## Document Title
Specification of Network Management for SAE J1939

<table>
<thead>
<tr>
<th>Document Owner</th>
<th>AUTOSAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Responsibility</td>
<td>AUTOSAR</td>
</tr>
<tr>
<td>Document Identification No</td>
<td>612</td>
</tr>
</tbody>
</table>

| Document Status       | Final                    |
| Part of AUTOSAR Standard | Classic Platform          |
| Part of Standard Release | 4.4.0                   |

## Document Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Release</th>
<th>Changed by</th>
<th>Change Description</th>
</tr>
</thead>
</table>
| 2018-10-31 | 4.4.0   | AUTOSAR Release Management | • Changed header file structure  
                          • Harmonized J1939NM_E_UNINIT                                                  |
| 2017-12-08 | 4.3.1   | AUTOSAR Release Management | • Fixed include hierarchy  
                          • Removed multiple configuration container                                        |
| 2016-11-30 | 4.3.0   | AUTOSAR Release Management | • Address coordination for gateways  
                          • Improved handling of meta data  
                          • Reliable TxConfirmation replaces timeout  
                          • SetEventStatus replaces ReportErrorStatus                                      |
| 2015-07-31 | 4.2.2   | AUTOSAR Release Management | • Support for networks without address claiming  
                          • Fixed state notifications to BswM  
                          • Introduction of further error classes                                           |
| 2014-10-31 | 4.2.1   | AUTOSAR Release Management | • Enhanced description of extended production error  
                          J1939NM_E_ADDRESS_LOST  
                          • Fixed usage of ‘const’ in NM APIs  
                          • Harmonized with SWS BSW General                                                   |
| 2014-03-31 | 4.1.3   | AUTOSAR Release Management | • Introduction of random delays  
                          • Fixed state diagram  
                          • Removed configurable dummy APIs                                                 |
| 2013-10-31 | 4.1.2   | AUTOSAR Release Management | • Additional development errors for function parameter checks  
                          • Removed change documentation                                                   |
## Document Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Release</th>
<th>Changed by</th>
<th>Change Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-03-15</td>
<td>4.1.1</td>
<td>AUTOSAR Administration</td>
<td>• Initial Release</td>
</tr>
</tbody>
</table>
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** ............................................................ 6
   1.1 Network Management according to SAE J1939 .............................................. 6
   1.2 J1939 Network Management BSW Module ................................................. 6

2. **Acronyms and abbreviations** ........................................................................ 7

3. **Related documentation** .................................................................................. 8
   3.1 Input documents .............................................................................................. 8
   3.2 Related standards and norms ......................................................................... 9
   3.3 Related specification ..................................................................................... 9

4. **Constraints and assumptions** ......................................................................... 10
   4.1 Limitations .................................................................................................... 10
   4.2 Applicability to car domains ......................................................................... 10

5. **Dependencies to other modules** ................................................................... 11
   5.1 File structure .................................................................................................. 11
      5.1.1 Code file structure .................................................................................. 11
      5.1.2 Header file structure ............................................................................. 11

6. **Requirements traceability** ............................................................................. 12

7. **Functional specification** ................................................................................ 13
   7.1 Overview ........................................................................................................ 13
   7.2 Module Handling ............................................................................................ 13
      7.2.1 Initialization ............................................................................................ 13
      7.2.2 Timing Related Functionality .................................................................. 13
   7.3 Network Management States of the J1939Nm .............................................. 14
      7.3.1 ECU Startup ............................................................................................ 15
      7.3.2 Address Loss ........................................................................................... 16
      7.3.3 ECU Shutdown ........................................................................................ 17
   7.4 Transmission of AddressClaimed ................................................................. 17
   7.5 Reception of AddressClaimed .......................................................................... 18
   7.6 Request for AddressClaimed .......................................................................... 19
   7.7 Address Coordination .................................................................................... 19
   7.8 Error Classification ....................................................................................... 20
      7.8.1 Development Errors ................................................................................ 20
      7.8.2 Runtime Errors ....................................................................................... 21
      7.8.3 Transient Faults ...................................................................................... 21
      7.8.4 Production Errors ................................................................................... 21
      7.8.5 Extended Production Errors .................................................................... 21
   7.9 API Parameter Checking .............................................................................. 22

8. **API specification** ............................................................................................ 23
   8.1 Imported types ............................................................................................... 23
   8.2 Type definitions ............................................................................................. 23
      8.2.1 J1939Nm_ConfigType ............................................................................ 23
   8.3 Function definitions ....................................................................................... 23
      8.3.1 J1939Nm_Init ......................................................................................... 23
      8.3.2 J1939Nm_DeInit .................................................................................... 24
8.3.3  J1939Nm_GetVersionInfo ................................................................. 24
8.3.4  J1939Nm_NetworkRequest ............................................................... 25
8.3.5  J1939Nm_NetworkRelease ............................................................... 25
8.3.6  J1939Nm_GetState ............................................................................. 26
8.3.7  J1939Nm_GetBusOffDelay ................................................................. 26
8.3.8  J1939Nm_PassiveStartUp ................................................................. 27
8.4   Call-back notifications ...................................................................... 27
     8.4.1  J1939Nm_RxIndication .................................................................. 27
     8.4.2  J1939Nm_TxConfirmation ............................................................. 28
     8.4.3  J1939Nm_RequestIndication ........................................................... 28
8.5   Scheduled functions .......................................................................... 29
     8.5.1  J1939Nm_MainFunction .................................................................. 29
8.6   Expected Interfaces ............................................................................ 30
     8.6.1  Mandatory Interfaces ..................................................................... 30
     8.6.2  Optional Interfaces ......................................................................... 30
     8.6.3  Configurable interfaces .................................................................. 31
9     Sequence diagrams ............................................................................. 32
     9.1   Transmission of AddressClaimed ..................................................... 32
     9.2   Reception of AddressClaimed ............................................................ 32
     9.3   Request for AddressClaimed ............................................................... 33
10  Configuration specification ................................................................. 34
    10.1  Containers and configuration parameters ........................................... 34
           10.1.1  J1939Nm ............................................................................... 38
           10.1.2  J1939NmGeneral ................................................................. 39
           10.1.3  J1939NmConfigSet ............................................................... 41
           10.1.4  J1939NmSharedAddressSpace ............................................... 41
           10.1.5  J1939NmChannel ................................................................. 42
           10.1.6  J1939NmTxPdu ..................................................................... 43
           10.1.7  J1939NmRxPdu ..................................................................... 44
           10.1.8  J1939NmNodeSpecificDemEventParameterRefs ................. 45
           10.1.9  J1939NmNode ................................................................. 45
           10.1.10 J1939NmExternalNode .......................................................... 49
    10.2  Configuration of NM Interface ......................................................... 53
    10.3  Published Information .................................................................... 53
1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module J1939 Network Management.

1.1 Network Management according to SAE J1939

In contrast to other AUTOSAR network management approaches, the task of J1939 network management is not to handle sleep and wake-up of ECUs, but to assign a unique address to each ECU. This is achieved by sending the AddressClaimed (AC, 0x0EE00) parameter group (PG) at start-up, which announces the desired address. If another ECU claims the same address, and has higher priority, the ECU has to go silent after sending the CannotClaimAddress parameter group (AC with null address 0xFE as source address). The AddressClaimed PG must also be sent upon request.

1.2 J1939 Network Management BSW Module

The J1939 Network Management module (J1939Nm) handles received and transmitted AddressClaimed (AC) PGs. It supports transmission of AC on start-up, after a contending AC received from another node, and on request (triggered by the J1939 Request Manager).

