1939 Request Manager BSW Module . . . 9
1.3 J1939 Terminology . . . 10
2 Acronyms, Abbreviations, and Terms 11
2.1 Definition of Acronyms and abbreviations . . . 11
2.2 Definition of Terms . . . 13
3 Related Documentation 15
3.1 Input Documents & Related Standards and Norms . . . 15
3.2 Related Specification . . . 16
4 Constraints and Assumptions 17
4.1 Limitations . . . 17
4.2 Applicability to Automotive Domains . . . 17
5 Dependencies to Other Modules 18
5.1 File Structure . . . 18
5.1.1 Code File Structure . . . 18
5.1.2 Header File Structure . . . 18
6 Requirements Tracing 20
7 Functional Specification 21
7.1 Overview . . . 21
7.2 Module Handling . . . 21
7.2.1 Initialization . . . 21
7.2.2 Timing Related Functionality . . . 22
7.3 Communication State Handling . . . 22
7.4 Reception of Requests . . . 22
7.4.1 Request Forwarding . . . 23
7.4.2 Request Handling via COM . . . 24
7.4.3 Request of Unknown PGNs . . . 24
7.5 Transmission of Acknowledgements . . . 24
7.6 Transmission of Requests . . . 25
7.7 Reception of Acknowledgements . . . 27
7.8 Timeout Supervision . . . 27
7.9 Routing of Requests and Acknowledgements . . . 28
7.10 Error Classification . . . 29
7.10.1 Development Errors . . . 29
7.10.2 Runtime Errors . . . 30
7.10.3 Production Errors . . . 30

8 API Specification 31
8.1 API Parameter Checking . . . 31
8.2 Imported Types . . . 31
8.3 Type Definitions . . . 32
8.3.1 J1939Rm_ConfigType . . . 32
8.3.2 J1939Rm_StateType . . . 32
8.4 Function Definitions . . . 33
8.4.1 Generally Available Functions . . . 33
8.4.1.1 J1939Rm_Init . . . 33
8.4.1.2 J1939Rm_DeInit . . . 33
8.4.1.3 J1939Rm_SetState . . . 34
8.4.1.4 J1939Rm_SendRequest . . . 35
8.4.1.5 J1939Rm_CancelRequestTimeout . . . 36
8.4.1.6 J1939Rm_SendAck . . . 37
8.4.1.7 J1939Rm_RxIndication . . . 39
8.4.1.8 J1939Rm_TxConfirmation . . . 39
8.4.2 Functions Provided for COM . . . 40
8.4.2.1 J1939Rm_CheckReceivedComIPdu . . . 40
8.5 Scheduled Functions . . . 41
8.5.1 J1939Rm_MainFunction . . . 41
8.6 Expected Interfaces . . . 42
8.6.1 Mandatory Interfaces . . . 42
8.6.2 Optional Interfaces . . . 42
8.6.3 Configurable Interfaces . . . 43
8.6.3.1 <User>_RequestIndication . . . 43
8.6.3.2 <User>_AckIndication . . . 44
8.6.3.3 <User>_RequestTimeoutIndication . . . 45
8.7 Service Interfaces . . . 45
8.7.1 Client-Server Interfaces . . . 45
8.7.1.1 AppSendAck . . . 46
8.7.1.2 AppSendRequest . . . 47
8.7.1.3 AppCancelRequestTimeout . . . 48
8.7.1.4 AppAckIndication . . . 49
8.7.1.5 AppRequestIndication . . . 50
8.7.1.6 AppRequestTimeoutIndication . . . 52
8.7.2 Implementation Data Types . . . 53
8.7.2.1 J1939Rm_AckCode . . . 53
8.7.2.2 J1939Rm_ExtIdType . . . 53
8.7.2.3 J1939Rm_ExtIdInfoType . . . 54
8.7.3 Provided Service Ports . . . 54
8.7.3.1 J1939Rm_SendAck . . . 54
8.7.3.2 J1939Rm_SendRequest . . . 55
8.7.3.3 J1939Rm_CancelRequestTimeout . . . 55

8.7.4 Required Service Ports . . . 55
8.7.4.1 J1939Rm_AckIndication . . . 55
8.7.4.2 J1939Rm_RequestIndication . . . 56
8.7.4.3 J1939Rm_RequestTimeoutIndication . . . 56
9 Sequence Diagrams 57
9.1 Reception of Request PG . . . 57
9.2 Transmission of Acknowledgement PG . . . 57
9.3 Handling of Request for a COM PDU . . . 58
9.4 Handling of Request for a Diagnostic PDU . . . 58
9.5 Transmission of Request PG . . . 59
9.6 Reception of Acknowledgement PG . . . 59
9.7 Monitoring of Request Timeout . . . 60
10 Configuration Specification 61
10.1 Containers and Configuration Parameters . . . 61
10.1.1 J1939Rm . . . 69
10.1.2 J1939RmGeneral . . . 69
10.1.3 J1939RmConfigSet . . . 72
10.1.4 J1939RmChannel . . . 73
10.1.5 J1939RmAckmRxPdu . . . 75
10.1.6 J1939RmAckmTxPdu . . . 76
10.1.7 J1939RmRqstRxPdu . . . 78
10.1.8 J1939RmRqstTxPdu . . . 79
10.1.9 J1939RmRqst2RxPdu . . . 80
10.1.10 J1939RmRqst2TxPdu . . . 81
10.1.11 J1939RmNode . . . 82
10.1.12 J1939RmUser . . . 83
10.1.13 J1939RmNmUser . . . 84
10.1.14 J1939RmDcmUser . . . 84
10.1.15 J1939RmCddUser . . . 85
10.1.16 J1939RmRteUser . . . 89
10.1.17 J1939RmComUser . . . 90
10.1.18 J1939RmComIPdu . . . 91
A Not Applicable Requirements 94
B Change History of AUTOSAR Traceable Items 95
B.1 Traceable Item History of this Document According to AUTOSAR Release R26-11 . . . 95
B.1.1 Added Specification Items in R26-11 . . . 95
B.1.2 Changed Specification Items in R26-11 . . . 95
B.1.3 Deleted Specification Items in R26-11 . . . 95
B.2 Traceable Item History of this Document According to AUTOSAR Release R25-11 . . . 95
B.2.1 Added Specification Items in R25-11 . . . 95

B.2.2 Changed Specification Items in R25-11 95
B.2.3 Deleted Specification Items in R25-11 95
B.3 Traceable Item History of this Document According to AUTOSAR Release R24-11 96
B.3.1 Added Specification Items in R24-11 96
B.3.2 Changed Specification Items in R24-11 96
B.3.3 Deleted Specification Items in R24-11 96
B.4 Traceable Item History of this Document According to AUTOSAR Release R23-11 96
B.4.1 Added Specification Items in R23-11 96
B.4.2 Changed Specification Items in R23-11 96
B.4.3 Deleted Specification Items in R23-11 96
B.5 Traceable Item History of this Document According to AUTOSAR Release R22-11 96
B.5.1 Added Specification Items in R22-11 96
B.5.2 Changed Specification Items in R22-11 97
B.5.3 Deleted Specification Items in R22-11 97
B.6 Traceable Item History of this Document According to AUTOSAR Release R21-11 97
B.6.1 Added Specification Items in R21-11 97
B.6.2 Changed Specification Items in R21-11 97
B.6.3 Deleted Specification Items in R21-11 97

1 Introduction and Functional Overview

This specification describes the functionality, the API, and the configuration of the AUTOSAR Basic Software module SAE J1939 Request Manager.

1.1 Request Management according to SAE J1939

J1939 defines a special parameter group (PG) called Request (RQST, PGN = 0x0EA00), which may be used to request transmission of any other parameter group (see SAE J1939-21). The Request parameter group just contains the PGN of the requested parameter group.

Depending on the destination address used by the Request PG, the response must be sent directly to the requesting ECU, or to all ECUs. For short parameter groups with PDU1 format, the destination address is set accordingly[1], for large parameter groups the suitable transport protocol mode (BAM or CMDT of J1939Tp) is used.

Depending on the requested parameter group and the destination address of the Request PG, ECUs answer either with the requested parameter group, with the special Acknowledgement parameter group (ACKM, PGN = 0x0E800), or not at all.

Finally, J1939 defines that the response to a Request will be expected within 1.25s after the Request was sent. The responding node is required to answer within 200ms.

Besides the Request PG, J1939 also defines a Request2 PG (RQST2, PGN = 0x0C900). The behavior of this PG is identical to that of the Request PG, with the following extensions:

- A transmission with the Transfer PG can be requested to provide the same PG from multiple ECUs.
- Extended identifier bytes can be specified to request a defined layout of a multiplexed message.

1.2 SAE J1939 Request Manager BSW Module

The SAE J1939 Request Manager (J1939Rm) handles received and transmitted Request, Request2, and Acknowledgement PGs. It natively supports handling of incoming requests for the AddressClaimed PG and is configurable to support incoming requests for diagnostic and other J1939 PGNs. Unknown incoming requests are answered with a negative Acknowledgement PG if they address a specific destination address.

[1] Short parameter groups with PDU2 format have no destination address, they are broadcast PGs by nature.

The J1939Rm also supports transmission of requests and timeout supervision for the resulting PG or acknowledgement.

1.3 J1939 Terminology

The terminology of J1939 differs noticeably from the usual AUTOSAR terminology. For consistency reasons, this introduction used the terms of the J1939 specification, while the remainder of this specification will use terms that are more common within AUTOSAR:

- I-PDU replaces `parameter group`

2 Acronyms, Abbreviations, and Terms

2.1 Definition of Acronyms and abbreviations

The table below contains all acronyms, abbreviations or initialisms that are component-specific and not available in the AUTOSAR glossary [1].

Acronym/ Abbreviation	Description
AC	SAE J1939 AddressClaimed PG (PGN = 0x0EE00), which contains the NAME and is used for the SAE J1939 address claiming, CannotClaimAddress when SA = 0xFE
ACK	SAE J1939 Acknowledgement PG (ACKM) with control byte set to 0, positive acknowledgement
ACKM	SAE J1939 Acknowledgement PG (PGN = 0x0E800)
BswM	Basic Software Mode Manager, controlling modes of the whole BSW, see [2, SWS Basic Software Mode Manager]
CA	Controller Application, role of an ECU tied to one address
CanIf	CAN Interface, abstracting from CAN drivers and transceivers, see [3, SWS CAN Interface]
CDD	Complex Driver, any software that interfaces directly with AUTOSAR BSW modules, but is not defined by AUTOSAR, see [4, CDD Design And Integration Guideline] and [5, TPS ECU Configuration]
ComM	Communication Manager, controlling the bus communication, see [6, SWS Communication Manager]
DA	Destination Address, the address of the receiving CA of a PG. Part of the 29 bit CAN identifier of SAE J1939 PGs.
DP	Data Page, the most significant bit (MSB) of the 18 bit PGN
EcuM	ECU State Manager, mainly responsible for startup and shutdown of the ECU, see [7, SWS ECU State Manager]
EDP	Extended Data Page, the second bit (after MSB) of the 18 bit PGN
J1939Dcm	SAE J1939 Diagnostic Communication Manager, implementing the SAE J1939 Diagnostics, see [8, SWS SAE J1939 Diagnostic Communication Manager]
J1939Nm	SAE J1939 Network Management module, implementing the SAE J1939 Address Claiming, see [9, SWS SAE J1939 Network Management]
J1939Rm	SAE J1939 Request Manager, this BSW module, implementing the SAE J1939 Request Handling
J1939Tp	SAE J1939 Transport Layer, implementing the SAE J1939 Transport Protocol, see [10, SWS SAE J1939 Transport Layer]
NACK	SAE J1939 Acknowledgement PG (ACKM) with control byte set to 1, negative acknowledgement

▽

△

Acronym/ Abbreviation	Description
NAME	The 64 bit NAME of a CA used in the AddressClaimed PG
PDU1	SAE J1939 PDU Type 1, this kind of PG can be sent to a specific destination address
PDU2	SAE J1939 PDU Type 2, this kind of PG can only be sent as broadcast
PDUF	PDU Format, the middle byte of the 18 bit PGN which identifies the PG and determines the layout (PDU1/PDU2) of the PGN
PduR	PDU Router, establishes communication between BSW modules, see [11, SWS PDU Router]
PDUS	PDU Specific, the lower byte of the 18 bit PGN which further identifies PDU2 PGs which cannot have a destination address
PG	Parameter Group
PGN	Parameter Group Number
RQST	SAE J1939 Request PG (PGN = 0x0EA00)
RQST2	SAE J1939 Request2 PG (PGN = 0x0C900)
SA	Source Address, the address of the transmitting CA of a PG. Part of the 29 bit CAN identifier of SAE J1939 PGs.
SchM	Basic Software Schedule Manager, part of the RTE
SNA	Signal Not Available, all bits set to 1 in SAE J1939 PGs/Parameters
SPN	Suspect Parameter Number, unique identifier of an SAE J1939 Parameter
XFER	SAE J1939 Transfer PG (PGN = 0x0CA00)

Table 2.1: Acronyms and Abbreviations

The table below contains all document-independant acronyms or abbreviations that are defined in the AUTOSAR glossary [1].

Acronym/ Abbreviation	Description
BSW	Basic Software
COM	Communication
DET	Default Error Tracer, supports development and runtime error reporting, see [12, Specification of Default Error Tracer]
I-PDU	Interaction Layer Protocol Data Unit, see: I-PDU
PDU	Protocol Data Unit

▽

△

Acronym/ Abbreviation	Description
RTE	Run-Time Environment, provides the infrastructure services that enable communication to occur between AUTOSAR SWCs as well as acting as the means by which AUTOSAR SWCs access basic software modules including the OS and communication service. This includes the RTE's responsibility for the realization of the scheduling of the SWCs. See [13, Specification of RTE Software]
SAE	Society of Automotive Engineers, see: `SAE J1939`
SWC/SW-C	`Software Component`

Table 2.2: Acronyms and Abbreviations used from AUTOSAR Glossary

Missing Citations

The following entries in Table 2.2 lack citations:

COM Communication module, handling signal based communication and interface to the RTE, see [14, SWS Communication]

2.2 Definition of Terms

The table below contains all definitions of terms that are used in this document.

Glossary Term	Explanation
Basic Software module (BSW)	A module of the AUTOSAR stack below the RTE
Extended Identifier Bytes	These bytes represent multiplexor values in a multiplexed message which is requested via RQST2
Meta Data	Meta data transferred alongside a PDU, consisting of a set of `meta data items`
Meta Data Item	A single item of `meta data` of defined type and size
Parameter	A parameter is a signal of the `SAE J1939` application layer. The Parameter definition includes defined scale, limits, offset, and physical unit. Parameters are uniquely identified by the SPN.
Parameter Group	A Parameter Group is a message of the `SAE J1939` application layer. Each Parameter Group contains several `Parameters` (signals), and is uniquely identified by the PGN.

▽

△

Glossary Term	Explanation
Parameter Group Number	Unique identifier (18 bits: EDP, DP, PDUF, PDUS) of an SAE J1939 Parameter Group that is contained in the payload of many SAE J1939 protocol messages and in the 29bit CAN identifier of SAE J1939 PGs.
SAE J1939	Serial control and communications standard for heavy duty vehicle networks created by the SAE, see [15, SAE J1939]
SAE J1939-21	Data link layer for CAN 2.0 created by the SAE, see [16, SAE J1939-21]
SAE J1939-73	Diagnostics application layer created by the SAE, see [17, SAE J1939-73]
SAE J1939-81	Network management layer created by the SAE, see [18, SAE J1939-81]
SAE J1939 Address Claiming	The SAE J1939 address claiming procedure is the core feature of the network management of SAE J1939 defined in the standard document SAE J1939-81. Address claiming assigns a temporary 8-bit identifier to each CAs connected to an SAE J1939 network, based on the NAME of the CAs. Within each network, the 8-bit identifier is required to be unique. The 8-bit identifier is used as SA and DA of Parameter Groups (messages) transferred via the SAE J1939 network. The address claiming procedure is based on the exchange of AddressClaimed messages.
SAE J1939 Diagnostics	The SAE J1939 diagnostic layer is defined in the standard document SAE J1939-73. The SAE J1939 diagnostics is functionally similar to the UDS diagnostics, and has recently been extended to support OBD for emission related values.
SAE J1939 Request Handling	The SAE J1939 request handling allows a CA to request the transmission of a PG with transmission mode “On Request”. The request handling is defined in the network layer standard document SAE J1939-21.
SAE J1939 Transport Protocol	The SAE J1939 transport protocol is used for the segmented transmission of PGs with more than 8 bytes of data. The transport protocol is defined in the network layer standard document SAE J1939-21.

Table 2.3: Definition of terms in the scope of this Document

3 Related Documentation

3.1 Input Documents & Related Standards and Norms

[1] Glossary
AUTOSAR_FO_TR_Glossary

[2] Specification of Basic Software Mode Manager
AUTOSAR_CP_SWS_BSWModeManager

[3] Specification of CAN Interface
AUTOSAR_CP_SWS_CANInterface

[4] Complex Driver design and integration guideline
AUTOSAR_CP_EXP_CDDDesignAndIntegrationGuideline

[5] Specification of ECU Configuration
AUTOSAR_CP_TPS_ECUConfiguration

[6] Specification of Communication Manager
AUTOSAR_CP_SWS_COMManager

[7] Specification of ECU State Manager
AUTOSAR_CP_SWS_ECUStateManager

[8] Specification of a Diagnostic Communication Manager for SAE J1939
AUTOSAR_CP_SWS_SAEJ1939DiagnosticCommunicationManager

[9] Specification of Network Management for SAE J1939
AUTOSAR_CP_SWS_SAEJ1939NetworkManagement

[10] Specification of a Transport Layer for SAE J1939
AUTOSAR_CP_SWS_SAEJ1939TransportLayer

[11] Specification of PDU Router
AUTOSAR_CP_SWS_PDURouter

[12] Specification of Default Error Tracer
AUTOSAR_CP_SWS_DefaultErrorTracer

[13] Specification of RTE Software
AUTOSAR_CP_SWS_RTE

[14] Specification of Communication
AUTOSAR_CP_SWS_COM

[15] SAE J1939 – Serial Control and Communications Heavy Duty Vehicle Network

[16] SAE J1939-21 Data Link Layer

[17] SAE J1939-73 Application Layer – Diagnostics

[18] SAE J1939-81 Network Management

[19] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[20] Layered Software Architecture
AUTOSAR_CP_EXP_LayeredSoftwareArchitecture

[21] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[22] Specification of Communication Stack Types
AUTOSAR_CP_SWS_CommunicationStackTypes

[23] Specification of Standard Types
AUTOSAR_CP_SWS_StandardTypes

[24] System Template
AUTOSAR_CP_TPS_SystemTemplate

3.2 Related Specification

AUTOSAR provides a General Specification on Basic Software modules [19, SWS BSW General], which is also valid for the SAE J1939 Request Manager.

Thus, the specification [19, SWS BSW General] shall be considered as additional and required specification for the SAE J1939 Request Manager.

4 Constraints and Assumptions

4.1 Limitations

The SAE J1939 Request Manager only implements Request, Request2, and Acknowledgement PGs. It does not provide support for the Transfer PG.

4.2 Applicability to Automotive Domains

J1939 is developed by the SAE as a standard for heavy-duty on-highway, farming, and construction vehicles. It is not applicable to passenger cars or light trucks.

5 Dependencies to Other Modules

The [20, EXP Layered Software Architecture] shows an overview of the neighboring modules of the SAE J1939 Request Manager.

The SAE J1939 Request Manager (J1939Rm) has direct interfaces towards COM, the PDU Router (PduR), the SAE J1939 Network Management module (J1939Nm), the SAE J1939 Diagnostic Communication Manager (J1939Dcm), and the Default Error Tracer (DET), and also to application software components (SWCs) via the AUTOSAR Runtime Environment (RTE) and Complex Drivers (CDD). Besides these, there are also indirect dependencies towards the CAN Interface (CanIf) and the Communication Manager (ComM).

The SAE J1939 Request Manager includes header files of COM, J1939Nm, J1939Dcm, PduR, DET, CDDs, and the RTE.

5.1 File Structure

5.1.1 Code File Structure

For details, refer to [19, SWS BSW General] Chapter 5.1.6 *"Code file structure"*.

5.1.2 Header File Structure

Besides the files defined in [19, SWS BSW General] Chapter 5.1.7 *"Header file structure"*, the SAE J1939 Request Manager needs to include the files defined below.

[SWS_J1939Rm_00114]

Upstream requirements: SRS_BSW_00301

⌈J1939Rm shall include the header file Com.h if at least one J1939RmComUser is configured.⌋

[SWS_J1939Rm_00111]

Upstream requirements: SRS_BSW_00301

⌈J1939Rm shall include the header file J1939Nm.h if at least one J1939RmNmUser is configured.⌋

[SWS_J1939Rm_00112]

Upstream requirements: SRS_BSW_00301

⌈J1939Rm shall include the header file J1939Dcm.h if at least one J1939RmDcmUser is configured.⌋

[SWS_J1939Rm_00113]

Upstream requirements: SRS_BSW_00301

⌈J1939Rm shall include a header file named `<apiServicePrefix>_J1939Rm.h` for every configured `J1939RmCddUser`.⌋

Please note: `Complex driver` (CDD) APIs use the module prefix configured by the `apiServicePrefix` of the CDD's module description file.

6 Requirements Tracing

The following table references the requirements specified in [21, RS BSW General] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[SRS_BSW_00171]	Optional functionality of a Basic-SW component that is not required in the ECU shall be configurable at pre-compile-time	[SWS_J1939Rm_00074] [SWS_J1939Rm_00075] [SWS_J1939Rm_00076] [SWS_J1939Rm_00079] [SWS_J1939Rm_00080] [SWS_J1939Rm_00081] [SWS_J1939Rm_00082] [SWS_J1939Rm_00083] [SWS_J1939Rm_00084] [SWS_J1939Rm_00085] [SWS_J1939Rm_00086] [SWS_J1939Rm_00087]
[SRS_BSW_00301]	All AUTOSAR Basic Software Modules shall only import the necessary information	[SWS_J1939Rm_00111] [SWS_J1939Rm_00112] [SWS_J1939Rm_00113] [SWS_J1939Rm_00114]
[SRS_BSW_00350]	All AUTOSAR Basic Software Modules shall allow the enabling/ disabling of detection and reporting of development errors.	[SWS_J1939Rm_00011]
[SRS_BSW_00386]	The BSW shall specify the configuration and conditions for detecting an error	[SWS_J1939Rm_00011] [SWS_J1939Rm_00033] [SWS_J1939Rm_00040] [SWS_J1939Rm_00041] [SWS_J1939Rm_00067] [SWS_J1939Rm_00068] [SWS_J1939Rm_00069] [SWS_J1939Rm_00070] [SWS_J1939Rm_00071] [SWS_J1939Rm_00096]
[SRS_BSW_00478]	Timing limits of main functions	[SWS_J1939Rm_00043] [SWS_J1939Rm_00072]

Table 6.1: Requirements Tracing

7 Functional Specification

This chapter defines the behavior of the SAE J1939 Request Manager. The API of the module is defined in Chapter 8, while the configuration is defined in Chapter 10.

7.1 Overview

On one side, the SAE J1939 Request Manager is responsible for routing incoming RQST and RQST2 PGs to the correct destination, and to provide an infrastructure for sending responding ACKM PGs.

On the other side, the SAE J1939 Request Manager also provides an infrastructure to send RQST and RQST2 PGs, and to supervise timeout of the response(s), including but not limited to ACKM PGs.

The SAE J1939 Request Manager uses meta data items of type CAN_ID_32 of the received and transmitted ACKM and RQST PGs to access the source address, the destination address, and the priority which are encoded in the CAN ID.

[SWS_J1939Rm_00119] ⌈Meta data items of type CAN_ID_32 contain the source address in the fourth (least significant) byte.⌋

[SWS_J1939Rm_00120] ⌈Meta data items of type CAN_ID_32 contain the destination address in the third byte.⌋

[SWS_J1939Rm_00121] ⌈Meta data items of type CAN_ID_32 contain the priority in the bits 2-4 of the first (most significant) byte, where bit 0 is the least significant bit of a byte.⌋

7.2 Module Handling

This section contains description of auxiliary functionality of the SAE J1939 Request Manager.

7.2.1 Initialization

The SAE J1939 Request Manager is initialized via J1939Rm_Init, and de-initialized via J1939Rm_DeInit. Except for J1939Rm_Init, the API functions of the SAE J1939 Request Manager may only be called after the module has been properly initialized.

[SWS_J1939Rm_00012] ⌈A call to J1939Rm_Init initializes all internal variables and sets the SAE J1939 Request Manager to the initialized state.⌋

[SWS_J1939Rm_00013] ⌈A call to J1939Rm_DeInit sets the SAE J1939 Request Manager back to the uninitialized state.⌋

[SWS_J1939Rm_00011]

Upstream requirements: SRS_BSW_00350, SRS_BSW_00386

⌈When J1939Rm_Init is called in initialized state, the SAE J1939 Request Manager shall not re-initialize its internal variables. It shall instead call Det_ReportError with the error code J1939RM_E_REINIT if development error detection is enabled (see J1939RmDevErrorDetect).⌋

7.2.2 Timing Related Functionality

To be able to measure times, the SAE J1939 Request Manager is triggered cyclically via the J1939Rm_MainFunction.

[SWS_J1939Rm_00072]

Upstream requirements: SRS_BSW_00478

⌈The SAE J1939 Request Manager shall use the J1939Rm_MainFunction for timing related purposes.⌋

7.3 Communication State Handling

In general, request handling is only active when the ECU is online (see SAE J1939 Network Management for details). The exceptions to this rule are received and transmitted requests for the AddressClaimed PG, which must be possible in all cases. The SAE J1939 Request Manager provides an API that is used by the BSW Mode Manager (BswM) to notify the J1939 communication state.

[SWS_J1939Rm_00073] ⌈During initialization via J1939Rm_Init, the SAE J1939 Request Manager assumes the offline state for all nodes on all channels.⌋

[SWS_J1939Rm_00014] ⌈A call to J1939Rm_SetState sets the state of a node's channel to online or offline.⌋

[SWS_J1939Rm_00015] ⌈In the offline state, the SAE J1939 Request Manager only processes requests for the AddressClaimed PG, while timeout supervision and acknowledgement handling are completely disabled.⌋

7.4 Reception of Requests

The SAE J1939 Request Manager receives request PGs (RQST and RQST2) via J1939Rm_RxIndication from the CAN Interface. The SAE J1939 Request

Manager shall use the meta data item type CAN_ID_32 to be able to identify the sender, the destination address, and the priority of the request.

[SWS_J1939Rm_00122] ⌈The SAE J1939 Request Manager shall use a meta data item of type CAN_ID_32 to determine the source address, destination address, and priority of received request PGs.⌋

[SWS_J1939Rm_00007] ⌈The SAE J1939 Request Manager shall only accept requests addressed to the whole network (global DA), or to one of the configured addresses of the ECU (see J1939RmNmNodeRef).⌋

Requests for the AddressClaimed PG (AC, PGN = 0x0EE00) always go to the SAE J1939 Network Management module. Requests for the DMx PGs (DM01 to DM60) always go to the SAE J1939 Diagnostic Communication Manager, the destination of these and other PGNs is configured via J1939RmUserRequestPGN.

Besides forwarding to the SAE J1939 Network Management module, the SAE J1939 Diagnostic Communication Manager, and CDDs, the SAE J1939 Request Manager can also forward requests to SWCs, and trigger COM to send requested PGs.

7.4.1 Request Forwarding

Forwarding to other BSW modules is done via the generic callout function User_RequestIndication. Forwarding to SWC uses a dedicated service port function with the same signature as the User_RequestIndication.

[SWS_J1939Rm_00002] ⌈When J1939Rm_RxIndication is called by the PDU Router to indicate reception of a request, and the requested PGN is configured via J1939RmUserRequestPGN to trigger either the SAE J1939 Diagnostic Communication Manager or a CDD, the SAE J1939 Request Manager shall call the corresponding User_RequestIndication.⌋

[SWS_J1939Rm_00116] ⌈When J1939Rm_RxIndication is called by the PDU Router to indicate reception of a request, and the requested PGN is AddressClaimed (AC, 0x0EE00), the SAE J1939 Request Manager shall call J1939Nm_RequestIndication.⌋

[SWS_J1939Rm_00003] ⌈When J1939Rm_RxIndication is called by the PDU Router to indicate reception of a request, and the requested PGN is configured via J1939RmUserRequestPGN to be forwarded to the RTE, the SAE J1939 Request Manager shall call the corresponding service port function.⌋

7.4.2 Request Handling via COM

If COM is configured as destination for the request of a certain PGN, the SAE J1939 Request Manager will prepare the meta data, and request COM to send the PDU with the meta data provided via Com_TriggerIPDUSendWithMetaData. This sequence is shown in Figure 9.3.

[SWS_J1939Rm_00115] ⌈When J1939Rm_RxIndication is called by the PDU Router to indicate reception of a request, and the requested PGN is configured via J1939RmComIPduPGN to be handled via COM, and when the extended identifier bytes of an RQST2 match the multiplexor values of a multiplexed message, the SAE J1939 Request Manager shall prepare the meta data from the given information and provide it to COM via Com_TriggerIPDUSendWithMetaData together with the PduId of the transmitted COM I-PDU referenced by J1939RmComIPduRef.⌋

7.4.3 Request of Unknown PGNs

The SAE J1939 Request Manager shall respond to requests for unknown PGNs with a NACK, but only when the request was sent to a specific destination address.

[SWS_J1939Rm_00008] ⌈When J1939Rm_RxIndication is called by the PDU Router to indicate reception of a request, and the requested PGN or the requested extended identifier bytes are not configured, and the destination address is not the broadcast address, the SAE J1939 Request Manager shall call PduR_J1939RmTransmit to send a negative acknowledgement (NACK).⌋

7.5 Transmission of Acknowledgements

For unknown PGNs, the SAE J1939 Request Manager transmits a negative acknowledgement by itself (see section 7.4.3 above). Modules that receive requests from the SAE J1939 Request Manager may use the API J1939Rm_SendAck to transmit the acknowledgement variants defined by the J1939 standard (see section 5.4.4 in [16, SAE J1939-21]).

The Acknowledgement PG is supposed to have a fixed destination address (0xFF), configured via CanIfTxPduCanId in the CAN Interface. The SAE J1939 Request Manager shall use the meta data item type CAN_ID_32 so that it can modify the priority and source address.

[SWS_J1939Rm_00009] ⌈When a BSW module, a CDD, or an SWC (via service port and RTE) calls J1939Rm_SendAck, the SAE J1939 Request Manager shall call PduR_J1939RmTransmit to send the required acknowledgement.⌋

[SWS_J1939Rm_00123] ⌈The SAE J1939 Request Manager shall use a meta data item of type CAN_ID_32 to provide the source address and priority of transmitted Acknowledgement PGs.⌋

There is only one I-PDU available to send Acknowledgement PGs. Still, it must be ensured, that no Acknowledgement PG is lost, even when a new transmission is initiated while this I-PDU is already occupied by another transmission. To achieve this, the SAE J1939 Request Manager needs to queue Acknowledgement PGs.

[SWS_J1939Rm_00018] ⌈Transmission requests for the Acknowledgement PG shall be queued when a previous transmission of this PG is still pending. The size of this queue is determined by J1939RmAckQueueSize.⌋

[SWS_J1939Rm_00019] ⌈The SAE J1939 Request Manager shall use the J1939Rm_TxConfirmation with result E_OK of the associated I-PDU to trigger transmission of pending Acknowledgement PGs.⌋

[SWS_J1939Rm_00020] ⌈If the J1939Rm_TxConfirmation is called with result E_NOT_OK, the SAE J1939 Request Manager shall flush the Acknowledgement PG queue.⌋

The acknowledgement type (Control byte), the extended identifier bytes, and the Address parameter of the Acknowledgement PG are set according to the arguments of the J1939Rm_SendAck function. The destination address is always the global address, as defined in SAE J1939-21.

[SWS_J1939Rm_00126] ⌈When an acknowledgement is sent, it shall also be handled internally as if it was received via J1939Rm_RxIndication on the same channel.⌋

7.6 Transmission of Requests

As stated in section 7.1, the SAE J1939 Request Manager also supports transmission of requests, reception of responding acknowledgements, and timeout supervision for the responses.

To trigger the transmission of a request, the SAE J1939 Request Manager provides the API J1939Rm_SendRequest.

The SAE J1939 Request Manager shall use the meta data item type CAN_ID_32 to be able to set the priority and the source and destination address freely. The CAN Interface must be configured such that the PDUF and data page bits are fixed, while the remaining bits of the CAN ID are variable.

[SWS_J1939Rm_00016] ⌈When a BSW module, a CDD, or an SWC (via service port and RTE) calls J1939Rm_SendRequest, the SAE J1939 Request Manager shall call PduR_J1939RmTransmit to send the request.⌋

[SWS_J1939Rm_00117] ⌈When no extended identifier bytes are provided with J1939Rm_SendRequest, J1939Rm shall send an RQST PG. When one or more extended identifier bytes are provided, an RQST2 PG shall be sent.⌋

[SWS_J1939Rm_00124] ⌈The SAE J1939 Request Manager shall use a meta data item of type CAN_ID_32 to provide the source address, destination address, and priority of transmitted Request and Request2 PGs.⌋

There is only one I-PDU available to send Request PGs, and one for Request2 PGs. Still, it must be ensured that no request PG is lost, even when a new transmission is initiated while this I-PDU is already occupied by another transmission. To achieve this, the SAE J1939 Request Manager needs to queue request PGs.

[SWS_J1939Rm_00021] ⌈Transmission requests for the Request PG shall be queued when a previous transmission of this PG is still pending. The size of this queue is determined by J1939RmRequestQueueSize.⌋

[SWS_J1939Rm_00118] ⌈Transmission requests for the Request2 PG shall be queued when a previous transmission of this PG is still pending. The size of this queue is determined by J1939RmRequest2QueueSize.⌋

[SWS_J1939Rm_00022] ⌈The SAE J1939 Request Manager shall use the J1939Rm_TxConfirmation with result E_OK of the associated I-PDU to trigger transmission of pending Request and Request2 PGs.⌋

[SWS_J1939Rm_00023] ⌈If the J1939Rm_TxConfirmation is called with result E_NOT_OK, the SAE J1939 Request Manager shall flush the corresponding request PG queue.⌋

To be able to do timeout supervision, the SAE J1939 Request Manager needs to remember the initiator, the destination address, extended identifier bytes, and the PGN of the request.

[SWS_J1939Rm_00024] ⌈When J1939Rm_SendRequest is called with the parameter checkTimeout set to TRUE and a destination address that is not the broadcast address (0xFF), and timeout handling is enabled for the caller via J1939RmUserTimeoutSupervision: The SAE J1939 Request Manager shall store (separately for each node) the calling module's user ID, the PGN, extended identifier bytes, the source address, and the destination address of the request.⌋

Finally, requests to the global address must also be handled internally as described in section 7.4.

[SWS_J1939Rm_00025] ⌈When a request is sent with the global destination address, it shall also be handled internally as if it was received via J1939Rm_RxIndication on the same channel.⌋

7.7 Reception of Acknowledgements

The SAE J1939 Request Manager receives Acknowledgement PGs (ACKM) via J1939Rm_RxIndication from the CAN Interface. The SAE J1939 Request Manager shall use the meta data item type CAN_ID_32 to be able to identify the priority and the sender of the acknowledgement.

[SWS_J1939Rm_00125] ⌈The SAE J1939 Request Manager shall use a meta data item of type CAN_ID_32 to determine the source address and priority of received Acknowledgement PGs.⌋

[SWS_J1939Rm_00026] ⌈The SAE J1939 Request Manager shall only accept acknowledgements where the AddressAcknowledged is set to one of the configured addresses of the ECU (see J1939RmNmNodeRef).⌋

The scheduling of received Acknowledgement PGs is configured similarly to the Request PG, see section 7.4.1, but the destinations are restricted to CDD and Application, because the J1939Nm and the J1939Dcm currently do not need to request any information from other ECUs.

[SWS_J1939Rm_00066] ⌈When J1939Rm_RxIndication is called by the PDU Router to indicate reception of an acknowledgement which matches a pending request (acknowledged PGN, source address, acknowledged address), the SAE J1939 Request Manager shall call the User_AckIndication or the service port function corresponding to the stored user ID.⌋

[SWS_J1939Rm_00027] ⌈When J1939Rm_RxIndication is called by the PDU Router to indicate reception of an acknowledgement which does not match a pending request, and the acknowledged PGN is configured via J1939RmUserAckPGN to trigger a CDD, the SAE J1939 Request Manager shall call the corresponding User_AckIndication.⌋

[SWS_J1939Rm_00028] ⌈When J1939Rm_RxIndication is called by the PDU Router to indicate reception of an acknowledgement which does not match a pending request, and the acknowledged PGN is configured via J1939RmUserAckPGN to be forwarded to the RTE, the SAE J1939 Request Manager shall call the corresponding service port function.⌋

7.8 Timeout Supervision

The SAE J1939 specification SAE J1939-21 defines a maximum delay of 200ms for the answer to a request. This delay is not supervised by the SAE J1939 Request Manager. On the other hand, the timeout of 1.25s for the reception of the answer to a request will be supervised by the SAE J1939 Request Manager, if configured accordingly via J1939RmUserTimeoutSupervision. In that case, when the request

is transmitted, the timer is started and the request data is stored as described in [SWS_J1939Rm_00024].

[SWS_J1939Rm_00017] ⌈If timeout supervision is enabled for the caller of J1939Rm_SendRequest via J1939RmUserTimeoutSupervision, and the parameter checkTimeout is TRUE, and the destination address is not the broadcast address (0xFF): The SAE J1939 Request Manager shall start timeout supervision.⌋

[SWS_J1939Rm_00029] ⌈When an acknowledgement matching the request is received, when J1939Rm_CheckReceivedComIPdu is called which matches the request, or when a CDD or an application SWC calls J1939Rm_CancelRequestTimeout, the timeout supervision of the request is stopped.⌋

[SWS_J1939Rm_00030] ⌈If the timeout supervision for a request reaches 1.25s, the SAE J1939 Request Manager shall call the User_RequestTimeoutIndication corresponding to the userId parameter of the initial J1939Rm_SendRequest.⌋

7.9 Routing of Requests and Acknowledgements

Depending on the configuration of J1939NmSharedAddressSpace and J1939NmExternalNodeGatewayedChannelRef referring to J1939NmChannels that reference the same ComMChannels as the J1939RmChannels, the Request, Request2, and Acknowledgement PGs need to be routed from one J1939RmChannel to another.