Besides this, the J1939 Network Management module also ensures that the ECU does not send any messages during startup or after address loss.
2 Acronyms and abbreviations

<table>
<thead>
<tr>
<th>Abbreviation / Acronym:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>J1939 AddressClaimed PG (PGN = 0x0EE00)</td>
</tr>
<tr>
<td>BSW</td>
<td>Basic Software (module)</td>
</tr>
<tr>
<td>DET</td>
<td>Default Error Tracer, supports development and runtime error reporting</td>
</tr>
<tr>
<td>Node</td>
<td>J1939 node – can be attached to more than one channel</td>
</tr>
<tr>
<td>NodeChannel</td>
<td>The connection of a node to one channel</td>
</tr>
<tr>
<td>PG</td>
<td>Parameter Group</td>
</tr>
<tr>
<td>PGN</td>
<td>Parameter Group Number</td>
</tr>
<tr>
<td>RQST</td>
<td>J1939 Request PG (PGN = 0x0EA00)</td>
</tr>
</tbody>
</table>
3  Related documentation

3.1  Input documents

[1] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList.pdf

[2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

AUTOSAR_SRS_BSWGeneral.pdf

AUTOSAR_SWS_BSWGeneral.pdf

AUTOSAR_SRS_J1939.pdf

[6] Requirements on Network Management
AUTOSAR_SRS_J1939.pdf

AUTOSAR_SWS_CommunicationStackTypes.pdf

[8] System Template
AUTOSAR_TPS_SystemTemplate.pdf

AUTOSAR_SWS_CANInterface.pdf

AUTOSAR_SWS_NetworkManagementInterface.pdf

AUTOSAR_SWS_BSWModeManager.pdf

AUTOSAR_SWS_SAEJ1939RequestManager.pdf

AUTOSAR_SWS_DefaultErrorTracer.pdf

AUTOSAR_SWS_DiagnosticEventManager.pdf

AUTOSAR_SWS_BSWScheduler.pdf
3.2 Related standards and norms


3.3 Related specification

AUTOSAR provides a General Specification on Basic Software modules [4] (SWS BSW General), which is also valid for the SAE J1939 Network Management module.

Thus, the specification SWS BSW General shall be considered as additional and required specification for SAE J1939 Transport Layer.
4 Constraints and assumptions

4.1 Limitations

The J1939 Network Management module does not support changing of the address, neither after a CommandedAddress PG, nor after address loss. It also does not support for Name Management.

4.2 Applicability to car 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 J1939 Network Management module (J1939Nm) has interfaces towards the CAN Interface (CanIf), the J1939 Request Manager (J1939Rm), the Network Management Interface (Nm), the Diagnostic Event Manager (DEM), and the Default Error Tracer (DET).

The J1939 Network Management module includes header files of the CAN Interface, the Network Management Interface, the J1939 Request Manager, the Diagnostic Event Manager, and the Default Error Tracer.

5.1 File structure

5.1.1 Code file structure

For details, refer to the section 5.1.6 "Code file structure" of the SWS BSW General [4].

5.1.2 Header file structure

For details, refer to the section 5.1.7 "Header file structure" of the SWS BSW General [4].
### Requirements traceability

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
<th>Satisfied by</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS_BSW_00407</td>
<td>Each BSW module shall provide a function to read out the version information of a dedicated module implementation</td>
<td>SWS_J1939Nm_00033</td>
</tr>
<tr>
<td>SRS_J1939_00030</td>
<td>The J1939 Network Management module shall provide an interface for module initialization</td>
<td>SWS_J1939Nm_00002, SWS_J1939Nm_00007, SWS_J1939Nm_00031</td>
</tr>
<tr>
<td>SRS_J1939_00031</td>
<td>The J1939 Network Management module shall provide an interface for module shutdown</td>
<td>SWS_J1939Nm_00003, SWS_J1939Nm_00032</td>
</tr>
<tr>
<td>SRS_J1939_00032</td>
<td>The J1939 Network Management module shall report a failed address claim to the Diagnostic Event Manager</td>
<td>SWS_J1939Nm_00012</td>
</tr>
<tr>
<td>SRS_J1939_00033</td>
<td>The J1939 Network Management module shall perform an initial address claim at startup</td>
<td>SWS_J1939Nm_00009, SWS_J1939Nm_00016, SWS_J1939Nm_00017, SWS_J1939Nm_00019, SWS_J1939Nm_00062</td>
</tr>
<tr>
<td>SRS_J1939_00034</td>
<td>The J1939 Network Management module shall react correctly to contending address claims</td>
<td>SWS_J1939Nm_00014, SWS_J1939Nm_00016, SWS_J1939Nm_00017, SWS_J1939Nm_00018, SWS_J1939Nm_00019, SWS_J1939Nm_00020, SWS_J1939Nm_00021, SWS_J1939Nm_00062</td>
</tr>
<tr>
<td>SRS_J1939_00035</td>
<td>The J1939 Network Management module shall react to requests for the AddressClaimed PG</td>
<td>SWS_J1939Nm_00016, SWS_J1939Nm_00017, SWS_J1939Nm_00018, SWS_J1939Nm_00019, SWS_J1939Nm_00022, SWS_J1939Nm_00023, SWS_J1939Nm_00043, SWS_J1939Nm_00062</td>
</tr>
<tr>
<td>SRS_J1939_00036</td>
<td>The J1939 Network Management module shall only allow communication after a successful address claim</td>
<td>SWS_J1939Nm_00010, SWS_J1939Nm_00011, SWS_J1939Nm_00015, SWS_J1939Nm_00021, SWS_J1939Nm_00044, SWS_J1939Nm_00045, SWS_J1939Nm_00063, SWS_J1939Nm_00064, SWS_J1939Nm_00065, SWS_J1939Nm_00066</td>
</tr>
<tr>
<td>SRS_J1939_00037</td>
<td>The J1939 Network Management module shall delay communication after initial address claim</td>
<td>SWS_J1939Nm_00010, SWS_J1939Nm_00013, SWS_J1939Nm_00061, SWS_J1939Nm_00063</td>
</tr>
</tbody>
</table>
7 Functional specification

This chapter defines the behavior of the J1939 Network Management module. The API of the module is defined in chapter 8, while the configuration is defined in chapter 10.

7.1 Overview

The J1939 Network Management module supports transmission and reception of AddressClaimed PGs, and handling of requests for the AddressClaimed PG. It also ensures that the ECU does not send messages during the initial address claiming phase or after the ECU sent a CannotClaimAddress PG because it lost its address to a contending address claim.

7.2 Module Handling

This section contains description of auxiliary functionality of the J1939 Network Management module.

7.2.1 Initialization

The J1939 Network Management module is initialized via J1939Nm_Init, and de-initialized via J1939Nm_DeInit. Except for J1939Nm_GetVersionInfo and J1939Nm_Init, the API functions of the J1939 Network Management module may only be called after the module has been properly initialized.

[SWS_J1939Nm_00002] [A call to J1939Nm_Init initializes all internal variables and sets the J1939 Network Management module to the initialized state.] (SRS_J1939_00030)

[SWS_J1939Nm_00003] [A call to J1939Nm_DeInit sets the J1939 Network Management module back to the uninitialized state.] (SRS_J1939_00031)

[SWS_J1939Nm_00005] [When J1939Nm_Init is called in initialized state, the J1939 Network Management module shall not re-initialize its internal variables. It shall instead call Det_ReportError with the error code J1939NM_E_REINIT if DET reporting is enabled via J1939NmDevErrorDetect.] ()

7.2.2 Timing Related Functionality

To be able to measure times, the J1939 Network Management module is triggered cyclically via the J1939Nm_MainFunction.

[SWS_J1939Nm_00006] [The J1939 Network Management module shall use the J1939Nm_MainFunction for timing related purposes.] ()

The recovery after a bus off must be delayed by a random time to avoid repeating bus offs when two nodes try to claim the same address. This random delay is also
required when sending a CannotClaimAddress PG after a contending address claim or after a request for the AddressClaimed PG.

[SWS_J1939Nm_00068] [The J1939Nm shall calculate a random number for delaying bus off recovery and transmission of a CannotClaimAddress PG. The calculation shall use the NAME of a node as seed.] ()

[SWS_J1939Nm_00069] [When J1939Nm_GetBusOffDelay is called, J1939Nm shall return a random number based on the NAMEs of all nodes attached to the reported channel. This random number gives the delay time, based on the ticktime configured via J1939NmBusOffDelayTickPeriod.] ()

### 7.3 Network Management States of the J1939Nm

While the NM Interface handles network management states on channel level, the J1939 Network Management module needs a finer granularity, because several nodes can be attached to each channel. The connection of a node to one channel is called NodeChannel hereafter.

The following picture shows the internal NM related states of the J1939 Network Management module for one of its NodeChannels (i.e. one channel of a single node), and the transitions between these states:

![Diagram of J1939Nm states](image)

**Figure 1: Internal states of J1939Nm with startup delay**

The J1939 Network Management module reports state changes to the NM Interface and to the Basic Software Mode Manager (BswM).
While the states reported to the NM Interface are accumulated states of all NodeChannels of a CAN channel, the J1939 Network Management module reports states to the BswM separately for each NodeChannel.

### 7.3.1 ECU Startup

The J1939 Network Management module starts all NodeChannels in ‘SleepMode’ (corresponding to NM_MODE_BUS_SLEEP). The CAN channels will be switched to ‘NetworkMode’ (corresponding to NM_MODE_NETWORK) immediately afterwards by a network request issued from the ComM via NM Interface.

[SWS_J1939Nm_00007] [During initialization via J1939Nm_Init, the J1939 Network Management module shall silently assume the ‘SleepMode’ for all NodeChannels.] (SRS_J1939_00030)

[SWS_J1939Nm_00009] [A call to J1939Nm_NetworkRequest shall set all NodeChannels of the reported channel to ‘NetworkMode’. The J1939 Network Management module shall notify this mode change to the NM Interface via Nm_NetworkMode, and shall trigger transmission of an AddressClaimed PG for each NodeChannel where J1939NmChannelUsesAddressArbitration is enabled.] (SRS_J1939_00033)

The transmission of the AddressClaimed PG is described in detail in section 7.4.

When entering the network mode, the behavior of the J1939 Network Management module depends on the configuration parameter J1939NmNodeStartupDelay. Controlled by this parameter, the J1939 Network Management module switches the state of the affected NodeChannels either to the sub state ‘Claiming’ of the state ‘Offline’ (corresponding to NM_STATE_OFFLINE), or to the state ‘NormalOperation’ (corresponding to NM_STATE_NORMAL_OPERATION).

[SWS_J1939Nm_00010] [If a node of the J1939 Network Management module is configured for deferred online state (J1939NmNodeStartUpDelay enabled), its NodeChannels shall enter the sub state ‘Claiming’ of the state ‘Offline’ immediately after the switch from ‘SleepMode’ to ‘NetworkMode’. The J1939 Network Management module shall report this state change to the Basic Software Mode Manager via BswM_J1939Nm_StateChangeNotification(NM_STATE_OFFLINE).] (SRS_J1939_00036, SRS_J1939_00037)

[SWS_J1939Nm_00011] [If a node of the J1939 Network Management module is configured for immediate online state (J1939NmNodeStartUpDelay disabled), its NodeChannels shall enter the state ‘NormalOperation’ immediately after the switch from ‘SleepMode’ to ‘NetworkMode’. The J1939 Network Management module shall report this state change to the Basic Software Mode Manager via BswM_J1939Nm_StateChangeNotification(NM_STATE_NORMAL_OPERATION).] (SRS_J1939_00036)

The NM Interface expects an accumulated channel state.
When all NodeChannels of a channel are configured for deferred online state (J1939NmNodeStartUpDelay enabled), the J1939 Network Management module shall report the state change of these NodeChannels to the ‘Offline’ state immediately to the NM Interface via Nm_StateChangeNotification(NM_STATE_OFFLINE).] (SRS_J1939_00036, SRS_J1939_00037)

When the first NodeChannel of a channel changes its state to ‘NormalOperation’, the J1939 Network Management module shall report this state change immediately to the NM Interface via Nm_StateChangeNotification(NM_STATE_NORMAL_OPERATION).] (SRS_J1939_00036)

When a NodeChannel has stayed for 250ms in state ‘Claiming’ after transmission of the initial AddressClaimed PG, it will switch to state ‘NormalOperation’.

When J1939Nm_TxConfirmation is called with result E_OK for the initial AddressClaimed PG of a NodeChannel (transmitted during the transition to the ‘Claiming’ sub state), the J1939 Network Management module shall start the delay timer for this NodeChannel.] (SRS_J1939_00037)

When the delay timer of a NodeChannel expires in sub state ‘Claiming’, the J1939 Network Management module shall switch that NodeChannel to state ‘NormalOperation’ and shall report this state change to the Basic Software Mode Manager via BswM_J1939Nm_StateChangeNotification(NM_STATE_NORMAL_OPERATION).] (SRS_J1939_00037)

### 7.3.2 Address Loss

When a node of the J1939 Network Management module loses its claimed address on one of its channels (see section 7.5), it will switch that NodeChannel to the sub state ‘AcLost’ of state ‘Offline’, notifying the NM Interface and the BswM of this state change and sending a CannotClaimAddress PG for the losing node on that channel (see section 7.4).

When a NodeChannel loses its address in ‘NetworkMode’, it shall switch to the sub state ‘AcLost’ of state ‘Offline’ and, after a delay calculated according to [SWS_J1939Nm_00068], trigger transmission of a CannotClaimAddress PG.] (SRS_J1939_00034)

When a NodeChannel switches from state ‘NormalOperation’ to the sub state ‘AcLost’ of state ‘Offline’, the J1939 Network Management module shall notify the Basic Software Mode Manager via BswM_J1939Nm_StateChangeNotification(NM_STATE_OFFLINE).] (SRS_J1939_00036)
[SWS_J1939Nm_00066] [When the last NodeChannel of a channel changes its state to ‘Offline’, the J1939 Network Management module shall report this state change immediately to the NM Interface via Nm_StateChangeNotification(NM_STATE_OFFLINE).] (SRS_J1939_00036)

7.3.3 ECU Shutdown

To shut down the network, ComM calls the Nm_NetworkRelease API of the NM Interface, which in turn calls J1939Nm_NetworkRelease. The J1939 Network Management module will then switch to ‘SleepMode’, and notify this to the NM Interface.

[SWS_J1939Nm_00015] [A call to J1939Nm_NetworkRelease shall set all NodeChannels of the reported channel to ‘SleepMode’. The J1939 Network Management module shall notify this mode change to the NM Interface via Nm_BusSleepMode, and shall report a state change to ‘SleepMode’ to the NM Interface via Nm_StateChangeNotification(NM_STATE_BUS_SLEEP) and to BswM via BswM_J1939Nm_StateChangeNotification(NM_STATE_BUS_SLEEP).] (SRS_J1939_00036)

7.4 Transmission of AddressClaimed

For each NodeChannel, the J1939 Network Management module needs to ensure that a contending AddressClaimed PG or a request for AddressClaimed is answered by at least one AddressClaimed PG. If an AddressClaimed PG is still pending for that NodeChannel, but now a CannotClaimAddress PG must be sent, it suffices to send the CannotClaimAddress. Therefore, a single buffer per NodeChannel that stores only the last transmission request is sufficient.