[SWS_J1939Rm_00127] ⌈If J1939RmGatewaySupport is enabled, and a J1939RmChannel is linked to another J1939RmChannel via a J1939NmSharedAddressSpace: All Request, Request2, and Acknowledgement PGs that are received on the first J1939RmChannel shall be forwarded to the second J1939RmChannel.⌋

Note: The complete path between two J1939RmChannels linked via a J1939NmSharedAddressSpace is:

J1939RmChannel → J1939RmComMNetworkHandleRef → ComMChannel ← J1939NmComMNetworkHandleRef ← J1939NmChannel ← J1939NmSharedChannelRef ← J1939NmSharedAddressSpace → J1939NmSharedChannelRef → J1939NmChannel → J1939NmComMNetworkHandleRef → ComMChannel ← J1939RmComMNetworkHandleRef ← J1939RmChannel

[SWS_J1939Rm_00128] ⌈If J1939RmGatewaySupport is enabled, and a J1939RmChannel is referenced by another J1939RmChannel via a J1939NmExternalNodeGatewayedChannelRef: All Request, Request2, and Acknowledgement PGs that are received on the first J1939RmChannel shall be forwarded to the second J1939RmChannel.⌋

Note: The complete path between two J1939RmChannels linked via a J1939NmExternalNodeGatewayedChannelRef is:

J1939RmChannel → J1939RmComMNetworkHandleRef → ComMChannel ← J1939NmComMNetworkHandleRef ← J1939NmChannel ← J1939NmExternalNodeGatewayedChannelRef ← J1939NmExternalNode → J1939NmExternalNodeChannelRef → J1939NmChannel → J1939NmComMNetworkHandleRef → ComMChannel ← J1939RmComMNetworkHandleRef ← J1939RmChannel

[SWS_J1939Rm_00129] ⌈Request and Request2 PGs shall only be forwarded if the destination address of the PG is the global address (0xFF) or a destination address that does not correspond to any J1939NmNodePreferredAddress referenced by a J1939RmNode that references the J1939RmChannel on which the PG was received.⌋

7.10 Error Classification

Chapter [19, General Specification of Basic Software Modules] Chapter 7.2 *"Error Handling"* describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of following error types which may occur in BSW modules.

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

7.10.1 Development Errors

[SWS_J1939Rm_00031] Definition of development errors in module J1939Rm ⌈

Type of error	*Related error code*	*Error value*
An API was called while the module was uninitialized	J1939RM_E_UNINIT	0x01
The Init API was called twice	J1939RM_E_REINIT	0x02
J1939Rm_Init was called with an invalid configuration pointer	J1939RM_E_INIT_FAILED	0x03
An API service was called with a NULL pointer	J1939RM_E_PARAM_POINTER	0x10
An API service was called with a wrong ID	J1939RM_E_INVALID_PDU_SDU_ID	0x11
An API service was called with wrong network handle	J1939RM_E_INVALID_NETWORK_ID	0x12
The API J1939Rm_SetState was called with a wrong state	J1939RM_E_INVALID_STATE	0x13
An API was called with an illegal user ID	J1939RM_E_INVALID_USER	0x14
An API was called with an unknown or illegal PGN	J1939RM_E_INVALID_PGN	0x15
An API was called with an illegal priority	J1939RM_E_INVALID_PRIO	0x16
An API was called with an illegal node address	J1939RM_E_INVALID_ADDRESS	0x17

▽

△

Type of error	*Related error code*	*Error value*
An API was called with an illegal Boolean option	J1939RM_E_INVALID_OPTION	0x18
An API was called with an illegal AckCode	J1939RM_E_INVALID_ACK_CODE	0x19
An API was called with an illegal node ID	J1939RM_E_INVALID_NODE_ID	0x1a
An API was called with invalid extended identifier bytes	J1939RM_E_INVALID_EXTID_INFO	0x1b

⌋

7.10.2 Runtime Errors

There are no runtime errors.

7.10.3 Production Errors

There are no production errors.

8 API Specification

8.1 API Parameter Checking

The SAE J1939 Request Manager performs parameter checks for all called APIs. It reports the development error J1939RM_E_PARAM_POINTER when a call provides a NULL pointer, J1939RM_E_INVALID_PDU_SDU_ID when a check of a PDU ID fails, J1939RM_E_INVALID_NETWORK_ID when a check of a network handle fails, and J1939RM_E_INVALID_NODE_ID when a check of a node handle fails.

J1939RM_E_PARAM_POINTER is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00212].

[SWS_J1939Rm_00033]

Upstream requirements: SRS_BSW_00386

⌈If development error detection is enabled via J1939RmDevErrorDetect, the SAE J1939 Request Manager shall check PduIdType parameters (PDU IDs) of its API functions against the configured IDs, and shall report the development error J1939RM_E_INVALID_PDU_SDU_ID when an unknown ID is provided by the call.⌋

[SWS_J1939Rm_00041]

Upstream requirements: SRS_BSW_00386

⌈If development error detection is enabled via J1939RmDevErrorDetect, the SAE J1939 Request Manager shall check NetworkHandleType parameters (network handles) of its API functions against the referenced network handles of ComM, and shall report the development error J1939RM_E_INVALID_NETWORK_ID when an unknown handle is provided by the call.⌋

[SWS_J1939Rm_00096]

Upstream requirements: SRS_BSW_00386

⌈If development error detection is enabled via J1939RmDevErrorDetect, the SAE J1939 Request Manager shall check node handle parameters of its API functions against the node handles of J1939Nm referenced via J1939RmNmNodeRef, and shall report the development error J1939RM_E_INVALID_NODE_ID when an unknown handle is provided by the call.⌋

8.2 Imported Types

This section lists all types used by the SAE J1939 Request Manager together with the defining module.

[SWS_J1939Rm_00035] Definition of imported datatypes of module J1939Rm ⌈

Module	*Header File*	*Imported Type*
Comtype	ComStack_Types.h	NetworkHandleType
	ComStack_Types.h	PduIdType
	ComStack_Types.h	PduInfoType
	ComStack_Types.h	PduLengthType
Std	Std_Types.h	Std_ReturnType

⌋

The types that are declared in `ComStack_Types.h` (Module `Comtype`) are defined in [22, SWS Communication Stack Types], while the types declared in `Std_Types.h` (Module `Std`) are defined in [23, SWS Standard Types].

8.3 Type Definitions

8.3.1 J1939Rm_ConfigType

[SWS_J1939Rm_00036] Definition of datatype J1939Rm_ConfigType ⌈

Name	J1939Rm_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	–
Description	This is the base type for the configuration of the J1939 Request Manager. A pointer to an instance of this structure will be used in the initialization of the J1939 Request Manager. The content of this structure is defined in chapter 10 Configuration specification.	
Available via	J1939Rm.h	

⌋

8.3.2 J1939Rm_StateType

[SWS_J1939Rm_00049] Definition of datatype J1939Rm_StateType ⌈

Name	J1939Rm_StateType		
Kind	Enumeration		
Range	J1939RM_STATE_OFFLINE	0x00	Only Request for AC
	J1939RM_STATE_ONLINE	0x01	Normal communication
Description	This type represents the communication state of the J1939 Request Manager.		
Available via	J1939Rm.h		

⌋

8.4 Function Definitions

This chapter contains all APIs provided by the module.

8.4.1 Generally Available Functions

This chapter contains all APIs which are provided by the module via `J1939Rm.h`.

8.4.1.1 J1939Rm_Init

[SWS_J1939Rm_00037] Definition of API function J1939Rm_Init ⌈

Service Name	J1939Rm_Init	
Syntax	`void J1939Rm_Init ( const J1939Rm_ConfigType* configPtr )`	
Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	configPtr	Pointer to selected configuration structure
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This function initializes the J1939 Request Manager.	
Available via	J1939Rm.h	

⌋

See section 7.2.1 for details.

See section 8.1 for parameter checks.

`J1939RM_E_INIT_FAILED` is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00050].

8.4.1.2 J1939Rm_DeInit

[SWS_J1939Rm_00038] Definition of API function J1939Rm_DeInit ⌈

Service Name	J1939Rm_DeInit
Syntax	`void J1939Rm_DeInit ( void )`
Service ID [hex]	0x02
Sync/Async	Synchronous
Reentrancy	Non Reentrant

▽

△

Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	This function resets the J1939 Request Manager to the uninitialized state.
Available via	J1939Rm.h

⌋

See section 7.2.1 for details.

8.4.1.3 J1939Rm_SetState

[SWS_J1939Rm_00048] Definition of API function J1939Rm_SetState ⌈

Service Name	J1939Rm_SetState	
Syntax	`Std_ReturnType J1939Rm_SetState ( NetworkHandleType channel, uint8 node, J1939Rm_StateType newState )`	
Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	channel	Channel for which the state shall be changed.
	node	Node for which the state shall be changed.
	newState	New state the J1939Rm shall enter, see definition of J1939Rm_StateType for available states.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: New communication state was set `E_NOT_OK`: Communication state was not changed due to wrong value in NewState or wrong initialization state of the module.
Description	Changes the communication state of J1939Rm to offline (only Request for AC supported) or online.	
Available via	J1939Rm.h	

⌋

[SWS_J1939Rm_00040]

Upstream requirements: SRS_BSW_00386

⌈The `SAE J1939 Request Manager` shall reject the state change by returning `E_NOT_OK` when the `newState` is not in the valid range. If development error detection is enabled via `J1939RmDevErrorDetect`, the development error `J1939RM_E_INVALID_STATE` shall be reported.⌋

`J1939RM_E_UNINIT` is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00243].

See Chapter 8.1 for parameter checks.

8.4.1.4 J1939Rm_SendRequest

[SWS_J1939Rm_00054] Definition of API function J1939Rm_SendRequest ⌈

Service Name	J1939Rm_SendRequest	
Syntax	`Std_ReturnType J1939Rm_SendRequest ( uint8 userId, NetworkHandleType channel, uint32 requestedPgn, const J1939Rm_ExtIdInfoType* extIdInfo, uint8 destAddress, uint8 priority, boolean checkTimeout )`	
Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	userId	Identification of the calling module.
	channel	Channel on which the request shall be sent.
	requestedPgn	PGN of the requested PG.
	extIdInfo	Extended identifier bytes. J1939RM_EXTID_NONE is assumed if a NULL pointer is provided.
	destAddress	Address of the destination node or 0xFF for broadcast.
	priority	Priority of the Request PG.
	checkTimeout	TRUE: Timeout supervision will be performed FALSE: No timeout supervision will be started
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Transmission request is accepted `E_NOT_OK`: Transmission request is not accepted
Description	Requests transmission of a Request or Request2 PG.	
Available via	J1939Rm.h	

⌋

[SWS_J1939Rm_00074]

Upstream requirements: SRS_BSW_00171

⌈The `J1939Rm_SendRequest` API function shall only be available if `J1939RmUserSendRequest` is set for at least one user.⌋

See section 7.6 for details.

[SWS_J1939Rm_00067]

Upstream requirements: SRS_BSW_00386

⌈The `SAE J1939 Request Manager` shall reject transmission of a request by returning `E_NOT_OK` when the `requestedPgn`, the `extIdType` element within the `extIdInfo`, the `destAddress`, or the `priority` are not in the valid range, or when the `userId` is not one of the configured user IDs (see `J1939RmUserId`), or when `checkTimeout` is true but timeout handling is disabled for the calling module (see `J1939RmUserTimeoutSupervision`). If development error detection is enabled via `J1939RmDevErrorDetect`, the corresponding development error shall be

reported: J1939RM_E_INVALID_USER for userId, J1939RM_E_INVALID_EXTID_INFO for extIdInfo, J1939RM_E_INVALID_PGN for requestedPgn, J1939RM_E_INVALID_PRIO for priority, J1939RM_E_INVALID_ADDRESS for destAddress, and J1939RM_E_INVALID_OPTION for checkTimeout.⌋

[SWS_J1939Rm_00068]

Upstream requirements: SRS_BSW_00386

⌈The SAE J1939 Request Manager shall reject transmission of a request by returning E_NOT_OK when another request is pending and the request queue is full.⌋

J1939RM_E_UNINIT is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00243].

See Chapter 8.1 for further parameter checks.

8.4.1.5 J1939Rm_CancelRequestTimeout

[SWS_J1939Rm_00055] Definition of API function J1939Rm_CancelRequestTimeout ⌈

Service Name	J1939Rm_CancelRequestTimeout	
Syntax	`Std_ReturnType J1939Rm_CancelRequestTimeout ( uint8 userId, NetworkHandleType channel, uint32 requestedPgn, const J1939Rm_ExtIdInfoType* extIdInfo, uint8 destAddress )`	
Service ID [hex]	0x08	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	userId	Identification of the calling module.
	channel	Channel on which the request was sent.
	requestedPgn	PGN of the requested PG.
	extIdInfo	Extended identifier bytes. J1939RM_EXTID_NONE is assumed if a NULL pointer is provided.
	destAddress	Address of the destination node.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Cancellation of request timeout was successful E_NOT_OK: Cancellation of request timeout was not successful
Description	Cancels timeout monitoring of a request. If the request is not active, or timeout monitoring was not requested, this call has no effect.	
Available via	J1939Rm.h	

⌋

[SWS_J1939Rm_00075]

Upstream requirements: SRS_BSW_00171

⌈The `J1939Rm_CancelRequestTimeout` API function shall only be available if `J1939RmUserTimeoutSupervision` is set for at least one user.⌋

See section 7.8 for details.

[SWS_J1939Rm_00069]

Upstream requirements: SRS_BSW_00386

⌈The `SAE J1939 Request Manager` shall ignore the timeout cancellation request when the `requestedPgn`, the `extIdType` element within the `extIdInfo`, or the `destAddress` are not in the valid range, or when the `userId` is not one of the configured user IDs (see `J1939RmUserId`), or if no suitable entry can be found in the list of pending requests. If development error detection is enabled via `J1939RmDevErrorDetect`, the corresponding development error shall be reported: `J1939RM_E_INVALID_USER` for `userId`, `J1939RM_E_INVALID_PGN` for `requestedPgn`, `J1939RM_E_INVALID_EXTID_INFO` for `extIdInfo`, and `J1939RM_E_INVALID_ADDRESS` for `destAddress`.⌋

`J1939RM_E_UNINIT` is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00243].

See Chapter 8.1 for further parameter checks.

8.4.1.6 J1939Rm_SendAck

[SWS_J1939Rm_00056] Definition of API function J1939Rm_SendAck ⌈

Service Name	J1939Rm_SendAck	
Syntax	`Std_ReturnType J1939Rm_SendAck ( uint8 userId, NetworkHandleType channel, uint32 ackPgn, const J1939Rm_ExtIdInfoType* extIdInfo, J1939Rm_AckCode ackCode, uint8 ackAddress, uint8 priority, boolean broadcast )`	
Service ID [hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	userId	Identification of the calling module.
	channel	Channel on which the acknowledgement shall be sent.
	ackPgn	Acknowledged PGN.

▽

△

	extIdInfo	Extended identifier bytes. J1939RM_EXTID_NONE is assumed if a NULL pointer is provided.
	ackCode	Type of acknowledgement, see definition of J1939Rm_AckCode for available codes.
	ackAddress	Address of the node that sent the request.
	priority	Priority of the Acknowledgement PG.
	broadcast	Indicates whether the ACKM is a response to a broadcast request.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Transmission request is accepted E_NOT_OK: Transmission request is not accepted
Description	Requests transmission of an Acknowledgement PG.	
Available via	J1939Rm.h	

⌋

[SWS_J1939Rm_00076]

Upstream requirements: SRS_BSW_00171

⌈The J1939Rm_SendAck API function shall only be available if J1939RmUserSendAck is set for at least one user.⌋

See section 7.5 for details.

[SWS_J1939Rm_00070]

Upstream requirements: SRS_BSW_00386

⌈The SAE J1939 Request Manager shall reject transmission of an acknowledgement by returning E_NOT_OK when the ackPgn, the extIdType element within the extIdInfo, the ackAddress, or the priority are not in the valid range, or when the userId is not one of the configured user IDs (see J1939RmUserId). If development error detection is enabled via J1939RmDevErrorDetect, the corresponding development error shall be reported: J1939RM_E_INVALID_USER for userId, J1939RM_E_INVALID_EXTID_INFO for extIdInfo, J1939RM_E_INVALID_PGN for ackPgn, J1939RM_E_INVALID_ACK_CODE for ackCode, J1939RM_E_INVALID_ADDRESS for ackAddress, and J1939RM_E_INVALID_PRIO for priority.⌋

[SWS_J1939Rm_00071]

Upstream requirements: SRS_BSW_00386

⌈The SAE J1939 Request Manager shall reject transmission of an acknowledgement by returning E_NOT_OK when another acknowledgement is pending and the acknowledgement queue is full.⌋

J1939RM_E_UNINIT is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00243].

See Chapter 8.1 for further parameter checks.

8.4.1.7 J1939Rm_RxIndication

[SWS_J1939Rm_00058] Definition of callback function J1939Rm_RxIndication ⌈

Service Name	J1939Rm_RxIndication	
Syntax	`void J1939Rm_RxIndication ( PduIdType RxPduId, const PduInfoType* PduInfoPtr )`	
Service ID [hex]	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	RxPduId	ID of the received PDU.
	PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indication of a received PDU from a lower layer communication interface module.	
Available via	J1939Rm.h	

⌋

[SWS_J1939Rm_00080]

Upstream requirements: SRS_BSW_00171

⌈The `J1939Rm_RxIndication` call back function shall only be available if `J1939Rm_UserAckIndication` or `J1939RmUserRequestIndication` is set for at least one user.⌋

See sections 7.4 and 7.7 for details.

`J1939RM_E_UNINIT` is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00243].

See Chapter 8.1 for parameter checks.

8.4.1.8 J1939Rm_TxConfirmation

[SWS_J1939Rm_00059] Definition of callback function J1939Rm_TxConfirmation ⌈

Service Name	J1939Rm_TxConfirmation
Syntax	`void J1939Rm_TxConfirmation ( PduIdType TxPduId, Std_ReturnType result )`
Service ID [hex]	0x40
Sync/Async	Synchronous

▽

△

Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	TxPduId	ID of the PDU that has been transmitted.
	result	`E_OK`: The PDU was transmitted successfully. `E_NOT_OK`: Transmission of the PDU failed.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.	
Available via	J1939Rm.h	

⌋

[SWS_J1939Rm_00081]

Upstream requirements: SRS_BSW_00171

⌈The `J1939Rm_TxConfirmation` call back function shall only be available if `J1939RmUserSendAck` or `J1939RmUserSendRequest` is set for at least one user.⌋

See sections 7.5 and 7.6 for details.

`J1939RM_E_UNINIT` is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00243].

See Chapter 8.1 for parameter checks.

8.4.2 Functions Provided for COM

This chapter contains all APIs which are provided to COM via the private module header (`J1939Rm_Com.h`).