For the transmission of both the AddressClaimed and the CannotClaimAddress PG, the J1939 Network Management module uses just one PDU per channel with variable source address contained in the meta data of the PDU.

[SWS_J1939Nm_00016] [When the J1939 Network Management module needs to send an AddressClaimed (or CannotClaimAddress) PG, and no previous transmission is pending, it shall directly forward the corresponding PDU to the CAN Interface via CanIf_Transmit.] (SRS_J1939_00033, SRS_J1939_00034, SRS_J1939_00035)

[SWS_J1939Nm_00073] [The J1939 Network Management module shall use a meta data item of type CAN_ID_32 to provide the source address of transmitted AddressClaimed and CannotClaimAddress PGs. The source address resides in the last (least significant) byte of the meta data item.] ()

[SWS_J1939Nm_00017] [When the J1939 Network Management module needs to send an AddressClaimed (or CannotClaimAddress) PG, and the CAN Interface has not yet called J1939Nm_TxConfirmation for the previous transmission, the J1939 Network Management module shall buffer this PG for later transmission.] (SRS_J1939_00033, SRS_J1939_00034, SRS_J1939_00035)
[SWS_J1939Nm_00018] [Apart from the initial AddressClaimed PG, the J1939 Network Management module shall buffer only the latest AddressClaimed or CannotClaimAddress PG.] (SRS_J1939_00034, SRS_J1939_00035)

Rationale: The initial AddressClaimed PG must be transmitted before any CannotClaimAddress PG according to [18]. Otherwise, the J1939 Network Management module should report current state even if the original request preceded a state change.

[SWS_J1939Nm_00019] [A call to J1939Nm_TxConfirmation with result E_OK shall trigger transmission of a buffered AddressClaimed or CannotClaimAddress PG via CanIf_Transmit.] (SRS_J1939_00033, SRS_J1939_00034, SRS_J1939_00035)

[SWS_J1939Nm_00062] [When J1939Nm_TxConfirmation is called with result E_NOT_OK, the transmission of that PG shall be triggered again.] (SRS_J1939_00033, SRS_J1939_00034, SRS_J1939_00035)

7.5 Reception of AddressClaimed

The source address of received AddressClaimed PGs must be immediately compared to the source addresses of all NodeChannels attached to the same channel (see J1939NmNodePreferredAddress). If any of these match, the payload of the received PG must be compared to the configured NAME for the matching source address (see J1939NmNodeNameXxx), and depending on the relative priority, the J1939 Network Management module must send an AddressClaimed or a CannotClaimAddress PG. The priority is determined by the numerical value of the NAME.

To be able to identify the source address, the PDU associated with the AddressClaimed PG shall have a variable source address contained in the meta data of the PDU. In addition, the priority needs to be ignored for this PDU.

[SWS_J1939Nm_00074] [The J1939 Network Management module shall use a meta data item of type CAN_ID_32 to determine the source address of received AddressClaimed and CannotClaimAddress PGs. The source address resides in the last (least significant) byte of the meta data item.] ()

[SWS_J1939Nm_00020] [If J1939NmChannelUsesAddressArbitration is enabled, a call to J1939Nm_RxIndication indicating reception of an AddressClaimed PG with one of the source addresses configured via J1939NmNodePreferredAddress and a payload that has a higher numerical value than the NAME for this source address configured via J1939NmNodeNameXxx shall trigger transmission of an AddressClaimed PG for the according NodeChannel (see section 7.4).] (SRS_J1939_00034)

[SWS_J1939Nm_00021] [If J1939NmChannelUsesAddressArbitration is enabled, a call to J1939Nm_RxIndication indicating reception of an AddressClaimed PG with one of the source addresses configured via J1939NmNodePreferredAddress and a
payload that has a lower numerical value than the NAME for this source address configured via J1939NmNodeNameXxx shall induce a state change of the according NodeChannel to the sub state ‘AcLost’ of state ‘Offline’. (SRS_J1939_00034, SRS_J1939_00036)

The state change to ‘Offline’ will be notified to the NM Interface and the Basic Software Mode Manager and will trigger transmission of a CannotClaimAddress PG (see section 7.3.2).

Sometimes, the application needs to know the content of all address claimed messages on the bus, e.g. to build up a table that maps functions to addresses. The J1939 Network Management module shall support this use case via a generic call-out function (see section 8.6.3.1).

[SWS_J1939Nm_00060] [If enabled via J1939NmUserCallout, the J1939Nm shall forward the source address and the content of each AddressClaimed PG to the call-out function <User_AddressClaimedIndication> (see SWS_J1939Nm_00028).] ()

7.6 Request for AddressClaimed

When the J1939 Network Management module receives a request for the AddressClaimed PGN from the J1939 Request Manager, it will answer either with an AddressClaimed or with a CannotClaimAddress PG, depending on the current state (see below).

Independent of the request being global or specific, the transmitted PG is always global.

[SWS_J1939Nm_00022] [A call to J1939Nm_RequestIndication shall trigger transmission of an AddressClaimed PG when the addressed NodeChannel is in state ‘NormalOperation’ or sub state ‘Claiming’ of state ‘Offline’.] (SRS_J1939_00035)

[SWS_J1939Nm_00023] [A call to J1939Nm_RequestIndication shall trigger transmission of a CannotClaimAddress PG after a delay calculated according to [SWS_J1939Nm_00068] when the addressed NodeChannel is in sub state ‘AcLost’ of state ‘Offline’.] (SRS_J1939_00035)

The J1939Nm_RequestIndication will never be triggered in state ‘SleepMode’, because then no CAN messages can be received.

7.7 Address Coordination

The J1939 Network Management module is able to coordinate the addresses of different J1939 channels connected to a gateway, so that routed messages have valid addresses on every bus on which they appear.

There are two basic strategies:

1. Several J1939 channels form one common address space. In this scenario, the J1939 Network Management module replicates all AddressClaimed
messages appearing on one of the networks on all other networks of the same address space. Nodes connected via the gateway perform a direct arbitration of addresses.

2. Selected nodes of one channel appear also on one or more other channels. In this scenario, the J1939 Network Management Module claims the addresses of configured external nodes. Address arbitration is performed between the gateway and the nodes on one channel.

A single gateway can combine both strategies for different sets of channels. The main difference of the strategies is that addresses are not shared in the second strategy, and therefore more than 254 nodes can be present within one system.

[SWS_J1939Nm_00071] [If gateway support is enabled via J1939NmGatewaySupport, and the configuration contains a J1939NmSharedAddressSpace, the J1939Nm shall transmit all AddressClaimed messages received on one of the channels referenced by J1939NmSharedAddressSpace on all other channels referenced by the same J1939NmSharedAddressSpace.] ()

[SWS_J1939Nm_00072] [If gateway support is enabled via J1939NmGatewaySupport, and the configuration contains a J1939NmExternalNode, the channels referenced by J1939NmExternalNodeGatewayedChannelRef shall be treated like internal node channels, with the difference that the state transition from SleepMode to NetworkMode is triggered by the reception of an AddressClaimed message from the external node and enters NormalOperation immediately, and the transition to SleepMode is triggered by the reception of a CannotClaimAddress message from the same node.] ()

7.8 Error Classification

The J1939 Network Management module supports reporting of development, runtime, and extended production errors.

7.8.1 Development Errors

The supported development errors are defined in the following table.

[SWS_J1939Nm_00024] [Table of development errors used by the J1939 Network Management module:]

<table>
<thead>
<tr>
<th>Type or error</th>
<th>Related error code</th>
<th>Value [hex]</th>
</tr>
</thead>
<tbody>
<tr>
<td>An API was called while the module was uninitialized</td>
<td>J1939NM_E_UNINIT</td>
<td>0x01</td>
</tr>
<tr>
<td>The Init API was called twice</td>
<td>J1939NM_E_REINIT</td>
<td>0x02</td>
</tr>
<tr>
<td>J1939Nm_Init was called with an invalid configuration pointer</td>
<td>J1939NM_E_INIT_FAILED</td>
<td>0x03</td>
</tr>
<tr>
<td>An API service was called with a NULL pointer</td>
<td>J1939NM_E_PARAM_POINTER</td>
<td>0x10</td>
</tr>
</tbody>
</table>
An API service was called with a wrong ID
J1939NM_E_INVALID_PDU_SDU_ID 0x11

An API service was called with wrong network handle
J1939NM_E_INVALID_NETWORK_ID 0x12

An API was called with an unsupported PGN
J1939NM_E_INVALID_PGN 0x13

An API was called with an illegal priority
J1939NM_E_INVALID_PRIOR 0x14

An API was called with an illegal node address
J1939NM_E_INVALID_ADDRESS 0x15

An API was called with an illegal node ID
J1939NM_E_INVALID_NODE 0x16

Development error values are of type uint8.
}

7.8.2 Runtime Errors

Runtime errors have not yet been classified.

7.8.3 Transient Faults

There are no transient faults.

7.8.4 Production Errors

There are no production errors.

7.8.5 Extended Production Errors

Extended production errors are handled as events of the Diagnostic Event Manager. The event IDs are defined in the following tables, while the actual values are assigned externally by the configuration of the Diagnostic Event Manager, and are included in the J1939 Network Management module via Dem.h.

[SWS_J1939Nm_00012] [ |  

<table>
<thead>
<tr>
<th>Error Name:</th>
<th>J1939NM_E_ADDRESS_LOST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Description:</td>
<td>The desired address could not be claimed.</td>
</tr>
<tr>
<td>Long Description:</td>
<td>During start-up of the ECU, all J1939Nm nodes need to send an address claim to the bus and wait for acceptance of the claimed address. If another ECU claims the same address and has a higher priority, the ECU loses its address and stops communication. This is a critical problem, because J1939Nm was not specified for networks where this can happen.</td>
</tr>
<tr>
<td>Detection Criteria:</td>
<td>Fail When address claiming failed, because an AddressClaimed message with higher priority was received (see [SWS_J1939Nm_00021]), the J1939 Network Management module shall report the extended production error J1939NM_E_ADDRESS_LOST with</td>
</tr>
</tbody>
</table>
### 7.9 API Parameter Checking

The J1939 Network Management module performs parameter checks for all called APIs. It reports the development error J1939NM_E_INVALID_PDU_SDU_ID when a check of a PDU/SDU ID fails, J1939NM_E_INVALID_NETWORK_ID when a check of a network handle fails, and J1939NM_E_PARAM_POINTER when a call provides a NULL pointer.

[SWS_J1939Nm_00025] [If DET reporting is enabled via J1939NmDevErrorDetect, the J1939 Network Management module shall check PduIdType parameters (SDU/PDU IDs) of its API functions against the configured IDs, and shall report the development error J1939NM_E_INVALID_PDU_SDU_ID when an unknown ID is provided by the call.] ()

[SWS_J1939Nm_00026] [If DET reporting is enabled via J1939NmDevErrorDetect, the J1939 Network Management module shall check NetworkHandleType parameters (network handles) of its API functions against the referenced network handles of ComM, and shall report the development error J1939NM_E_INVALID_NETWORK_ID when an unknown handle is provided by the call.] ()

J1939NM_E_PARAM_POINTER shall be reported as specified in [4] by SWS_BSW_00212.
8 API specification

8.1 Imported types

In this section, all types used by the J1939 Network Management module are listed together with the defining module:

<table>
<thead>
<tr>
<th>Module</th>
<th>Header File</th>
<th>Imported Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComStack_Types</td>
<td>ComStackTypes.h</td>
<td>NetworkHandleType</td>
</tr>
<tr>
<td></td>
<td>ComStackTypes.h</td>
<td>PdulIdType</td>
</tr>
<tr>
<td></td>
<td>ComStackTypes.h</td>
<td>PduInfoType</td>
</tr>
<tr>
<td>Dem</td>
<td>Rte_Dem_Type.h</td>
<td>Dem_EventIdType</td>
</tr>
<tr>
<td></td>
<td>Rte_Dem_Type.h</td>
<td>Dem_EventStatusType</td>
</tr>
<tr>
<td>J1939Rm</td>
<td>Rte_J1939Rm_Type.h</td>
<td>J1939Rm_ExtldInfoType</td>
</tr>
<tr>
<td>Nm</td>
<td>NmStack_types.h</td>
<td>Nm_ModeType</td>
</tr>
<tr>
<td></td>
<td>NmStack_types.h</td>
<td>Nm_StateType</td>
</tr>
<tr>
<td>Std_Types</td>
<td>StandardTypes.h</td>
<td>Std_ReturnType</td>
</tr>
<tr>
<td></td>
<td>StandardTypes.h</td>
<td>Std_VersionTypeInfo</td>
</tr>
</tbody>
</table>

8.2 Type definitions

8.2.1 J1939Nm_ConfigType

This is the base type for the configuration of the J1939 Network Management module. A pointer to an instance of this structure will be used in the initialization of the J1939 Network Management module. The content of this structure is defined in chapter 10 Configuration specification.

Available via: J1939Nm.h

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 J1939Nm_Init

Service name: J1939Nm_Init
Syntax: void J1939Nm_Init();
### J1939Nm_Init

<table>
<thead>
<tr>
<th>Service name</th>
<th>J1939Nm_Init</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>void J1939Nm_Init(configPtr)</td>
</tr>
<tr>
<td>Service ID[hex]</td>
<td>0x01</td>
</tr>
<tr>
<td>Sync/Async</td>
<td>Synchronous</td>
</tr>
<tr>
<td>Reentrancy</td>
<td>Non Reentrant</td>
</tr>
<tr>
<td>Parameters (in)</td>
<td>configPtr</td>
</tr>
<tr>
<td>Parameters (inout)</td>
<td>None</td>
</tr>
<tr>
<td>Parameters (out)</td>
<td>None</td>
</tr>
<tr>
<td>Return value</td>
<td>None</td>
</tr>
<tr>
<td>Description</td>
<td>This function initializes the J1939 Network Management module.</td>
</tr>
<tr>
<td>Available via</td>
<td>J1939Nm.h</td>
</tr>
</tbody>
</table>

See section 7.2.1 for details.

See section 7.9 for parameter checks.

J1939NM_E_INIT_FAILED shall be reported as specified in [4] by SWS_BSW_00050.

### J1939Nm_DeInit

<table>
<thead>
<tr>
<th>Service name</th>
<th>J1939Nm_DeInit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>void J1939Nm_DeInit()</td>
</tr>
<tr>
<td>Service ID[hex]</td>
<td>0x02</td>
</tr>
<tr>
<td>Sync/Async</td>
<td>Synchronous</td>
</tr>
<tr>
<td>Reentrancy</td>
<td>Non Reentrant</td>
</tr>
<tr>
<td>Parameters (in)</td>
<td>None</td>
</tr>
<tr>
<td>Parameters (inout)</td>
<td>None</td>
</tr>
<tr>
<td>Parameters (out)</td>
<td>None</td>
</tr>
<tr>
<td>Return value</td>
<td>None</td>
</tr>
<tr>
<td>Description</td>
<td>This function resets the J1939 Network Management module to the uninitialized state.</td>
</tr>
<tr>
<td>Available via</td>
<td>J1939Nm.h</td>
</tr>
</tbody>
</table>

See section 7.2.1 for details.