8.4.2.1 J1939Rm_CheckReceivedComIPdu

[SWS_J1939Rm_00062] Definition of callback function J1939Rm_CheckReceivedComIPdu ⌈

Service Name	J1939Rm_CheckReceivedComIPdu	

Syntax	`boolean J1939Rm_CheckReceivedComIPdu (`
	`  PduIdType PduId,`
	`  const PduInfoType* PduInfoPtr`
	`)`	

Service ID [hex]	0x28	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	PduId	ID of the received ComIPdu.

▽

△

	PduInfoPtr	Length (SduLength) of the received ComIPdu and a pointer to the data of the ComIPdu (SduDataPtr).
Parameters (inout)	None	
Parameters (out)	None	
Return value	boolean	Shall be always true to ensure the ComIPdu is received.
Description	Reports a received ComIPdu. If this ComIPdu was requested via J1939Rm_SendRequest or the SendRequest service operation, a request timeout for this request is stopped.	
Available via	J1939Rm_Com.h	

⌋

[SWS_J1939Rm_00079]

Upstream requirements: SRS_BSW_00171

⌈The function `J1939Rm_CheckReceivedComIPdu` shall only be available if `J1939RmUserTimeoutSupervision` is set for at least one user.⌋

See section 7.8 for details.

`J1939RM_E_UNINIT` is expected to be reported as specified in [19, SWS BSW General] by [SWS_BSW_00243].

See Chapter 8.1 for parameter checks.

8.5 Scheduled Functions

This function is directly called by the `Basic Software Scheduler` (SchM).

8.5.1 J1939Rm_MainFunction

[SWS_J1939Rm_00042] Definition of scheduled function J1939Rm_MainFunction ⌈

Service Name	J1939Rm_MainFunction
Syntax	`void J1939Rm_MainFunction ( void )`
Service ID [hex]	0x04
Description	Main function of the J1939 Request Manager. Used for scheduling purposes and timeout supervision.
Available via	SchM_J1939Rm.h

⌋

[SWS_J1939Rm_00043]

Upstream requirements: SRS_BSW_00478

⌈The frequency of invocations of `J1939Rm_MainFunction` is determined by the configuration parameter `J1939RmMainFunctionPeriod`.⌋

8.6 Expected Interfaces

This section lists all interfaces required from other modules.

8.6.1 Mandatory Interfaces

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

[SWS_J1939Rm_00044] Definition of mandatory interfaces required by module J1939Rm ⌈

API Function	*Header File*	*Description*
PduR_J1939RmTransmit	PduR_J1939Rm.h	Requests transmission of a PDU.

⌋

8.6.2 Optional Interfaces

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

[SWS_J1939Rm_00045] Definition of optional interfaces requested by module J1939Rm ⌈

API Function	*Header File*	*Description*
Com_TriggerIPDUSendWithMetaData	Com.h	By a call to Com_TriggerIPDUSendWithMetaData the AUTOSAR COM module updates its internal metadata for the I-PDU with the given ID by copying the metadata from the given position and with respect to length of the globally configured Meta DataType of this I-PDU. Then the I-PDU is triggered for transmission.
Det_ReportError	Det.h	Service to report development errors.
J1939Dcm_RequestIndication	J1939Dcm.h	Indicates reception of a Request or Request2 PG.
J1939Nm_RequestIndication	J1939Nm.h	Indicates reception of a Request or Request2 PG.

⌋

[SWS_J1939Rm_00082]

Upstream requirements: SRS_BSW_00171

⌈The `Com_TriggerIPDUSendWithMetaData` function is only required if at least one J1939RmComUser is configured.⌋

[SWS_J1939Rm_00083]

Upstream requirements: SRS_BSW_00171

⌈The `J1939Dcm_RequestIndication` function is only required if at least one J1939RmDcmUser is configured.⌋

[SWS_J1939Rm_00084]

Upstream requirements: SRS_BSW_00171

⌈The `J1939Nm_RequestIndication` function is only required if at least one J1939RmNmUser is configured.⌋

8.6.3 Configurable Interfaces

This section lists all interfaces where the target function can be configured. The target function is usually a call-back function. The name of this kind of interface is not fixed because they are configurable.

8.6.3.1 <User>_RequestIndication

[SWS_J1939Rm_00063] Definition of configurable interface < User >_Request Indication ⌈

Service Name	< User >_RequestIndication	

Syntax	`void < User >_RequestIndication (`
	`  uint8 node,`
	`  NetworkHandleType channel,`
	`  uint32 requestedPgn,`
	`  const J1939Rm_ExtIdInfoType* extIdInfo,`
	`  uint8 sourceAddress,`
	`  uint8 destAddress,`
	`  uint8 priority`
	`)`	

Service ID [hex]	0x47	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	node	Node by which the request was received.
	channel	Channel on which the request was received.
	requestedPgn	PGN of the requested PG.
	extIdInfo	Extended identifier bytes.
	sourceAddress	Address of the node that sent the Request PG.
	destAddress	Address of this node or 0xFF for broadcast.
	priority	Priority of the Request PG.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indicates reception of a Request or Request2 PG.	
Available via	<apiServicePrefix>_J1939Rm.h, where apiServicePrefix is taken from the CDD's module definition reference via J1939RmUserCddRef	

⌋

[SWS_J1939Rm_00085]

Upstream requirements: SRS_BSW_00171

⌈The configured `User_RequestIndication` function shall be available for each user that has `J1939RmUserRequestIndication` enabled.⌋

See section 7.4 for details.

8.6.3.2 <User>_AckIndication

[SWS_J1939Rm_00064] Definition of configurable interface < User >_AckIndication ⌈

Service Name	< User >_AckIndication	

Syntax	`void < User >_AckIndication (`
	`  uint8 node,`
	`  NetworkHandleType channel,`
	`  uint32 ackPgn,`
	`  const J1939Rm_ExtIdInfoType* extIdInfo,`
	`  J1939Rm_AckCode ackCode,`
	`  uint8 ackAddress,`
	`  uint8 sourceAddress,`
	`  uint8 priority`
	`)`	

Service ID [hex]	0x4d	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	node	Node by which the acknowledgement was received.
	channel	Channel on which the acknowledgement was received.
	ackPgn	Acknowledged PGN.
	extIdInfo	Extended identifier bytes.
	ackCode	Type of acknowledgement, see definition of J1939Rm_AckCode for available codes.
	ackAddress	Address of this node.
	sourceAddress	Address of the node that sent the Acknowledgement PG.
	priority	Priority of the Acknowledgement PG.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indicates reception of an Acknowledgement PG.	
Available via	<apiServicePrefix>_J1939Rm.h, where apiServicePrefix is taken from the CDD's module definition reference via J1939RmUserCddRef	

⌋

[SWS_J1939Rm_00086]

Upstream requirements: SRS_BSW_00171

⌈The configured `User_AckIndication` function shall be available for each user that has `J1939RmUserAckIndication` enabled.⌋

See section 7.7 for details.

8.6.3.3 <User>_RequestTimeoutIndication

[SWS_J1939Rm_00065] Definition of configurable interface < User >_Request TimeoutIndication ⌈

Service Name	< User >_RequestTimeoutIndication	
Syntax	`void < User >_RequestTimeoutIndication ( uint8 node, NetworkHandleType channel, uint32 requestedPgn, const J1939Rm_ExtIdInfoType* extIdInfo, uint8 destAddress )`	
Service ID [hex]	0x4e	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	node	Node by which the request was sent.
	channel	Channel on which the request was sent.
	requestedPgn	PGN of the requested PG.
	extIdInfo	Extended identifier bytes.
	destAddress	Address of the destination node or 0xFF for broadcast.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indicates timeout of a request triggered with the same parameters.	
Available via	<apiServicePrefix>_J1939Rm.h, where apiServicePrefix is taken from the CDD's module definition reference via J1939RmUserCddRef	

⌋

[SWS_J1939Rm_00087]

Upstream requirements: SRS_BSW_00171

⌈The configured `User_RequestTimeoutIndication` function shall be available for each user that has `J1939RmUserTimeoutSupervision` enabled.⌋

See section 7.8 for details.

8.7 Service Interfaces

This section defines the client server interfaces and the derived service ports used by J1939Rm to communicate with application software components (SWCs).

8.7.1 Client-Server Interfaces

This section lists the client-server interfaces used by the ports provided and required by the `SAE J1939 Request Manager`.

Please note: The availability of these interfaces depends on the configuration of the SAE J1939 Request Manager. The relevant parameters of the SAE J1939 Request Manager configuration are listed as “Variation” of the operations.

8.7.1.1 AppSendAck

[SWS_J1939Rm_00103] Definition of ClientServerInterface AppSendAck ⌈

Name	AppSendAck		
Comment	–		
IsService	true		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportAckTransmission)} == true		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SendAck	
Comment	Requests transmission of an Acknowledgement PG.	
Relates to	J1939Rm_SendAck	
Variation	–	
Parameters	channel	
	Type	NetworkHandleType
	Direction	IN
	Comment	Channel on which the acknowledgement shall be sent.
	Variation	–
	ackPgn	
	Type	uint32
	Direction	IN
	Comment	Acknowledged PGN.
	Variation	–
	extIdInfo	
	Type	J1939Rm_ExtIdInfoType
	Direction	IN
	Comment	–
	Variation	–
	ackCode	
	Type	J1939Rm_AckCode
	Direction	IN
	Comment	Type of acknowledgement, see definition of J1939Rm_AckCode for available codes.
	Variation	–
	ackAddress	
	Type	uint8
	Direction	IN
	Comment	Address of the node that sent the request.
	Variation	–
	priority	

⌋

△

	Type	uint8
	Direction	IN
	Comment	Priority of the Acknowledgement PG.
	Variation	–
	broadcast	
	Type	boolean
	Direction	IN
	Comment	Indicates whether the ACKM is a response to a broadcast request.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.7.1.2 AppSendRequest

[SWS_J1939Rm_00104] Definition of ClientServerInterface AppSendRequest ⌈

Name	AppSendRequest		
Comment	–		
IsService	true		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportRequestTransmission)} == true		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SendRequest	
Comment	Requests transmission of a Request or Request2 PG.	
Relates to	J1939Rm_SendRequest	
Variation	–	
Parameters	channel	
	Type	NetworkHandleType
	Direction	IN
	Comment	Channel on which the request shall be sent.
	Variation	–
	requestedPgn	
	Type	uint32
	Direction	IN
	Comment	PGN of the requested PG.
	Variation	–
	extIdInfo	
	Type	J1939Rm_ExtIdInfoType
	Direction	IN
	Comment	–
	Variation	–

▽

△

	destAddress	
	Type	uint8
	Direction	IN
	Comment	Address of the destination node or 0xFF for broadcast.
	Variation	–
	priority	
	Type	uint8
	Direction	IN
	Comment	Priority of the Request PG.
	Variation	–
	checkTimeout	
	Type	boolean
	Direction	IN
	Comment	TRUE: Timeout supervision will be performed FALSE: No timeout supervision will be started
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.7.1.3 AppCancelRequestTimeout

[SWS_J1939Rm_00105] Definition of ClientServerInterface AppCancelRequestTimeout ⌈

Name	AppCancelRequestTimeout		
Comment	–		
IsService	true		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportTimeoutSupervision)} == true		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	CancelRequestTimeout	
Comment	Cancels timeout monitoring of a request. If the request is not active, or timeout monitoring was not requested, this call has no effect.	
Relates to	J1939Rm_CancelRequestTimeout	
Variation	–	
Parameters	channel	
	Type	NetworkHandleType
	Direction	IN
	Comment	Channel on which the request was sent.
	Variation	–
	requestedPgn	
	Type	uint32

▽

△

	Direction	IN
	Comment	PGN of the requested PG.
	Variation	–
	extIdInfo	
	Type	J1939Rm_ExtIdInfoType
	Direction	IN
	Comment	–
	Variation	–
	destAddress	
	Type	uint8
	Direction	IN
	Comment	Address of the destination node.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.7.1.4 AppAckIndication

[SWS_J1939Rm_00106] Definition of ClientServerInterface AppAckIndication ⌈

Name	AppAckIndication		
Comment	–		
IsService	true		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportAckIndication)} == true		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	AckIndication	
Comment	Indicates reception of an Acknowledgement PG.	
Relates to	< User >_AckIndication	
Variation	–	
Parameters	node	
	Type	uint8
	Direction	IN
	Comment	Node by which the acknowledgement was received.
	Variation	–
	channel	
	Type	NetworkHandleType
	Direction	IN
	Comment	Channel on which the acknowledgement was received.
	Variation	–
	ackPgn	
	Type	uint32

▽

△

	Direction	IN
	Comment	Acknowledged PGN.
	Variation	–
	extIdInfo	
	Type	J1939Rm_ExtIdInfoType
	Direction	IN
	Comment	Extended identifier bytes.
	Variation	–
	ackCode	
	Type	J1939Rm_AckCode
	Direction	IN
	Comment	Type of acknowledgement, see definition of J1939Rm_AckCode for available codes.
	Variation	–
	ackAddress	
	Type	uint8
	Direction	IN
	Comment	Address of this node.
	Variation	–
	sourceAddress	
	Type	uint8
	Direction	IN
	Comment	Address of the node that sent the Acknowledgement PG.
	Variation	–
	priority	
	Type	uint8
	Direction	IN
	Comment	Priority of the Acknowledgement PG.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.7.1.5 AppRequestIndication

[SWS_J1939Rm_00107] Definition of ClientServerInterface AppRequestIndication ⌈

Name	AppRequestIndication		
Comment	–		
IsService	true		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportRequestIndication)} == true		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	RequestIndication	
Comment	Indicates reception of a Request or Request2 PG.	
Relates to	< User >_RequestIndication	
Variation	–	
Parameters	node	
	Type	uint8
	Direction	IN
	Comment	Node by which the request was received.
	Variation	–
	channel	
	Type	NetworkHandleType
	Direction	IN
	Comment	Channel on which the request was received.
	Variation	–
	requestedPgn	
	Type	uint32
	Direction	IN
	Comment	PGN of the requested PG.
	Variation	–
	extIdInfo	
	Type	J1939Rm_ExtIdInfoType
	Direction	IN
	Comment	Extended identifier bytes.
	Variation	–
	sourceAddress	
	Type	uint8
	Direction	IN
	Comment	Address of the node that sent the Request PG.
	Variation	–
	destAddress	
	Type	uint8
	Direction	IN
	Comment	Address of this node or 0xFF for broadcast.
	Variation	–
	priority	
	Type	uint8
	Direction	IN
	Comment	Priority of the Request PG.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.7.1.6 AppRequestTimeoutIndication

[SWS_J1939Rm_00108] Definition of ClientServerInterface AppRequestTimeoutIndication ⌈

Name	AppRequestTimeoutIndication		
Comment	–		
IsService	true		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportTimeoutSupervision)} == true		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	RequestTimeoutIndication	
Comment	Indicates timeout of a request triggered with the same parameters.	
Relates to	< User >_RequestTimeoutIndication	
Variation	–	
Parameters	node	
	Type	uint8
	Direction	IN
	Comment	Node by which the request was sent.
	Variation	–
	channel	
	Type	NetworkHandleType
	Direction	IN
	Comment	Channel on which the request was sent.
	Variation	–
	requestedPgn	
	Type	uint32
	Direction	IN
	Comment	PGN of the requested PG.
	Variation	–
	extIdInfo	
	Type	J1939Rm_ExtIdInfoType
	Direction	IN
	Comment	Extended identifier bytes.
	Variation	–
	destAddress	
	Type	uint8
	Direction	IN
	Comment	Address of the destination node or 0xFF for broadcast.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.7.2 Implementation Data Types

In this section, the implementation data types used by the client-server interfaces of the SAE J1939 Request Manager are listed.

Please note: It is essential that the implementation of the SAE J1939 Request Manager does not define these data types twice, by including them both from the RTE generated header and the own types header.

8.7.2.1 J1939Rm_AckCode

[SWS_J1939Rm_00057] Definition of ImplementationDataType J1939Rm_AckCode ⌈

Name	J1939Rm_AckCode		
Kind	Type		
Derived from	uint8		
Range	J1939RM_ACK_POSITIVE	0x00	Positive Acknowledgement
	J1939RM_ACK_NEGATIVE	0x01	Negative Acknowledgement
	J1939RM_ACK_ACCESS_DENIED	0x02	Access Denied
	J1939RM_ACK_CANNOT_RESPOND	0x03	Cannot Respond
Description	This type represents the available kinds of acknowledgements.		
Variation	–		
Available via	Rte_J1939Rm_Type.h		

⌋

8.7.2.2 J1939Rm_ExtIdType

[SWS_J1939Rm_91000] Definition of ImplementationDataType J1939Rm_ExtIdType ⌈

Name	J1939Rm_ExtIdType		
Kind	Type		
Derived from	uint8		
Range	J1939RM_EXTID_NONE	0x00	No extended identifier bytes (0)
	J1939RM_EXTID_ONE	0x01	One extended identifier byte (1)
	J1939RM_EXTID_TWO	0x02	Two extended identifier bytes (2)
	J1939RM_EXTID_THREE	0x03	Three extended identifier bytes (3)
	J1939RM_EXTID_GF	0x04	Group function value, only for ACKM (4)
Description	This type represents the available kinds of extended identifier usage.		
Variation	–		
Available via	Rte_J1939Rm_Type.h		

⌋

8.7.2.3 J1939Rm_ExtIdInfoType

[SWS_J1939Rm_91001] Definition of ImplementationDataType J1939Rm_ExtId InfoType ⌈

Name	J1939Rm_ExtIdInfoType	
Kind	Structure	
Elements	extIdType	
	Type	J1939Rm_ExtIdType
	Comment	Denotes the number of extended identifier bytes.
	extId1	
	Type	uint8
	Comment	First extended identifier byte or group function for ACKM.
	extId2	
	Type	uint8
	Comment	Second extended identifier byte.
	extId3	
	Type	uint8
	Comment	Third extended identifier byte.
Description	This type represents a set of extended identifiers.	
Variation	–	
Available via	Rte_J1939Rm_Type.h	

⌋

8.7.3 Provided Service Ports

These service ports provide API functions of the J1939Rm to the application SWCs.

Please note: All three ports use a port defined argument value to provide the userId argument of the corresponding BSW interfaces.

8.7.3.1 J1939Rm_SendAck

[SWS_J1939Rm_00098] Definition of Port J1939Rm_SendAck_{user} provided by module J1939Rm ⌈

Name	J1939Rm_SendAck_{user}		
Kind	ProvidedPort	***Interface***	AppSendAck
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	{ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmRteUser/J1939RmUserId.value)}	
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportAckTransmission)} == true user = {ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser.SHORT-NAME)}		

⌋

8.7.3.2 J1939Rm_SendRequest

[SWS_J1939Rm_00097] Definition of Port J1939Rm_SendRequest_{user} provided by module J1939Rm ⌈

Name	J1939Rm_SendRequest_{user}		
Kind	ProvidedPort	***Interface***	AppSendRequest
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	{ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmRteUser/J1939RmUserId.value)}	
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportRequestTransmission)} == true user = {ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser.SHORT-NAME)}		

⌋

8.7.3.3 J1939Rm_CancelRequestTimeout

[SWS_J1939Rm_00099] Definition of Port J1939Rm_CancelRequestTimeout_{user} provided by module J1939Rm ⌈

Name	J1939Rm_CancelRequestTimeout_{user}		
Kind	ProvidedPort	***Interface***	AppCancelRequestTimeout
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	{ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmRteUser/J1939RmUserId.value)}	
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportTimeoutSupervision)} == true user = {ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser.SHORT-NAME)}		

⌋

8.7.4 Required Service Ports

These service ports provide call back functions of the J1939Rm to the application SWCs.

8.7.4.1 J1939Rm_AckIndication

[SWS_J1939Rm_00101] Definition of Port J1939Rm_AckIndication_{user} required by module J1939Rm ⌈

Name	J1939Rm_AckIndication_{user}		
Kind	RequiredPort	***Interface***	AppAckIndication
Description	–		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportAckIndication)} == true user = {ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser.SHORT-NAME)}		

⌋

8.7.4.2 J1939Rm_RequestIndication

[SWS_J1939Rm_00100] Definition of Port J1939Rm_RequestIndication_{user} required by module J1939Rm ⌈

Name	J1939Rm_RequestIndication_{user}		
Kind	RequiredPort	***Interface***	AppRequestIndication
Description	–		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportRequestIndication)} == true user = {ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser.SHORT-NAME)}		

⌋

8.7.4.3 J1939Rm_RequestTimeoutIndication

[SWS_J1939Rm_00102] Definition of Port J1939Rm_RequestTimeoutIndication_{user} required by module J1939Rm ⌈

Name	J1939Rm_RequestTimeoutIndication_{user}		
Kind	RequiredPort	***Interface***	AppRequestTimeoutIndication
Description	–		
Variation	{ecuc(J1939Rm/J1939RmGeneral.J1939RmSupportTimeoutSupervision)} == true user = {ecuc(J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser.SHORT-NAME)}		

⌋

9 Sequence Diagrams

The following sequence diagrams shall give an impression of the way the SAE J1939 Request Manager shall behave and interoperate with other BSW modules. They are not complete and not binding for the implementation.

9.1 Reception of Request PG

The following diagram shows the interaction with PduR and a J1939Rm User when a Request PG is received.

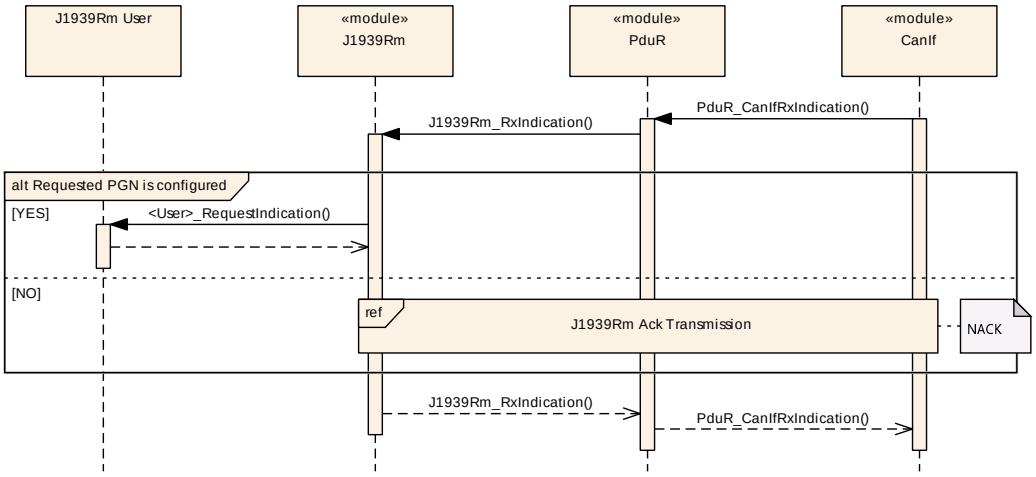

Figure 9.1: Reception of Request PG

9.2 Transmission of Acknowledgement PG

The following diagram shows the interaction with a J1939Rm User and PduR when an Acknowledgement PG is transmitted.

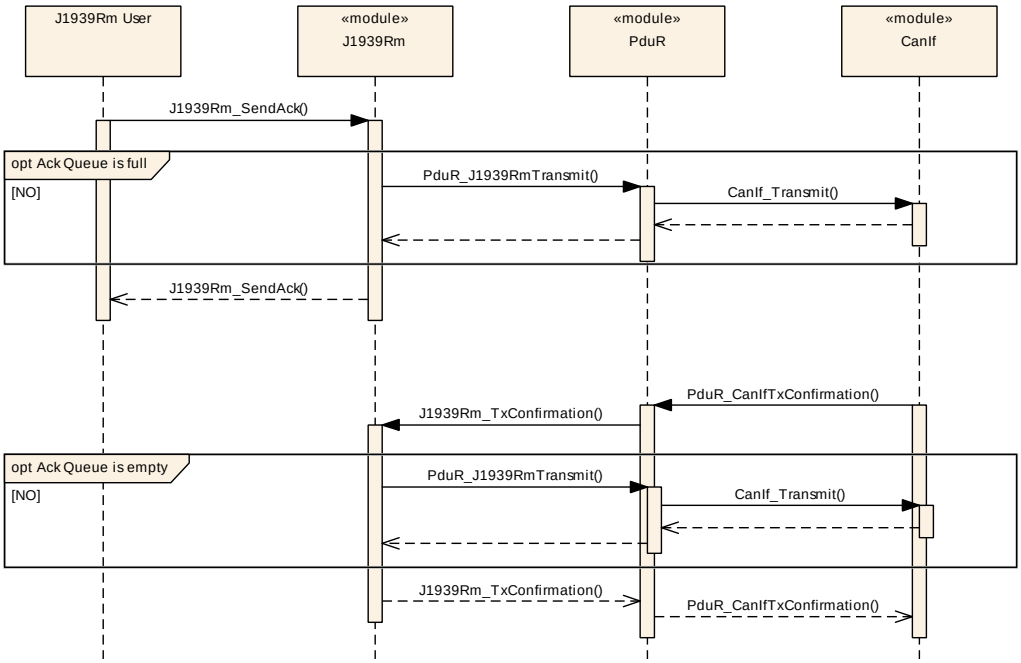

Figure 9.2: Transmission of Acknowledgement PG

9.3 Handling of Request for a COM PDU

The following diagram shows the interaction with PduR and COM when the SAE J1939 Request Manager receives a Request for a PG of PDU1 format that is transmitted as COM PDU.

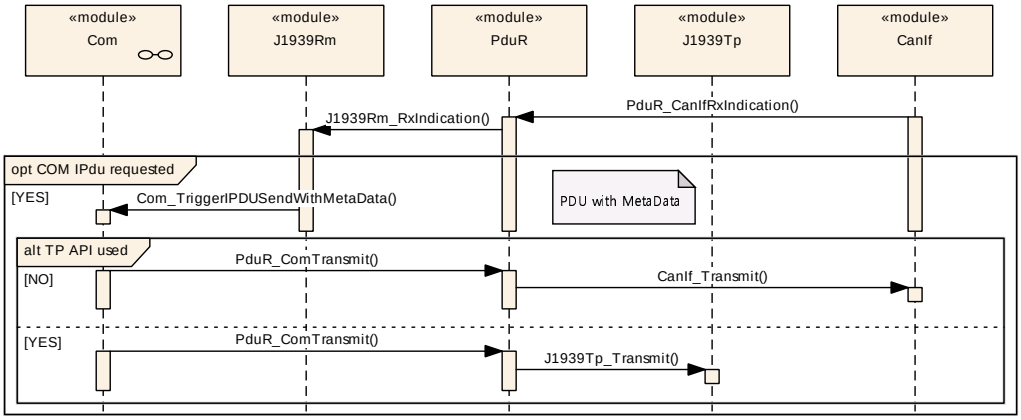

Figure 9.3: Handling of Request for a COM PDU with PDU1 format

9.4 Handling of Request for a Diagnostic PDU

The following diagram shows the interaction with PduR and J1939Dcm when a request for a diagnostic PG is received.

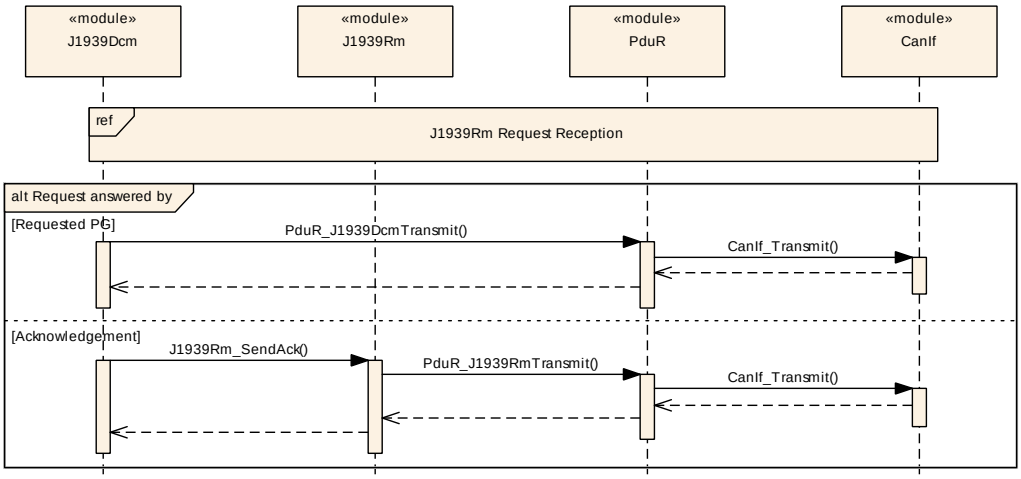

Figure 9.4: Handling of Request for a Diagnostic Pdu

9.5 Transmission of Request PG

The following diagram shows the interaction with a J1939Rm User and PduR when a `Request` PG is transmitted.

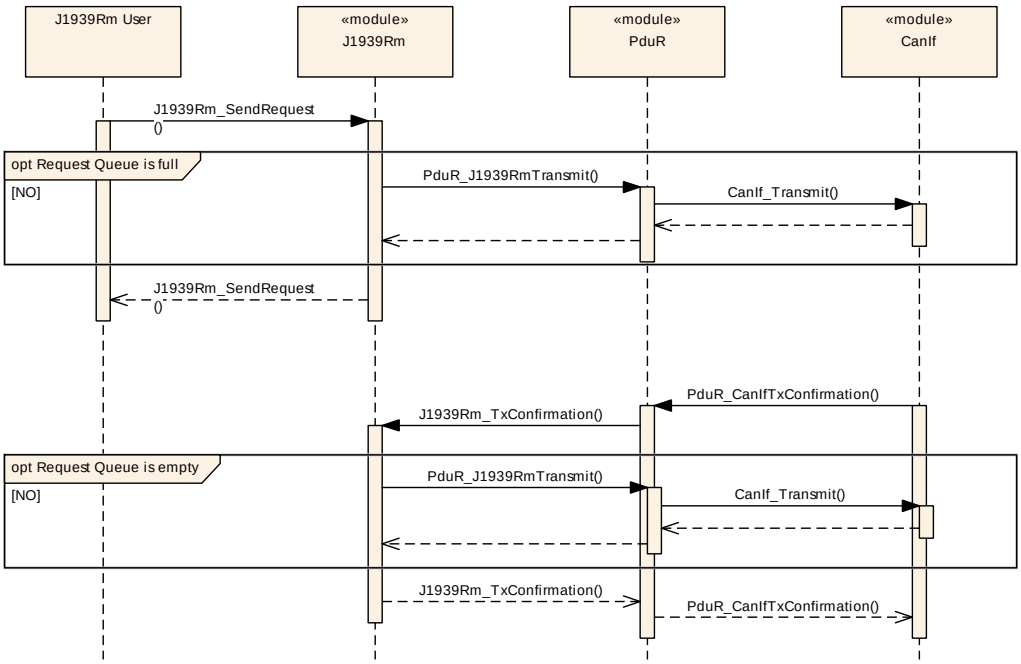

Figure 9.5: Transmission of Request PG

9.6 Reception of Acknowledgement PG

The following diagram shows the interaction with PduR and a J1939Rm User when an `Acknowledgement` PG is received.

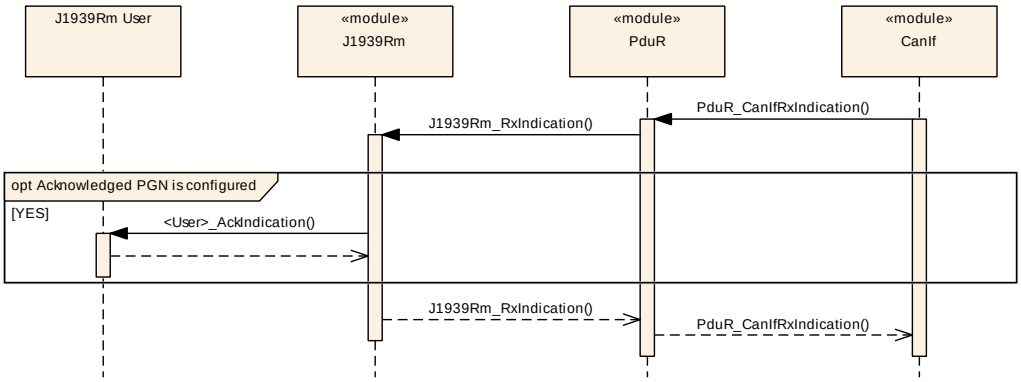

Figure 9.6: Reception of Acknowledgement PG

9.7 Monitoring of Request Timeout

The following diagram shows the interaction with a J1939Rm User and PduR when the J1939Rm monitors timeout of a transmitted `Request` PG.

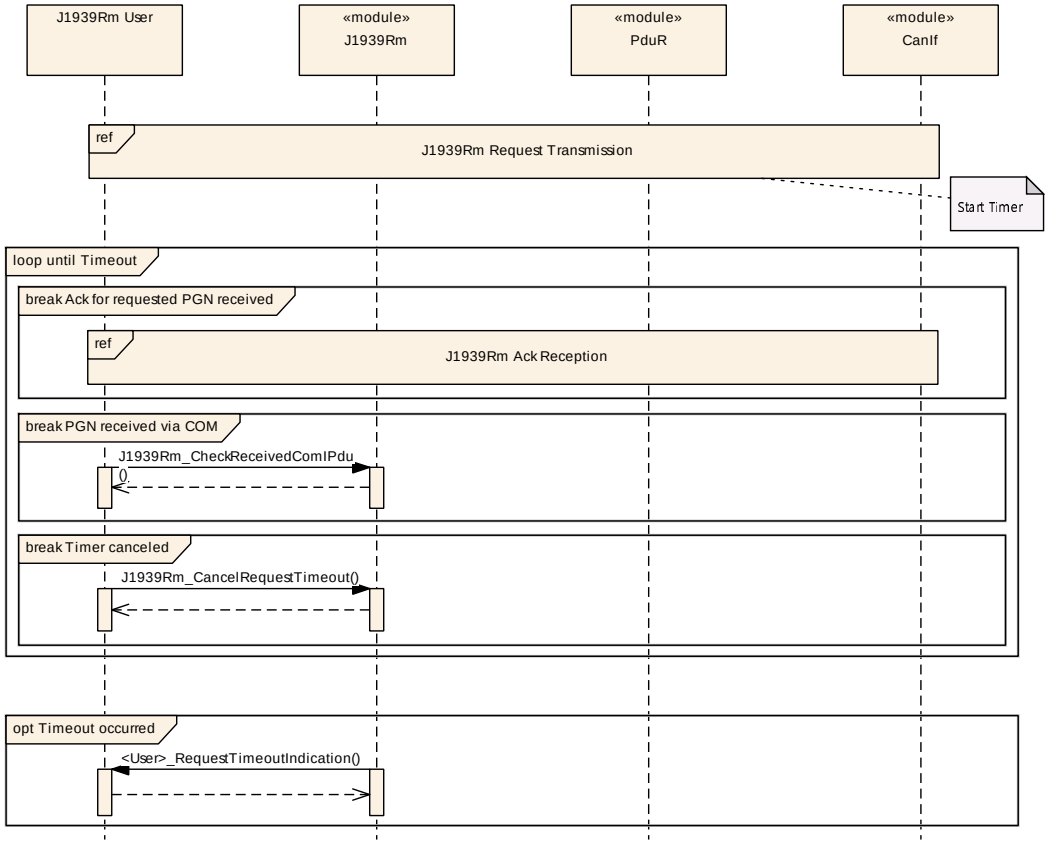

Figure 9.7: Monitoring of Request Timeout

10 Configuration Specification

In general, this chapter defines configuration parameters and their clustering into containers. For general information about the definition of containers and parameters, refer to [19, SWS BSW General] Chapter 10.1 *"Introduction to configuration specification"*. For details about published information of the `SAE J1939 Request Manager`, refer to [19, SWS BSW General] Chapter 10.3 *"Published Information"*.

The Chapter 10.1 specifies the structure (containers) and the parameters of the `SAE J1939 Request Manager`.

10.1 Containers and Configuration Parameters

The following sections summarize all configuration parameters of the `SAE J1939 Request Manager`. The detailed meaning of the parameters is described in Chapter 7 and Chapter 8.

Some of these containers and parameters are derived from classes and attributes of the [24, TPS System Template], which also contains the rules for these derivations.