### J1939Nm_GetVersionInfo

<table>
<thead>
<tr>
<th>Service name</th>
<th>J1939Nm_GetVersionInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>void J1939Nm_GetVersionInfo(versionInfo)</td>
</tr>
<tr>
<td>Service ID[hex]</td>
<td>0x03</td>
</tr>
</tbody>
</table>

8.3.3 J1939Nm_GetVersionInfo

<table>
<thead>
<tr>
<th>Service name</th>
<th>J1939Nm_GetVersionInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>void J1939Nm_GetVersionInfo(versionInfo)</td>
</tr>
<tr>
<td>Service ID[hex]</td>
<td>0x03</td>
</tr>
</tbody>
</table>
### 8.3.4 J1939Nm_NetworkRequest

**Service name:** J1939Nm_NetworkRequest  
**Syntax:**

```c
Std_ReturnType J1939Nm_NetworkRequest(
    NetworkHandleType nmChannelHandle
)
```

- **Service ID[hex]:** 0x05  
- **Sync/Async:** Synchronous  
- **Reentrancy:** Reentrant (but not for the same NM-Channel)  
- **Parameters (in):** `nmChannelHandle`  
- **Parameters (out):** None  
- **Return value:**  
  - `E_OK`: No error  
  - `E_NOT_OK`: Requesting of network has failed  
- **Description:** Request the network, since ECU needs to communicate on the bus.

**Available via:** `J1939Nm.h`

See section 7.3.1 for details.

See section 7.2.1 for error handling and section 7.9 for parameter checks.

### 8.3.5 J1939Nm_NetworkRelease

**Service name:** J1939Nm_NetworkRelease  
**Syntax:**

```c
Std_ReturnType J1939Nm_NetworkRelease(
    NetworkHandleType nmChannelHandle
)
```

- **Service ID[hex]:** 0x06  
- **Sync/Async:** Asynchronous  
- **Reentrancy:** Reentrant (but not for the same NM-Channel)  
- **Parameters (in):** `nmChannelHandle`  
- **Parameters (out):** None  
- **Return value:**  
  - `E_OK`: No error  
  - `E_NOT_OK`: Requesting of network has failed  
- **Description:** Release the network.

**Available via:** `J1939Nm.h`

See section 7.3.2 for details.

See section 7.2.1 for error handling and section 7.9 for parameter checks.
## 8.3.6 J1939Nm_GetState

<table>
<thead>
<tr>
<th>Service name:</th>
<th>J1939Nm_GetState</th>
</tr>
</thead>
</table>
| Syntax:     | Std_ReturnType J1939Nm_GetState(  
|             |   NetworkHandleType NetworkHandle,  
|             |   Nm_StateType* nmStatePtr,  
|             |   Nm_ModeType* nmModePtr  
|             ) |
| Service ID[hex]: | 0x0d |
| Sync/Async: | Synchronous |
| Reentrancy: | Reentrant |
| Parameters (in): | NetworkHandle Identification of the NM-channel |
| Parameters (inout): | None |
| Parameters (out): | nmStatePtr Pointer where state of the network management shall be copied to.  
|                   | nmModePtr Pointer where the mode of the network management shall be copied to. |
| Return value: | Std_ReturnType E_OK: No error  
|               | E_NOT_OK: Getting of NM state has failed |
| Description: | Returns the state and the mode of the network management. |
| Available via: | J1939Nm.h |

See section 7.2.1 for error handling and section 7.9 for parameter checks.

## 8.3.7 J1939Nm_GetBusOffDelay

<table>
<thead>
<tr>
<th>Service name:</th>
<th>J1939Nm_GetBusOffDelay</th>
</tr>
</thead>
</table>
| Syntax:     | void J1939Nm_GetBusOffDelay(  
|             |   NetworkHandleType network,  
|             |   uint8* delayCyclesPtr  
|             ) |
| Service ID[hex]: | 0x10 |
| Sync/Async: | Synchronous |
| Reentrancy: | Reentrant for different networks |
| Parameters (in): | network CAN network where a BusOff occurred. |
| Parameters (inout): | None |

See section 7.2.1 for error handling and section 7.9 for parameter checks.
### Specification of Network Management for SAE J1939

<table>
<thead>
<tr>
<th>Parameters (out):</th>
<th>delayCyclesPtr</th>
<th>Number of CanSM base cycles to wait additionally to L1/L2 after a BusOff occurred.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return value:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td>This callout function returns the number of CanSM base cycles to wait additionally to L1/L2 after a BusOff occurred.</td>
<td></td>
</tr>
<tr>
<td>Available via:</td>
<td>J1939Nm.h</td>
<td></td>
</tr>
</tbody>
</table>

See section 7.2.1 for error handling and section 7.9 for parameter checks.

#### 8.3.8 J1939Nm_PassiveStartUp

<table>
<thead>
<tr>
<th>Service name:</th>
<th>J1939Nm_PassiveStartUp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td>Std_ReturnType J1939Nm_PassiveStartUp( NetworkHandleType nmChannelHandle )</td>
</tr>
<tr>
<td>Service ID[hex]:</td>
<td>0x0f</td>
</tr>
<tr>
<td>Sync/Async:</td>
<td>Synchronous</td>
</tr>
<tr>
<td>Reentrancy:</td>
<td>Reentrant (but not for the same NM-Channel)</td>
</tr>
<tr>
<td>Parameters (in):</td>
<td>nmChannelHandle</td>
</tr>
<tr>
<td>Parameters (inout):</td>
<td>None</td>
</tr>
<tr>
<td>Parameters (out):</td>
<td>None</td>
</tr>
<tr>
<td>Return value:</td>
<td>Std_ReturnType E_OK: No error E_NOT_OK: Passive startup of network management has failed</td>
</tr>
<tr>
<td>Description:</td>
<td>Passive startup of the NM. It triggers the transition from Bus-Sleep Mode to the Network Mode without requesting the network.</td>
</tr>
<tr>
<td>Available via:</td>
<td>J1939Nm.h</td>
</tr>
</tbody>
</table>

This API is just a dummy to satisfy NM interface linkage. It shall always return E_NOT_OK.

See section 7.2.1 for error handling and section 7.9 for parameter checks.

#### 8.4 Call-back notifications

This is a list of functions provided for other modules.

#### 8.4.1 J1939Nm_RxIndication

<table>
<thead>
<tr>
<th>Service name:</th>
<th>J1939Nm_RxIndication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td>void J1939Nm_RxIndication( PduIdType RxPduId, const PduInfoType* PduInfoPtr )</td>
</tr>
<tr>
<td>Service ID[hex]:</td>
<td>0x42</td>
</tr>
<tr>
<td>Sync/Async:</td>
<td>Synchronous</td>
</tr>
</tbody>
</table>
### 8.4.2 J1939Nm_TxConfirmation

#### [SWS_J1939Nm_00037] [ ]

<table>
<thead>
<tr>
<th>Service name:</th>
<th>J1939Nm_TxConfirmation</th>
</tr>
</thead>
</table>
| Syntax:      | void J1939Nm_TxConfirmation(  
|              | PduIdType TxPduId,  
|              | Std_ReturnType result ) |
| Service ID[hex]: | 0x40 |
| Sync/Async: | Synchronous |
| Reentrancy: | Reentrant for different Pduds. Non reentrant for the same Pdud. |
| Parameters (in): |
| TxPduId | ID of the PDU that has been transmitted. |
| result | E_OK: The PDU was transmitted.  
|         | 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: | J1939Nm.h |

See section 7.4 for details.

See section 7.2.1 for error handling and section 7.9 for parameter checks.

### 8.4.3 J1939Nm_RequestIndication

#### [SWS_J1939Nm_00043] [ ]

<table>
<thead>
<tr>
<th>Service name:</th>
<th>J1939Nm_RequestIndication</th>
</tr>
</thead>
</table>
| Syntax:      | void J1939Nm_RequestIndication(  
|              | uint8 node,  
|              | NetworkHandleType channel,  
|              | uint32 requestedPgn, ) |

See section 7.4 for details.

See section 7.2.1 for error handling and section 7.9 for parameter checks.
const J1939Nm_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: J1939Nm.h

See section 7.6 for details.

[SWS_J1939Nm_00067] [The J1939 Network Management module shall ignore the request indication when the 'sourceAddress' or the 'priority' are not in the valid range, or when 'node' is not one of the configured node IDs (see J1939NmNodeId), or when 'requestedPgn' is not the PGN of AC, or when 'destAddress' is not 0xFF or the address of the reported node. If DET reporting is enabled via J1939NmDevErrorDetect, the J1939 Network Management module shall report the corresponding development error: J1939NM_E_INVALID_NODE for 'node', J1939NM_E_INVALID_PGN for 'requestedPgn', J1939NM_E_INVALID_ADDRESS for 'sourceAddress' or 'destAddress', and J1939NM_E_INVALID_PRIO for 'priority'.]

See section 7.2.1 for further error handling and section 7.9 for further parameter checks.

8.5 Scheduled functions

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

8.5.1 J1939Nm_MainFunction

[SWS_J1939Nm_00038] [Service name: J1939Nm_MainFunction
Syntax: void J1939Nm_MainFunction(
    void )
Service ID[hex]: 0x04
]
8.6 Expected Interfaces

In this section, all interfaces required from other modules are listed.

8.6.1 Mandatory Interfaces

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

<table>
<thead>
<tr>
<th>API function</th>
<th>Header File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BswM_J1939Nm_StateChangeNotification</td>
<td>BswM_J1939Nm.h</td>
<td>Notification of current J1939Nm state after state changes.</td>
</tr>
<tr>
<td>CanIf_Transmit</td>
<td>CanIf.h</td>
<td>Requests transmission of a PDU.</td>
</tr>
<tr>
<td>Dem_SetEventStatus</td>
<td>Dem.h</td>
<td>Called by SW-Cs or BSW modules to report monitor status information to Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value.</td>
</tr>
<tr>
<td>Nm_BusSleepMode</td>
<td>Nm.h</td>
<td>Notification that the network management has entered Bus-Sleep Mode.</td>
</tr>
<tr>
<td>Nm_NetworkMode</td>
<td>Nm.h</td>
<td>Notification that the network management has entered Network Mode.</td>
</tr>
<tr>
<td>Nm_StateChangeNotification</td>
<td>Nm.h</td>
<td>Notification that the state of the lower layer &lt;BusNm&gt; has changed.</td>
</tr>
</tbody>
</table>

8.6.2 Optional Interfaces

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

<table>
<thead>
<tr>
<th>API function</th>
<th>Header File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Det_ReportError</td>
<td>Det.h</td>
<td>Service to report development errors.</td>
</tr>
</tbody>
</table>
8.6.3 Configurable interfaces

In this section, all interfaces are listed where the target function could be configured. The target function is usually a call-back function. The name of this kind of interfaces is not fixed because they are configurable.

8.6.3.1 <User_AddressClaimedIndication>

<table>
<thead>
<tr>
<th>Service name:</th>
<th>&lt; User_AddressClaimedIndication &gt;</th>
</tr>
</thead>
</table>
| Syntax:      | void < User_AddressClaimedIndication >(
|              |     NetworkHandleType channel,
|              |      uint8 sourceAddress,
|              |     const uint8* name ) |
| Service ID[hex]: | 0x20 |
| Sync/Async:  | Synchronous |
| Reentrancy:  | Reentrant |
| Parameters (in): channel Channel on which the AC was received. sourceAddress Address of the node that sent the AC or NULL address (0xFE). name Pointer to the byte array containing the 64bit NAME. |
| Parameters (inout): None |
| Parameters (out): None |
| Return value: None |
| Description: Provides the content of received AddressClaimed (AC) PGs. |
| Available via: J1939Nm_Externals.h |

[SWS_J1939Nm_00028] [SWS_J1939Nm_00059] [The <User_AddressClaimedIndication> function shall only be available if J1939NmUserCallout is configured.]

See section 7.5 for details.
9 Sequence diagrams

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

9.1 Transmission of AddressClaimed

The following diagram shows the interaction with CanIf when an AddressClaimed is transmitted.

![Diagram showing transmission of AddressClaimed]

Figure 2: Transmission of AddressClaimed PG

9.2 Reception of AddressClaimed

The following diagram shows the interaction with CanIf when an AddressClaimed is received.
9.3 Request for AddressClaimed

The following diagram shows the interaction with J1939Rm and CanIf when a request for AddressClaimed is handled.
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 the chapter 10.1 “Introduction to configuration specification” in the SWS BSW General [4].

Section 10.1 specifies the structure (containers) and the parameters of the J1939 Network Management module.

Section 10.2 gives hints on how to configure the NM interface to support J1939Nm.

Section 10.3 specifies published information of the J1939 Network Management module.

10.1 Containers and configuration parameters

The following sections summarize all configuration parameters of the J1939 Network Management module. The detailed meaning of the parameters is described in chapters 7 and 8.

The following pictures show an overview of the configuration parameters available for J1939Nm:
Figure 5: Configuration container J1939Nm
Figure 6: Configuration container J1939NmChannel
Figure 7: Configuration container J1939NmNode
Figure 8: Configuration container J1939NmExternalNode

10.1.1 J1939Nm

**SWS Item**  
ECUC_J1939Nm_00028 : 

**Module Name**  
J1939Nm

**Module Description**  
Configuration of the J1939 Network Management module.
Post-Build Variant Support: true

Supported Config Variants: VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

Included Containers

<table>
<thead>
<tr>
<th>Container Name</th>
<th>Multiplicity</th>
<th>Scope / Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1939NmConfigSet</td>
<td>1</td>
<td>This container contains the configuration parameters and subcontainers of the AUTOSAR J1939Nm module.</td>
</tr>
<tr>
<td>J1939NmGeneral</td>
<td>1</td>
<td>Contains the general configuration parameters of the module.</td>
</tr>
</tbody>
</table>

10.1.2 J1939NmGeneral

SWS Item: ECUC_J1939Nm_00001:

<table>
<thead>
<tr>
<th>Container Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1939NmGeneral</td>
<td>Contains the general configuration parameters of the module.</td>
</tr>
</tbody>
</table>

Configuration Parameters

SWS Item: ECUC_J1939Nm_00034:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1939NmBusOffDelayTickPeriod</td>
<td>Duration of ticks that are used to time BusOff delays after conflicting address claims. This parameter must be synchronized with the main function period of the CAN State Manager.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiplicity</th>
<th>Type</th>
<th>Range</th>
<th>Default value</th>
<th>Post-Build Variant Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>EcucFloatParamDef</td>
<td>[0 .. INF]</td>
<td>0.02</td>
<td>false</td>
</tr>
</tbody>
</table>

Configuration Class:

- Pre-compile time: X VARIANT-PRE-COMPILE
- Link time: X VARIANT-LINK-TIME, VARIANT-POST-BUILD
- Post-build time: --

Value Configuration Class:

- Pre-compile time: X VARIANT-PRE-COMPILE
- Link time: X VARIANT-LINK-TIME, VARIANT-POST-BUILD
- Post-build time: --

Scope / Dependency: scope: local

SWS Item: ECUC_J1939Nm_00003:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1939NmDevErrorDetect</td>
<td>Switches the development error detection and notification on or off.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiplicity</th>
<th>Type</th>
<th>Default value</th>
<th>Post-Build Variant Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EcucBooleanParamDef</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