The following pictures show an overview of the configuration parameters available for J1939Rm:

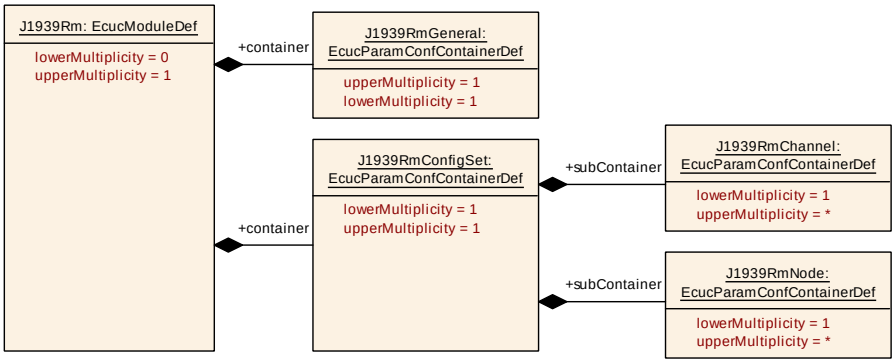

Figure 10.1: Configuration container J1939Rm with subcontainer J1939RmConfigSet

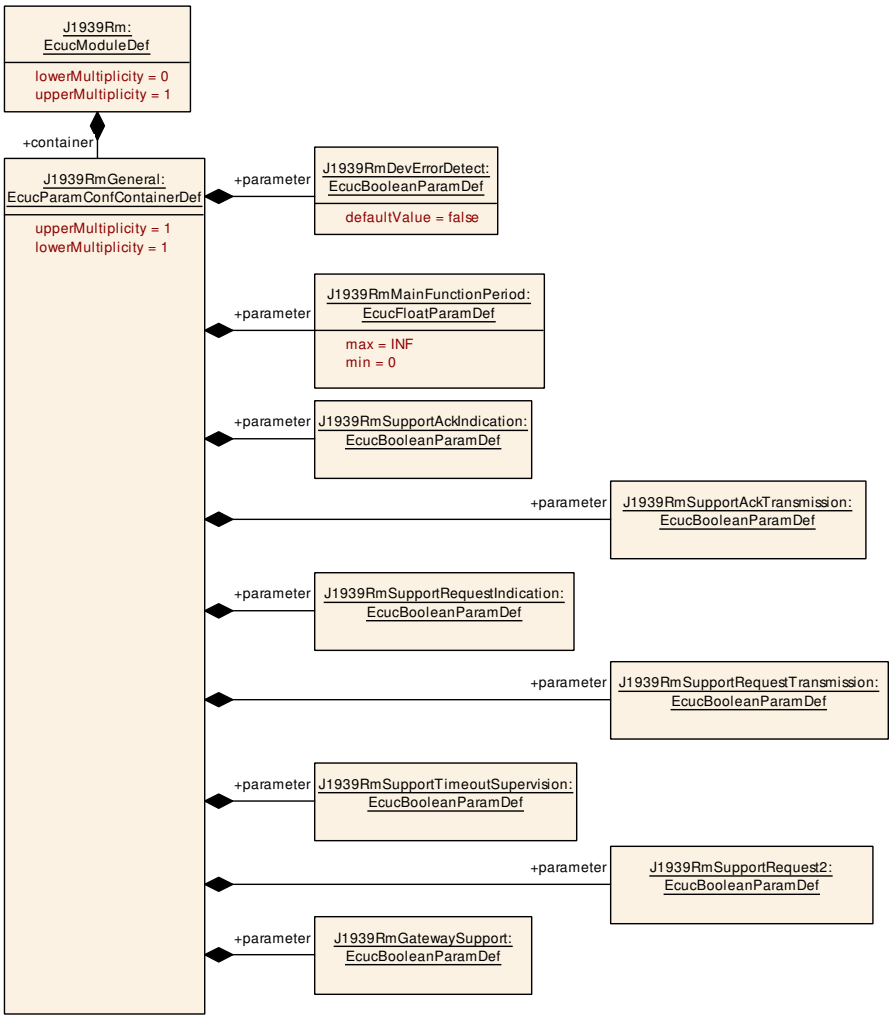

Figure 10.2: Configuration container J1939RmGeneral

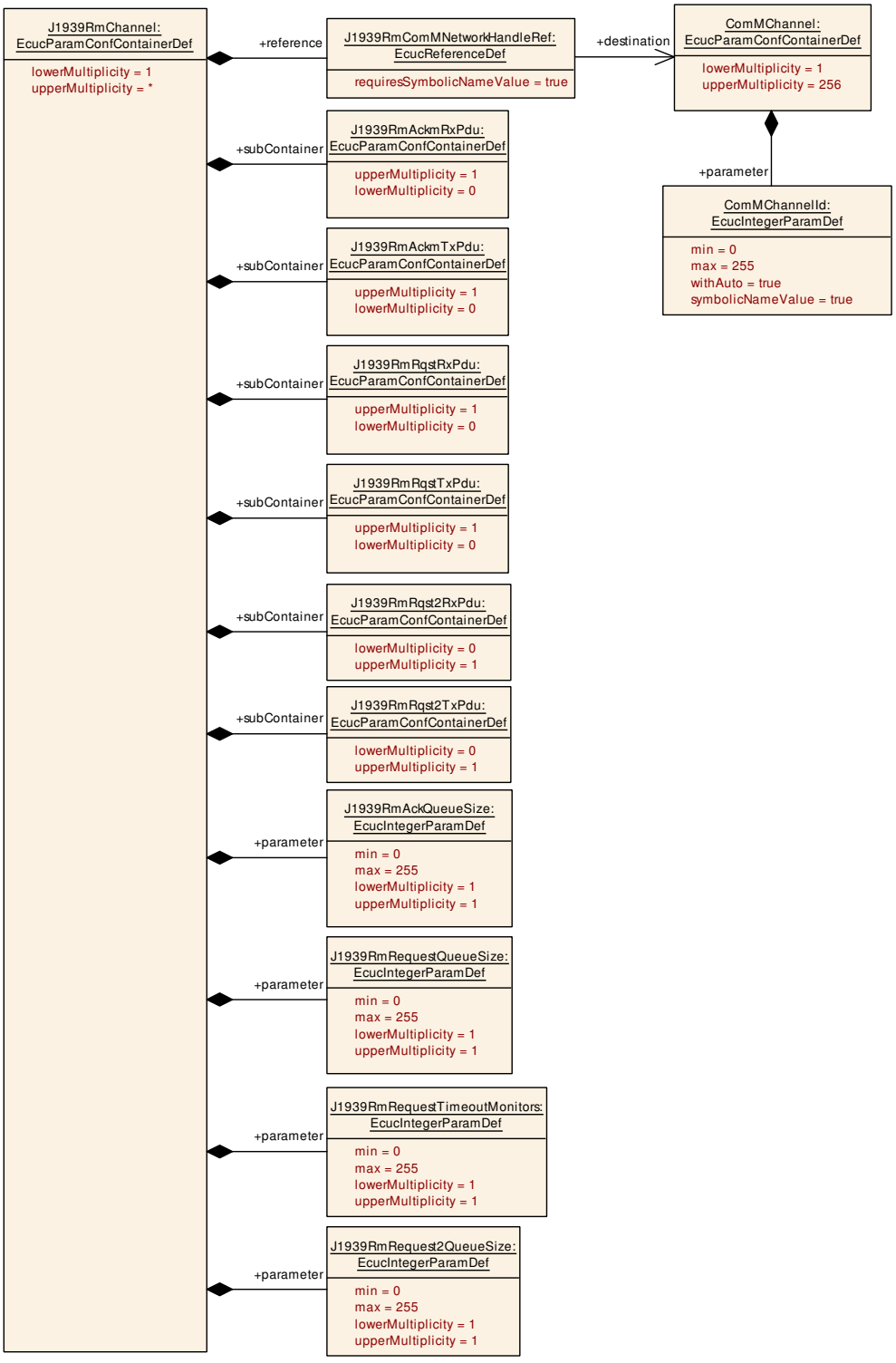

Figure 10.3: Configuration container J1939RmChannel

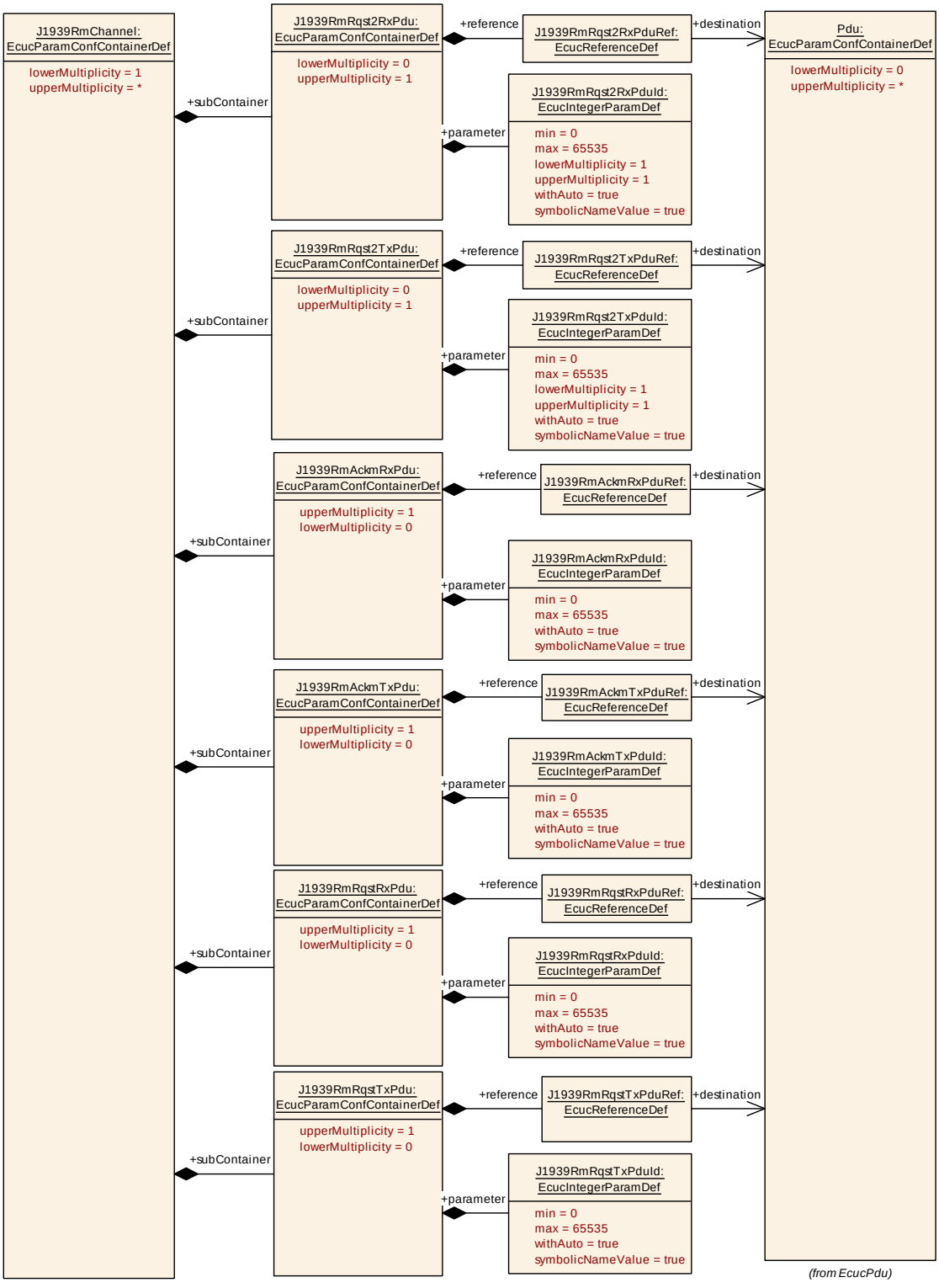

Figure 10.4: Configuration container J1939RmChannel with PDUs

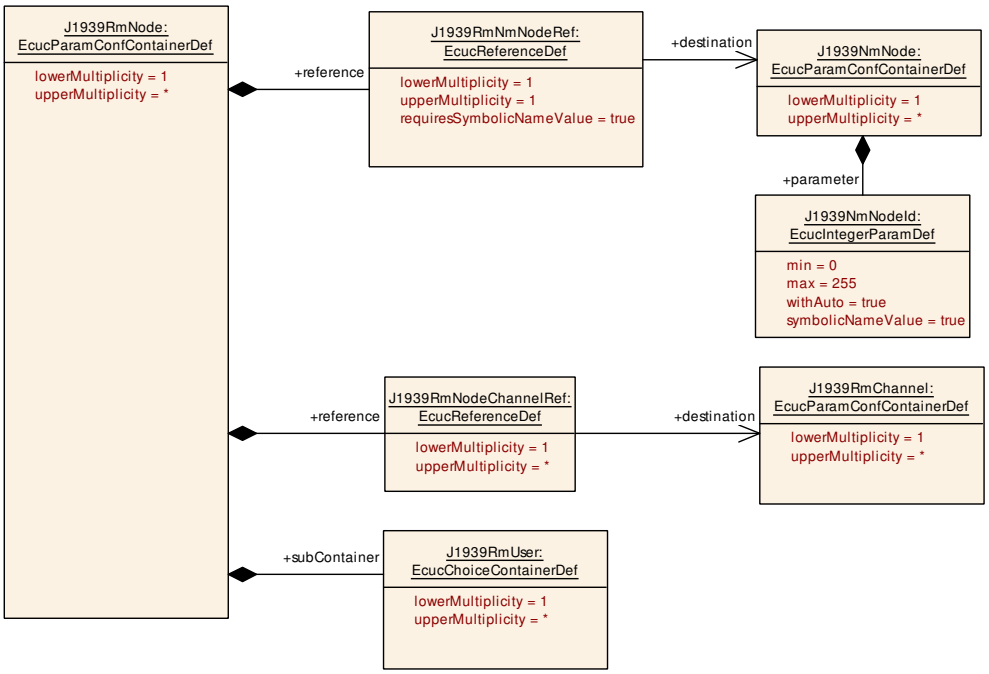

Figure 10.5: Configuration container J1939RmNode

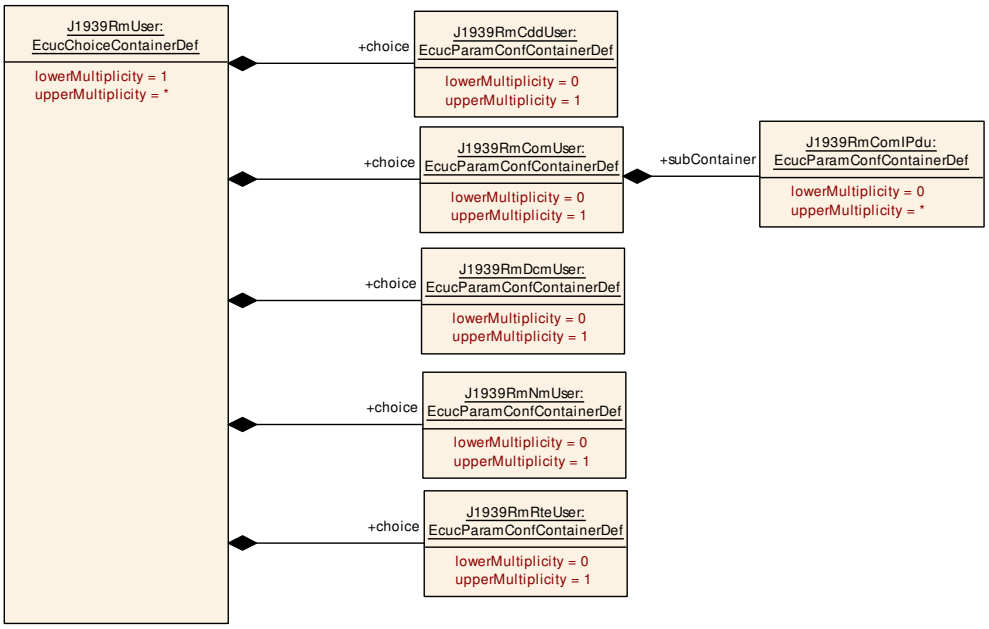

Figure 10.6: Configuration container J1939RmUser

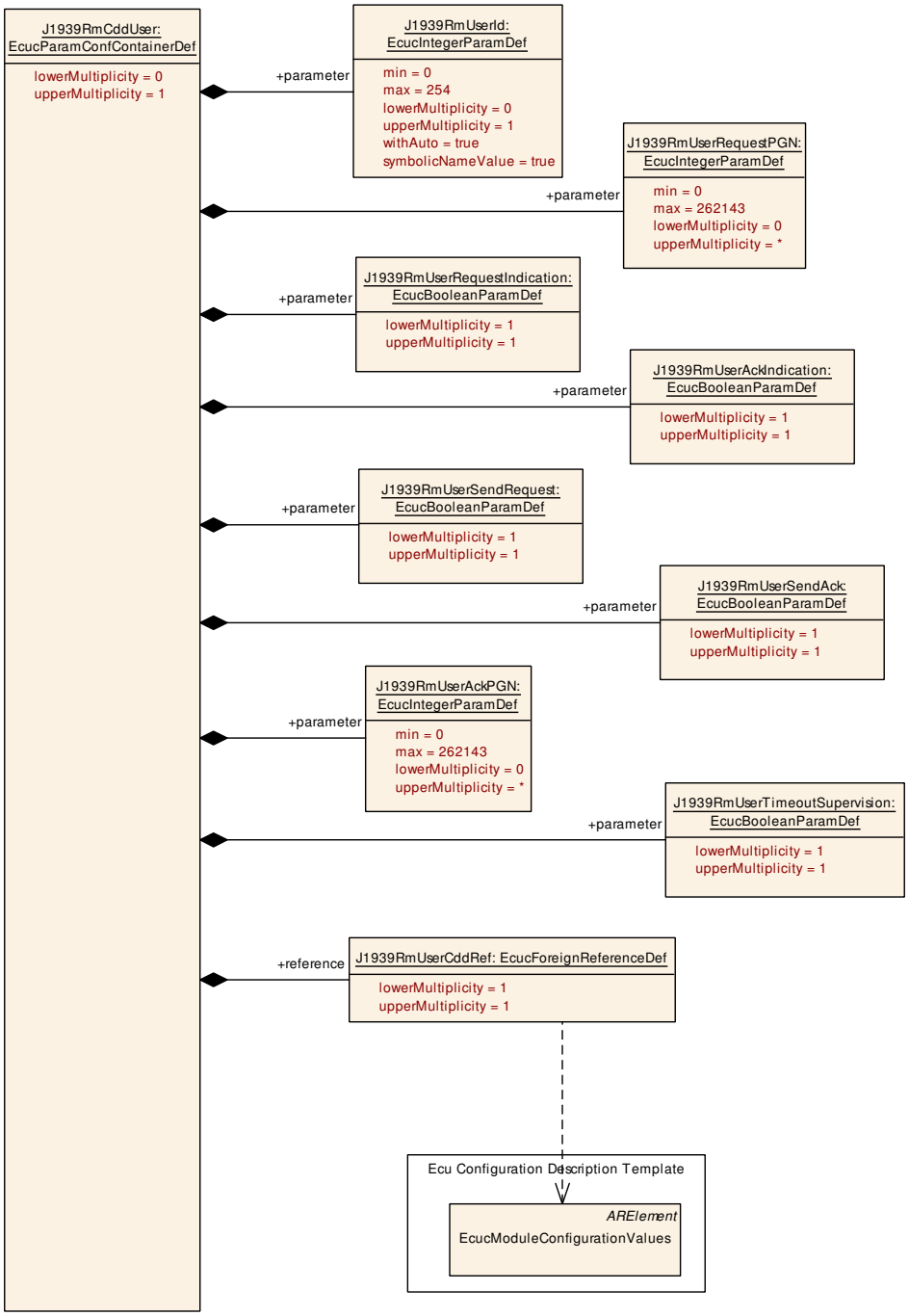

Figure 10.7: Configuration container J1939RmCddUser

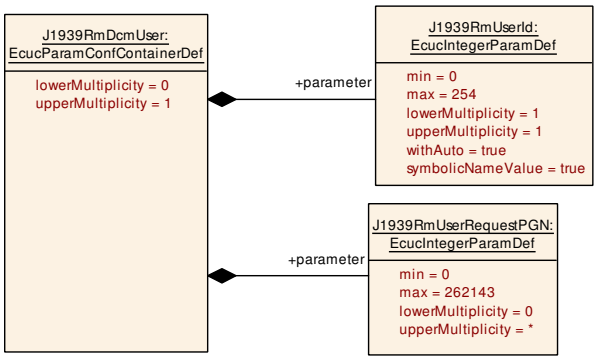

Figure 10.8: Configuration container J1939RmDcmUser

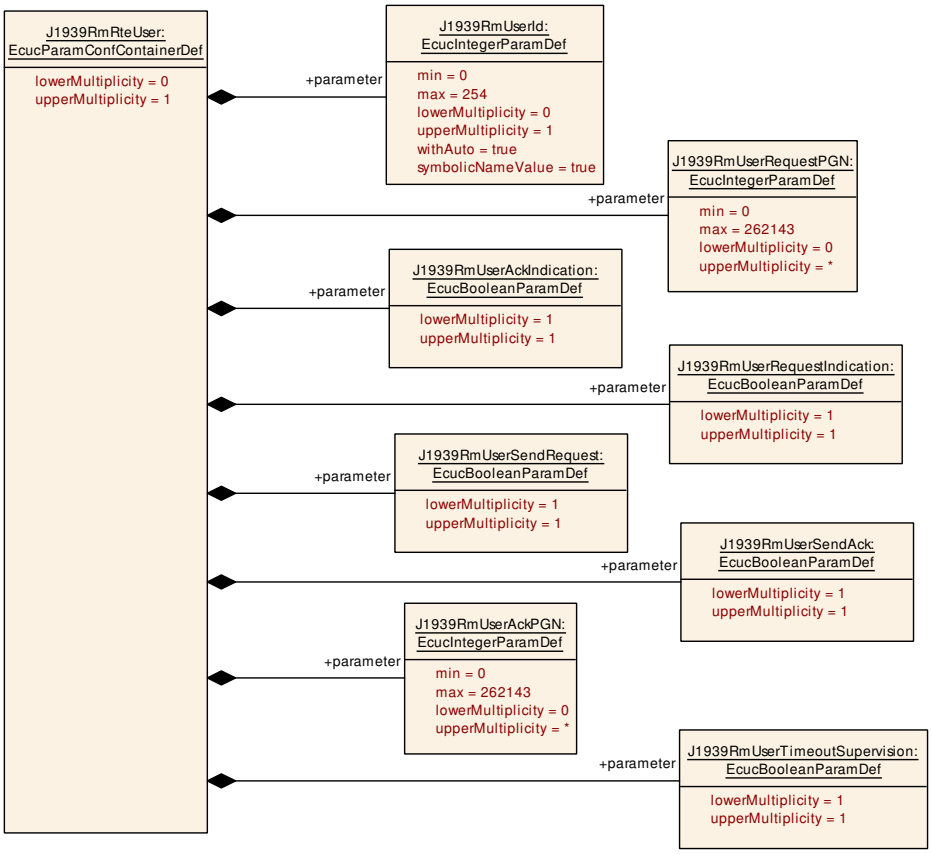

Figure 10.9: Configuration container J1939RmRteUser

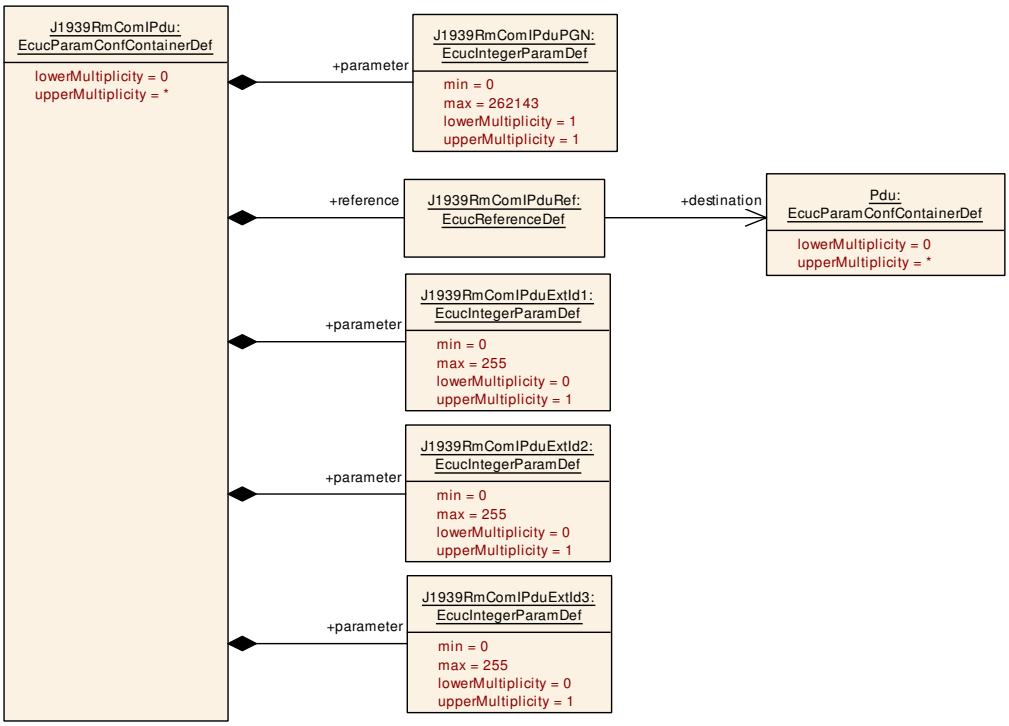

Figure 10.10: Configuration container J1939RmComIPdu

10.1.1 J1939Rm

[ECUC_J1939Rm_00043] Definition of EcucModuleDef J1939Rm ⌈

Module Name	J1939Rm
Description	Configuration of the J1939 Request Manager.
Post-Build Selectable Support	true
Post-Build Loadable Support	true

Included Containers		
Container Name	**Multiplicity**	**Description**
J1939RmConfigSet	1	This container contains the configuration parameters and sub containers of the AUTOSAR J1939Rm module.
J1939RmGeneral	1	Contains the general configuration parameters of the module.

⌋

10.1.2 J1939RmGeneral

[ECUC_J1939Rm_00001] Definition of EcucParamConfContainerDef J1939Rm General ⌈

Container Name	J1939RmGeneral
Parent Container	J1939Rm
Upstream mapping defined	–
Description	Contains the general configuration parameters of the module.
Multiplicity	1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmDevErrorDetect	1	[ECUC_J1939Rm_00003]
J1939RmGatewaySupport	1	[ECUC_J1939Rm_00084]
J1939RmMainFunctionPeriod	1	[ECUC_J1939Rm_00004]
J1939RmSupportAckIndication	1	[ECUC_J1939Rm_00054]
J1939RmSupportAckTransmission	1	[ECUC_J1939Rm_00055]
J1939RmSupportRequest2	1	[ECUC_J1939Rm_00073]
J1939RmSupportRequestIndication	1	[ECUC_J1939Rm_00056]
J1939RmSupportRequestTransmission	1	[ECUC_J1939Rm_00057]
J1939RmSupportTimeoutSupervision	1	[ECUC_J1939Rm_00058]

No Included Containers

⌋

[ECUC_J1939Rm_00003] Definition of EcucBooleanParamDef J1939RmDevErrorDetect ⌈

Parameter Name	J1939RmDevErrorDetect
Parent Container	J1939RmGeneral
Upstream mapping defined	–
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00084] Definition of EcucBooleanParamDef J1939RmGatewaySupport ⌈

Parameter Name	J1939RmGatewaySupport
Parent Container	J1939RmGeneral
Upstream mapping defined	–
Description	Enables/disables support for routing Request and Acknowledgement messages.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00004] Definition of EcucFloatParamDef J1939RmMainFunctionPeriod ⌈

Parameter Name	J1939RmMainFunctionPeriod	
Parent Container	J1939RmGeneral	
Upstream mapping defined	–	
Description	Execution cycle of J1939Rm_MainFunction in seconds.	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	]0 .. INF[	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00054] Definition of EcucBooleanParamDef J1939RmSupportAckIndication ⌈

Parameter Name	J1939RmSupportAckIndication
Parent Container	J1939RmGeneral
Upstream mapping defined	–
Description	Pre-processor switch for enabling support of acknowledgement indications.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00055] Definition of EcucBooleanParamDef J1939RmSupportAckTransmission ⌈

Parameter Name	J1939RmSupportAckTransmission
Parent Container	J1939RmGeneral
Upstream mapping defined	–
Description	Pre-processor switch for enabling support of acknowledgement transmission.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00073] Definition of EcucBooleanParamDef J1939RmSupportRequest2 ⌈

Parameter Name	J1939RmSupportRequest2
Parent Container	J1939RmGeneral
Upstream mapping defined	[24, CP_TPS_SystemTemplate]
Description	Pre-processor switch for enabling support of the Request2 PG. Please note: Transfer is not supported.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00056] Definition of EcucBooleanParamDef J1939RmSupportRequestIndication ⌈

Parameter Name	J1939RmSupportRequestIndication
Parent Container	J1939RmGeneral
Upstream mapping defined	–
Description	Pre-processor switch for enabling support of request indications.

▽

△

Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00057] Definition of EcucBooleanParamDef J1939RmSupportRequestTransmission ⌈

Parameter Name	J1939RmSupportRequestTransmission
Parent Container	J1939RmGeneral
Upstream mapping defined	–
Description	Pre-processor switch for enabling support of request transmission.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00058] Definition of EcucBooleanParamDef J1939RmSupportTimeoutSupervision ⌈

Parameter Name	J1939RmSupportTimeoutSupervision
Parent Container	J1939RmGeneral
Upstream mapping defined	–
Description	Pre-processor switch for enabling support of request timeout supervision.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

10.1.3 J1939RmConfigSet

[ECUC_J1939Rm_00017] Definition of EcucParamConfContainerDef J1939RmConfigSet ⌈

Container Name	J1939RmConfigSet
Parent Container	J1939Rm
Upstream mapping defined	–
Description	This container contains the configuration parameters and sub containers of the AUTOSAR J1939Rm module.
Multiplicity	1

▽

△

Pre-Compile only	true
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
J1939RmChannel	1..*	Contains the parameters for a CAN channel supported by the J1939 Request Manager.
J1939RmNode	1..*	Contains the parameters for the support of a logical J1939 node (identified by an ECU address).

⌋

10.1.4 J1939RmChannel

[ECUC_J1939Rm_00009] Definition of EcucParamConfContainerDef J1939Rm Channel ⌈

Container Name	J1939RmChannel	
Parent Container	J1939RmConfigSet	
Upstream mapping defined	[24, CP_TPS_SystemTemplate]	
Description	Contains the parameters for a CAN channel supported by the J1939 Request Manager.	
Multiplicity	1..*	
Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmAckQueueSize	1	[ECUC_J1939Rm_00007]
J1939RmRequest2QueueSize	1	[ECUC_J1939Rm_00074]
J1939RmRequestQueueSize	1	[ECUC_J1939Rm_00006]
J1939RmRequestTimeoutMonitors	1	[ECUC_J1939Rm_00008]
J1939RmComMNetworkHandleRef	1	[ECUC_J1939Rm_00051]

Included Containers		
Container Name	**Multiplicity**	**Description**
J1939RmAckmRxPdu	0..1	Contains the configuration of the I-PDU used to receive the Acknowledgement PG. This PDU consumes a meta data item of type CAN_ID_32.
J1939RmAckmTxPdu	0..1	Contains the configuration of the I-PDU used to transmit the Acknowledgement PG. This PDU produces a meta data item of type CAN_ID_32.
J1939RmRqst2RxPdu	0..1	Contains the configuration of the I-PDU used to receive the Request2 PG. This PDU consumes a meta data item of type CAN_ID_32.

▽

△

Included Containers		
Container Name	**Multiplicity**	**Description**
J1939RmRqst2TxPdu	0..1	Contains the configuration of the I-PDU used to transmit the Request2 PG. This PDU produces a meta data item of type CAN_ID_32.
J1939RmRqstRxPdu	0..1	Contains the configuration of the I-PDU used to receive the Request PG. This PDU consumes a meta data item of type CAN_ID_32.
J1939RmRqstTxPdu	0..1	Contains the configuration of the I-PDU used to transmit the Request PG. This PDU produces a meta data item of type CAN_ID_32.

⌋

[ECUC_J1939Rm_00007] Definition of EcucIntegerParamDef J1939RmAckQueueSize ⌈

Parameter Name	J1939RmAckQueueSize	
Parent Container	J1939RmChannel	
Upstream mapping defined	–	
Description	Number of transmitted Acknowledgement messages that can be stored.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00074] Definition of EcucIntegerParamDef J1939RmRequest2QueueSize ⌈

Parameter Name	J1939RmRequest2QueueSize	
Parent Container	J1939RmChannel	
Upstream mapping defined	–	
Description	Number of transmitted Request2 messages that can be stored.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00006] Definition of EcucIntegerParamDef J1939RmRequestQueueSize ⌈

Parameter Name	J1939RmRequestQueueSize
Parent Container	J1939RmChannel
Upstream mapping defined	–

▽

△

Description	Number of transmitted Request messages that can be stored.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00008] Definition of EcucIntegerParamDef J1939RmRequestTimeoutMonitors ⌈

Parameter Name	J1939RmRequestTimeoutMonitors	
Parent Container	J1939RmChannel	
Upstream mapping defined	–	
Description	Number of transmitted requests that can be monitored for timeout.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00051] Definition of EcucReferenceDef J1939RmComMNetworkHandleRef ⌈

Parameter Name	J1939RmComMNetworkHandleRef
Parent Container	J1939RmChannel
Upstream mapping defined	–
Description	Reference to the channel defined by the ComMChannel providing access to the unique channel index ComMChannelId.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.1.5 J1939RmAckmRxPdu

[ECUC_J1939Rm_00011] Definition of EcucParamConfContainerDef J1939RmAckmRxPdu ⌈

Container Name	J1939RmAckmRxPdu
Parent Container	J1939RmChannel
Upstream mapping defined	–

▽

△

Description	Contains the configuration of the I-PDU used to receive the Acknowledgement PG. This PDU consumes a meta data item of type CAN_ID_32.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmAckmRxPduId	1	[ECUC_J1939Rm_00015]
J1939RmAckmRxPduRef	1	[ECUC_J1939Rm_00016]

No Included Containers

⌋

[ECUC_J1939Rm_00015] Definition of EcucIntegerParamDef J1939RmAckmRxPduId ⌈

Parameter Name	J1939RmAckmRxPduId
Parent Container	J1939RmAckmRxPdu
Upstream mapping defined	–
Description	The I-PDU identifier used for RxIndication from PduR.
Multiplicity	1
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)
Range	0 .. 65535
Default value	–
With Auto	true
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00016] Definition of EcucReferenceDef J1939RmAckmRxPduRef ⌈

Parameter Name	J1939RmAckmRxPduRef
Parent Container	J1939RmAckmRxPdu
Upstream mapping defined	–
Description	Reference to the Pdu object representing the I-PDU.
Multiplicity	1
Type	Reference to Pdu
Pre-Compile only	true

⌋

10.1.6 J1939RmAckmTxPdu

[ECUC_J1939Rm_00012] Definition of EcucParamConfContainerDef J1939RmAckmTxPdu ⌈

Container Name	J1939RmAckmTxPdu
Parent Container	J1939RmChannel
Upstream mapping defined	–
Description	Contains the configuration of the I-PDU used to transmit the Acknowledgement PG. This PDU produces a meta data item of type CAN_ID_32.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
J1939RmAckmTxPduId	1	[ECUC_J1939Rm_00018]
J1939RmAckmTxPduRef	1	[ECUC_J1939Rm_00019]

No Included Containers

⌋

[ECUC_J1939Rm_00018] Definition of EcucIntegerParamDef J1939RmAckmTxPduId ⌈

Parameter Name	J1939RmAckmTxPduId	
Parent Container	J1939RmAckmTxPdu	
Upstream mapping defined	–	
Description	The I-PDU identifier used for TxConfirmation from PduR.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default value	–	
With Auto	true	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00019] Definition of EcucReferenceDef J1939RmAckmTxPduRef ⌈

Parameter Name	J1939RmAckmTxPduRef
Parent Container	J1939RmAckmTxPdu
Upstream mapping defined	–
Description	Reference to the Pdu object representing the I-PDU.
Multiplicity	1
Type	Reference to Pdu
Pre-Compile only	true

⌋

10.1.7 J1939RmRqstRxPdu

[ECUC_J1939Rm_00013] Definition of EcucParamConfContainerDef J1939RmRqstRxPdu ⌈

Container Name	J1939RmRqstRxPdu
Parent Container	J1939RmChannel
Upstream mapping defined	–
Description	Contains the configuration of the I-PDU used to receive the Request PG. This PDU consumes a meta data item of type CAN_ID_32.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmRqstRxPduId	1	[ECUC_J1939Rm_00020]
J1939RmRqstRxPduRef	1	[ECUC_J1939Rm_00021]

No Included Containers

⌋

[ECUC_J1939Rm_00020] Definition of EcucIntegerParamDef J1939RmRqstRxPduId ⌈

Parameter Name	J1939RmRqstRxPduId	
Parent Container	J1939RmRqstRxPdu	
Upstream mapping defined	–	
Description	The I-PDU identifier used for RxIndication from PduR.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default value	–	
With Auto	true	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00021] Definition of EcucReferenceDef J1939RmRqstRxPduRef ⌈

Parameter Name	J1939RmRqstRxPduRef
Parent Container	J1939RmRqstRxPdu
Upstream mapping defined	–
Description	Reference to the Pdu object representing the I-PDU.
Multiplicity	1
Type	Reference to Pdu

▽

△

Pre-Compile only	true

⌋

10.1.8 J1939RmRqstTxPdu

[ECUC_J1939Rm_00014] Definition of EcucParamConfContainerDef J1939RmRqstTxPdu ⌈

Container Name	J1939RmRqstTxPdu
Parent Container	J1939RmChannel
Upstream mapping defined	–
Description	Contains the configuration of the I-PDU used to transmit the Request PG. This PDU produces a meta data item of type CAN_ID_32.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmRqstTxPduId	1	[ECUC_J1939Rm_00022]
J1939RmRqstTxPduRef	1	[ECUC_J1939Rm_00023]

No Included Containers

⌋

[ECUC_J1939Rm_00022] Definition of EcucIntegerParamDef J1939RmRqstTxPduId ⌈

Parameter Name	J1939RmRqstTxPduId	
Parent Container	J1939RmRqstTxPdu	
Upstream mapping defined	–	
Description	The I-PDU identifier used for TxConfirmation from PduR.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default value	–	
With Auto	true	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00023] Definition of EcucReferenceDef J1939RmRqstTxPdu Ref ⌈

Parameter Name	J1939RmRqstTxPduRef
Parent Container	J1939RmRqstTxPdu
Upstream mapping defined	–
Description	Reference to the Pdu object representing the I-PDU.
Multiplicity	1
Type	Reference to Pdu
Pre-Compile only	true

⌋

10.1.9 J1939RmRqst2RxPdu

[ECUC_J1939Rm_00075] Definition of EcucParamConfContainerDef J1939Rm Rqst2RxPdu ⌈

Container Name	J1939RmRqst2RxPdu
Parent Container	J1939RmChannel
Upstream mapping defined	[24, CP_TPS_SystemTemplate]
Description	Contains the configuration of the I-PDU used to receive the Request2 PG. This PDU consumes a meta data item of type CAN_ID_32.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmRqst2RxPduId	1	[ECUC_J1939Rm_00078]
J1939RmRqst2RxPduRef	1	[ECUC_J1939Rm_00077]

No Included Containers

⌋

[ECUC_J1939Rm_00078] Definition of EcucIntegerParamDef J1939RmRqst2Rx PduId ⌈

Parameter Name	J1939RmRqst2RxPduId	
Parent Container	J1939RmRqst2RxPdu	
Upstream mapping defined	–	
Description	The I-PDU identifier used for RxIndication from PduR.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	

▽

△

Default value	–
With Auto	true
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00077] Definition of EcucReferenceDef J1939RmRqst2RxPduRef ⌈

Parameter Name	J1939RmRqst2RxPduRef
Parent Container	J1939RmRqst2RxPdu
Upstream mapping defined	–
Description	Reference to the Pdu object representing the I-PDU.
Multiplicity	1
Type	Reference to Pdu
Pre-Compile only	true

⌋

10.1.10 J1939RmRqst2TxPdu

[ECUC_J1939Rm_00076] Definition of EcucParamConfContainerDef J1939RmRqst2TxPdu ⌈

Container Name	J1939RmRqst2TxPdu
Parent Container	J1939RmChannel
Upstream mapping defined	[24, CP_TPS_SystemTemplate]
Description	Contains the configuration of the I-PDU used to transmit the Request2 PG. This PDU produces a meta data item of type CAN_ID_32.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmRqst2TxPduId	1	[ECUC_J1939Rm_00080]
J1939RmRqst2TxPduRef	1	[ECUC_J1939Rm_00079]

No Included Containers

⌋

[ECUC_J1939Rm_00080] Definition of EcucIntegerParamDef J1939RmRqst2TxPduId ⌈

Parameter Name	J1939RmRqst2TxPduId	
Parent Container	J1939RmRqst2TxPdu	
Upstream mapping defined	–	
Description	The I-PDU identifier used for TxConfirmation from PduR.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default value	–	
With Auto	true	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00079] Definition of EcucReferenceDef J1939RmRqst2TxPduRef ⌈

Parameter Name	J1939RmRqst2TxPduRef
Parent Container	J1939RmRqst2TxPdu
Upstream mapping defined	–
Description	Reference to the Pdu object representing the I-PDU.
Multiplicity	1
Type	Reference to Pdu
Pre-Compile only	true

⌋

10.1.11 J1939RmNode

[ECUC_J1939Rm_00049] Definition of EcucParamConfContainerDef J1939RmNode ⌈

Container Name	J1939RmNode	
Parent Container	J1939RmConfigSet	
Upstream mapping defined	[24, CP_TPS_SystemTemplate]	
Description	Contains the parameters for the support of a logical J1939 node (identified by an ECU address).	
Multiplicity	1..*	
Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmNmNodeRef	1	[ECUC_J1939Rm_00005]
J1939RmNodeChannelRef	1..*	[ECUC_J1939Rm_00052]

Included Containers		
Container Name	**Multiplicity**	**Description**
J1939RmUser	1..*	Contains the configuration of a module that uses the request and acknowledgement interfaces of J1939Rm.

⌋

[ECUC_J1939Rm_00005] Definition of EcucReferenceDef J1939RmNmNodeRef ⌈

Parameter Name	J1939RmNmNodeRef
Parent Container	J1939RmNode
Upstream mapping defined	–
Description	Reference to the corresponding J1939Nm node.
Multiplicity	1
Type	Symbolic name reference to J1939NmNode
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00052] Definition of EcucReferenceDef J1939RmNodeChannelRef ⌈

Parameter Name	J1939RmNodeChannelRef
Parent Container	J1939RmNode
Upstream mapping defined	[24, CP_TPS_SystemTemplate]
Description	Reference to the channels this node has access to.
Multiplicity	1..*
Type	Reference to J1939RmChannel
Pre-Compile only	true

⌋

10.1.12 J1939RmUser

[ECUC_J1939Rm_00010] Definition of EcucChoiceContainerDef J1939RmUser ⌈

Choice Container Name	J1939RmUser	
Parent Container	J1939RmNode	
Upstream mapping defined	[24, CP_TPS_SystemTemplate]	
Description	Contains the configuration of a module that uses the request and acknowledgement interfaces of J1939Rm.	
Multiplicity	1..*	
Post-Build Multiplicity	Loadable	Selectable

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Description**
J1939RmCddUser	0..1	J1939Rm User representing a complex driver (CDD). CDDs may use all services provided by J1939Rm.
J1939RmComUser	0..1	J1939Rm User representing AUTOSAR COM. Supports requests for COM I-PDUs.
J1939RmDcmUser	0..1	J1939Rm User representing the J1939Dcm. Requires request indication and transmission of acknowledgement.
J1939RmNmUser	0..1	J1939Rm User representing the J1939Nm. Requires request indication.
J1939RmRteUser	0..1	J1939Rm User representing an application software component (SW-C). SW-Cs may use all services provided by the J1939Rm via service ports.

⌋

10.1.13 J1939RmNmUser

[ECUC_J1939Rm_00071] Definition of EcucParamConfContainerDef J1939Rm NmUser ⌈

Container Name	J1939RmNmUser	
Parent Container	J1939RmUser	
Upstream mapping defined	–	
Description	J1939Rm User representing the J1939Nm. Requires request indication.	
Multiplicity	0..1	
Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

No Included Parameters

No Included Containers

⌋

10.1.14 J1939RmDcmUser

[ECUC_J1939Rm_00068] Definition of EcucParamConfContainerDef J1939Rm DcmUser ⌈

Container Name	J1939RmDcmUser
Parent Container	J1939RmUser
Upstream mapping defined	–
Description	J1939Rm User representing the J1939Dcm. Requires request indication and transmission of acknowledgement.
Multiplicity	0..1

▽

△

Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmUserId	1	[ECUC_J1939Rm_00072]
J1939RmUserRequestPGN	0..*	[ECUC_J1939Rm_00070]

No Included Containers

⌋

[ECUC_J1939Rm_00072] Definition of EcucIntegerParamDef J1939RmUserId ⌈

Parameter Name	J1939RmUserId	
Parent Container	J1939RmDcmUser	
Upstream mapping defined	–	
Description	Identifier used by J1939Dcm when calling J1939Rm_SendAck.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 254	
Default value	–	
With Auto	true	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00070] Definition of EcucIntegerParamDef J1939RmUserRequestPGN ⌈

Parameter Name	J1939RmUserRequestPGN	
Parent Container	J1939RmDcmUser	
Upstream mapping defined	[24, CP_TPS_SystemTemplate]	
Description	PGN of DMx PG supported by J1939Dcm.	
Multiplicity	0..*	
Type	EcucIntegerParamDef	
Range	0 .. 262143	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

10.1.15 J1939RmCddUser

[ECUC_J1939Rm_00066] Definition of EcucParamConfContainerDef J1939RmCddUser ⌈

Container Name	J1939RmCddUser
Parent Container	J1939RmUser
Upstream mapping defined	–
Description	J1939Rm User representing a complex driver (CDD). CDDs may use all services provided by J1939Rm.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
J1939RmUserAckIndication	1	[ECUC_J1939Rm_00028]
J1939RmUserAckPGN	0..*	[ECUC_J1939Rm_00061]
J1939RmUserId	0..1	[ECUC_J1939Rm_00025]
J1939RmUserRequestIndication	1	[ECUC_J1939Rm_00027]
J1939RmUserRequestPGN	0..*	[ECUC_J1939Rm_00026]
J1939RmUserSendAck	1	[ECUC_J1939Rm_00030]
J1939RmUserSendRequest	1	[ECUC_J1939Rm_00029]
J1939RmUserTimeoutSupervision	1	[ECUC_J1939Rm_00031]
J1939RmUserCddRef	1	[ECUC_J1939Rm_00042]

No Included Containers

⌋

[ECUC_J1939Rm_00028] Definition of EcucBooleanParamDef J1939RmUserAck Indication ⌈

Parameter Name	J1939RmUserAckIndication
Parent Container	J1939RmCddUser, J1939RmRteUser
Upstream mapping defined	–
Description	Enable AckIndication for this module. In case of CDD, the name is <apiServicePrefix>_ AckIndication. In case of RTE, the operation AckIndication of the required port J1939 Rm_AckIndication_{user} is called.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00061] Definition of EcucIntegerParamDef J1939RmUserAck PGN ⌈

Parameter Name	J1939RmUserAckPGN
Parent Container	J1939RmCddUser, J1939RmRteUser
Upstream mapping defined	–

▽

△

Description	PGN supported to be acknowledged to this module. The PGNs supported by different modules should usually be disjunctive.	
Multiplicity	0..*	
Type	EcucIntegerParamDef	
Range	0 .. 262143	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00025] Definition of EcucIntegerParamDef J1939RmUserId ⌈

Parameter Name	J1939RmUserId	
Parent Container	J1939RmCddUser, J1939RmRteUser	
Upstream mapping defined	–	
Description	Identifier used by a module using J1939Rm. This parameter is only required when the module uses transmission of requests or acknowledgements.	
Multiplicity	0..1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 254	
Default value	–	
With Auto	true	
Pre-Compile only	true	

⌋

[ECUC_J1939Rm_00027] Definition of EcucBooleanParamDef J1939RmUserRequestIndication ⌈

Parameter Name	J1939RmUserRequestIndication
Parent Container	J1939RmCddUser, J1939RmRteUser
Upstream mapping defined	–
Description	Enable RequestIndication for this module. In case of J1939Nm or J1939Dcm, the name is fixed. In case of CDD, the name is <apiServicePrefix>_RequestIndication. In case of RTE, J1939Rm will call the operation RequestIndication of the required port J1939Rm_RequestIndication_{user}.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00026] Definition of EcucIntegerParamDef J1939RmUserRequestPGN ⌈

Parameter Name	J1939RmUserRequestPGN
Parent Container	J1939RmCddUser, J1939RmRteUser
Upstream mapping defined	[24, CP_TPS_SystemTemplate]

▽

△

Description	PGN supported to be requested from this module. The PGNs supported by different modules should usually be disjunctive.
Multiplicity	0..*
Type	EcucIntegerParamDef
Range	0 .. 262143
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00030] Definition of EcucBooleanParamDef J1939RmUser SendAck ⌈

Parameter Name	J1939RmUserSendAck
Parent Container	J1939RmCddUser, J1939RmRteUser
Upstream mapping defined	–
Description	Enable the SendAck API for this module. In case of RTE, the operation SendAck of the provided port J1939Rm_SendAck_{user} is called.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00029] Definition of EcucBooleanParamDef J1939RmUser SendRequest ⌈

Parameter Name	J1939RmUserSendRequest
Parent Container	J1939RmCddUser, J1939RmRteUser
Upstream mapping defined	–
Description	Enable the SendRequest API for this module. In case of RTE, the operation Send Request of the provided port J1939Rm_SendRequest_{user} is called.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00031] Definition of EcucBooleanParamDef J1939RmUser TimeoutSupervision ⌈

Parameter Name	J1939RmUserTimeoutSupervision
Parent Container	J1939RmCddUser, J1939RmRteUser
Upstream mapping defined	–

▽

△

Description	Enable RequestTimeoutIndication and CancelRequestTimeout for this module. RequestTimeoutIndication: In case of CDD, the name is <apiServicePrefix>_Request TimeoutIndication. In case of RTE, the operation RequestTimeoutIndication of the required port J1939Rm_RequestTimeoutIndication_{user} is called. CancelRequestTimeout: In case of RTE, the operation CancelRequestTimeout of the provided port J1939Rm_CancelRequestTimeout_{user} is called.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_J1939Rm_00042] Definition of EcucForeignReferenceDef J1939RmUserCddRef ⌈

Parameter Name	J1939RmUserCddRef
Parent Container	J1939RmCddUser
Upstream mapping defined	–
Description	Reference to the CDD module description.
Multiplicity	1
Type	Foreign reference to ECUC-MODULE-CONFIGURATION-VALUES
Pre-Compile only	true

⌋

10.1.16 J1939RmRteUser

[ECUC_J1939Rm_00069] Definition of EcucParamConfContainerDef J1939RmRteUser ⌈

Container Name	J1939RmRteUser
Parent Container	J1939RmUser
Upstream mapping defined	–
Description	J1939Rm User representing an application software component (SW-C). SW-Cs may use all services provided by the J1939Rm via service ports.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmUserAckIndication	1	[ECUC_J1939Rm_00028]
J1939RmUserAckPGN	0..*	[ECUC_J1939Rm_00061]
J1939RmUserId	0..1	[ECUC_J1939Rm_00025]
J1939RmUserRequestIndication	1	[ECUC_J1939Rm_00027]

▽

△

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmUserRequestPGN	0..*	[ECUC_J1939Rm_00026]
J1939RmUserSendAck	1	[ECUC_J1939Rm_00030]
J1939RmUserSendRequest	1	[ECUC_J1939Rm_00029]
J1939RmUserTimeoutSupervision	1	[ECUC_J1939Rm_00031]

No Included Containers

⌋

For parameter table [ECUC_J1939Rm_00028] J1939RmUserAckIndication, see definition below container J1939RmCddUser.

For parameter table [ECUC_J1939Rm_00061] J1939RmUserAckPGN, see definition below container J1939RmCddUser.

For parameter table [ECUC_J1939Rm_00025] J1939RmUserId, see definition below container J1939RmCddUser.

For parameter table [ECUC_J1939Rm_00027] J1939RmUserRequestIndication, see definition below container J1939RmCddUser.

For parameter table [ECUC_J1939Rm_00026] J1939RmUserRequestPGN, see definition below container J1939RmCddUser.

For parameter table [ECUC_J1939Rm_00030] J1939RmUserSendAck, see definition below container J1939RmCddUser.

For parameter table [ECUC_J1939Rm_00029] J1939RmUserSendRequest, see definition below container J1939RmCddUser.

For parameter table [ECUC_J1939Rm_00031] J1939RmUserTimeoutSupervision, see definition below container J1939RmCddUser.

10.1.17 J1939RmComUser

[ECUC_J1939Rm_00067] Definition of EcucParamConfContainerDef J1939RmComUser ⌈

Container Name	J1939RmComUser	
Parent Container	J1939RmUser	
Upstream mapping defined	–	
Description	J1939Rm User representing AUTOSAR COM. Supports requests for COM I-PDUs.	
Multiplicity	0..1	
Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
J1939RmComIPdu	0..*	Contains the configuration of an I-PDU that is to be transmitted on request by COM.

⌋

10.1.18 J1939RmComIPdu

[ECUC_J1939Rm_00032] Definition of EcucParamConfContainerDef J1939RmComIPdu ⌈

Container Name	J1939RmComIPdu	
Parent Container	J1939RmComUser	
Upstream mapping defined	–	
Description	Contains the configuration of an I-PDU that is to be transmitted on request by COM.	
Multiplicity	0..*	
Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
J1939RmComIPduExtId1	0..1	[ECUC_J1939Rm_00081]
J1939RmComIPduExtId2	0..1	[ECUC_J1939Rm_00082]
J1939RmComIPduExtId3	0..1	[ECUC_J1939Rm_00083]
J1939RmComIPduPGN	1	[ECUC_J1939Rm_00033]
J1939RmComIPduRef	1	[ECUC_J1939Rm_00065]

No Included Containers

⌋

[ECUC_J1939Rm_00081] Definition of EcucIntegerParamDef J1939RmComIPduExtId1 ⌈

Parameter Name	J1939RmComIPduExtId1	
Parent Container	J1939RmComIPdu	
Upstream mapping defined	–	
Description	First extended identifier byte of the COM I-PDU.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_J1939Rm_00082] Definition of EcucIntegerParamDef J1939RmComIPdu ExtId2 ⌈

Parameter Name	J1939RmComIPduExtId2	
Parent Container	J1939RmComIPdu	
Upstream mapping defined	–	
Description	Second extended identifier byte of the COM I-PDU.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_J1939Rm_00083] Definition of EcucIntegerParamDef J1939RmComIPdu ExtId3 ⌈

Parameter Name	J1939RmComIPduExtId3	
Parent Container	J1939RmComIPdu	
Upstream mapping defined	–	
Description	Third extended identifier byte of the COM I-PDU.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_J1939Rm_00033] Definition of EcucIntegerParamDef J1939RmComIPdu PGN ⌈

Parameter Name	J1939RmComIPduPGN	
Parent Container	J1939RmComIPdu	
Upstream mapping defined	[24, CP_TPS_SystemTemplate]	
Description	PGN of the COM I-PDU.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 262143	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_J1939Rm_00065] Definition of EcucReferenceDef J1939RmComIPduRef

⌈

Parameter Name	J1939RmComIPduRef	
Parent Container	J1939RmComIPdu	
Upstream mapping defined	–	
Description	Reference to the Pdu object representing the I-PDU.	
Multiplicity	1	
Type	Reference to Pdu	
Post-Build Value	Loadable	-

⌋

A Not Applicable Requirements

[SWS_J1939Rm_NA_00001] Requirements Not Applicable to this Specification

Upstream requirements: SRS_BSW_00168, SRS_BSW_00170, SRS_BSW_00375, SRS_BSW_00383, SRS_BSW_00384, SRS_BSW_00388, SRS_BSW_00389, SRS_BSW_00390, SRS_BSW_00392, SRS_BSW_00393, SRS_BSW_00395, SRS_BSW_00403, SRS_BSW_00416, SRS_BSW_00417, SRS_BSW_00419, SRS_BSW_00422, SRS_BSW_00425, SRS_BSW_00469, SRS_BSW_00471, SRS_BSW_00472, SRS_BSW_00490, SRS_BSW_00491

⌈These requirements are not applicable to this specification.⌋

[SWS_J1939Rm_NA_00002] Only One Main Function

Upstream requirements: SRS_BSW_00432

⌈The `SAE J1939 Request Manager` does not have much to do in its main function, therefore a single main function is sufficient.⌋

B Change History of AUTOSAR Traceable Items

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

B.1 Traceable Item History of this Document According to AUTOSAR Release R26-11

B.1.1 Added Specification Items in R26-11

none

B.1.2 Changed Specification Items in R26-11

[ECUC_J1939Rm_00001] [SWS_J1939Rm_00035] [SWS_J1939Rm_00059] [SWS_J1939Rm_00079]

B.1.3 Deleted Specification Items in R26-11

[ECUC_J1939Rm_00002] [SWS_J1939Rm_00039]

B.2 Traceable Item History of this Document According to AUTOSAR Release R25-11

B.2.1 Added Specification Items in R25-11

none

B.2.2 Changed Specification Items in R25-11

none

B.2.3 Deleted Specification Items in R25-11

none

B.3 Traceable Item History of this Document According to AUTOSAR Release R24-11

B.3.1 Added Specification Items in R24-11

none

B.3.2 Changed Specification Items in R24-11

[SWS_J1939Rm_00025] [SWS_J1939Rm_00113] [SWS_J1939Rm_00126]

B.3.3 Deleted Specification Items in R24-11

none

B.4 Traceable Item History of this Document According to AUTOSAR Release R23-11

B.4.1 Added Specification Items in R23-11

none

B.4.2 Changed Specification Items in R23-11

[SWS_J1939Rm_00029] [SWS_J1939Rm_00055] [SWS_J1939Rm_00057] [SWS_J1939Rm_00063] [SWS_J1939Rm_00064] [SWS_J1939Rm_00065] [SWS_J1939Rm_00105] [SWS_J1939Rm_00118] [SWS_J1939Rm_91000]

B.4.3 Deleted Specification Items in R23-11

none

B.5 Traceable Item History of this Document According to AUTOSAR Release R22-11

B.5.1 Added Specification Items in R22-11

[SWS_J1939Rm_NA]

B.5.2 Changed Specification Items in R22-11

none

B.5.3 Deleted Specification Items in R22-11

none

B.6 Traceable Item History of this Document According to AUTOSAR Release R21-11

B.6.1 Added Specification Items in R21-11

none

B.6.2 Changed Specification Items in R21-11

[SWS_J1939Rm_00033] [SWS_J1939Rm_00118] [SWS_J1939Rm_00124] [SWS_J1939Rm_00127] [SWS_J1939Rm_00128] [SWS_J1939Rm_00129]

B.6.3 Deleted Specification Items in R21-11

none