Configuration Class:

- Pre-compile time: X VARIANT-PRE-COMPILE
- Link time: X VARIANT-LINK-TIME, VARIANT-POST-BUILD
- Post-build time: --

Value Configuration Class:

- Pre-compile time: X All Variants
### SWS Item: ECUC_J1939Nm_00036
- **Name:** J1939NmGatewaySupport
- **Parent Container:** J1939NmGeneral
- **Description:** Enables/disables support for claiming the addresses of routed messages.
- **Multiplicity:** 1
- **Type:** EcucBooleanParamDef
- **Default value:** --
- **Post-Build Variant Value:** false

#### Value Configuration Class
- **Pre-compile time:** X All Variants
- **Link time:** --
- **Post-build time:** --

### SWS Item: ECUC_J1939Nm_00004
- **Name:** J1939NmMainFunctionPeriod
- **Parent Container:** J1939NmGeneral
- **Description:** Call cycle in seconds of J1939Nm_MainFunction.
- **Multiplicity:** 1
- **Type:** EcucFloatParamDef
- **Range:** [0 .. INF]
- **Default value:** 0.01
- **Post-Build Variant Value:** false

#### Value Configuration Class
- **Pre-compile time:** X VARIANT-PRE-COMPILE
- **Link time:** X VARIANT-LINK-TIME, VARIANT-POST-BUILD
- **Post-build time:** --

### SWS Item: ECUC_J1939Nm_00032
- **Name:** J1939NmUserCallout
- **Parent Container:** J1939NmGeneral
- **Description:** Pre-processor switch for enabling the <User_AddressClaimedIndication> and defining the name of the callout function.
- **Multiplicity:** 0..1
- **Type:** EcucFunctionNameDef
- **Default value:** --
- **maxLength:** --
- **minLength:** --
- **regularExpression:** --
- **Post-Build Variant Value:** false

### SWS Item: ECUC_J1939Nm_00002
- **Name:** J1939NmVersionInfoApi
Parent Container: J1939NmGeneral

Description: Pre-processor switch for enabling version info API support.

Multiplicity: 1

Type: EcucBooleanParamDef

Default value: false

Post-Build Variant Value: false

Value Configuration Class:
- Pre-compile time: X All Variants
- Link time: --
- Post-build time: --

Scope / Dependency: scope: local

No Included Containers

10.1.3 J1939NmConfigSet

SWS Item: ECUC_J1939Nm_00027:

Container Name: J1939NmConfigSet

Description: This container contains the configuration parameters and sub containers of the AUTOSAR J1939Nm module.

Configuration Parameters:

Included Containers:

<table>
<thead>
<tr>
<th>Container Name</th>
<th>Multiplicity</th>
<th>Scope / Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1939NmChannel</td>
<td>1..*</td>
<td>Physical CAN channel handled by J1939Nm.</td>
</tr>
<tr>
<td>J1939NmExternalNode</td>
<td>0..*</td>
<td>Logical node implemented in another ECU. Configures potential communication partners. If this container is connected to more than one channel, the external ECU is linked to the local ECU by each of these channels.</td>
</tr>
<tr>
<td>J1939NmNode</td>
<td>1..*</td>
<td>Logical node representing one function handled by J1939Nm.</td>
</tr>
<tr>
<td>J1939NmSharedAddressSpace</td>
<td>0..*</td>
<td>Set of J1939NmChannels that share a common address space. Address claims will be routed between these channels.</td>
</tr>
</tbody>
</table>

10.1.4 J1939NmSharedAddressSpace

SWS Item: ECUC_J1939Nm_00037:

Container Name: J1939NmSharedAddressSpace

Description: Set of J1939NmChannels that share a common address space. Address claims will be routed between these channels.

Post-Build Variant Multiplicity: true

Multiplicity Configuration Class:
- Pre-compile time: X VARIANT-PRE-COMPILE
- Link time: X VARIANT-LINK-TIME
- Post-build time: X VARIANT-POST-BUILD

Configuration Parameters:

<table>
<thead>
<tr>
<th>SWS Item</th>
<th>Name</th>
<th>Parent Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECUC_J1939Nm_00037</td>
<td>J1939NmSharedChannelRef</td>
<td>J1939NmSharedAddressSpace</td>
</tr>
</tbody>
</table>
### Description
Reference to a channel that belongs to the shared address space.

### Multiplicity
2..*

### Type
Reference to [J1939NmChannel]

### Post-Build Variant

<table>
<thead>
<tr>
<th>Multiplicity Configuration Class</th>
<th>Pre-compile time</th>
<th>Link time</th>
<th>Post-build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### Value Configuration Class

<table>
<thead>
<tr>
<th>Pre-compile time</th>
<th>Link time</th>
<th>Post-build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### Scope / Dependency
scope: local

### No Included Containers

### 10.1.5 J1939NmChannel

#### SWS Item
ECUC_J1939Nm_00005:

- **Container Name**: J1939NmChannel
- **Description**: Physical CAN channel handled by J1939Nm.
- **Post-Build Variant**

<table>
<thead>
<tr>
<th>Multiplicity Configuration Class</th>
<th>Pre-compile time</th>
<th>Link time</th>
<th>Post-build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### Configuration Parameters

#### SWS Item
ECUC_J1939Nm_00035:

- **Name**: J1939NmChannelUsesAddressArbitration
- **Parent Container**: J1939NmChannel
- **Description**: Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes.
  - True: The initial address claim is sent, and the node reacts to address claims of other nodes.
  - False: The node only sends an address claim upon request, and does not react to other address claims.
- **Multiplicity**: 1
- **Type**: EcucBooleanParamDef
- **Default value**: true
- **Post-Build Variant Value**

<table>
<thead>
<tr>
<th>Value Configuration Class</th>
<th>Pre-compile time</th>
<th>Link time</th>
<th>Post-build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### Scope / Dependency
scope: local

#### SWS Item
ECUC_J1939Nm_00008:

- **Name**: J1939NmComMNetworkHandleRef
- **Parent Container**: J1939NmChannel
- **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] |

#### Post-Build Variant Value

| false |

#### Value Configuration Class

| Pre-compile time | X VARIANT-PRE-COMPILE |
| Link time | X VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| Post-build time | -- |

#### Scope / Dependency

| scope: local |

### Included Containers

<table>
<thead>
<tr>
<th>Container Name</th>
<th>Multiplicity</th>
<th>Scope / Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1939NmNodeSpecificDemEventParameterRefs</td>
<td>0..*</td>
<td>Container for the references to DemEventParameter elements related to one J1939NmNode which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.</td>
</tr>
<tr>
<td>J1939NmRxPdu</td>
<td>1</td>
<td>Contains the configuration of the PDU used to receive the AddressClaimed PG. This PDU consumes a meta data item of type CAN_ID_32.</td>
</tr>
<tr>
<td>J1939NmTxPdu</td>
<td>1</td>
<td>Contains the configuration of the PDU used to transmit the AddressClaimed PG. This PDU produces a meta data item of type CAN_ID_32.</td>
</tr>
</tbody>
</table>

### 10.1.6 J1939NmTxPdu

#### SWS Item

| ECUC_J1939Nm_00009 : |

#### Container Name

| J1939NmTxPdu |

#### Description

Contains the configuration of the PDU used to transmit the AddressClaimed PG. This PDU produces a meta data item of type CAN_ID_32.

#### SWS Item

| ECUC_J1939Nm_00011 : |

#### Name

| J1939NmTxPduId |

#### Parent Container

| J1939NmTxPdu |

#### Description

The PDU identifier used for TxConfirmation from CanIf.

#### Multiplicity

| 1 |

#### Type

| EcucIntegerParamDef (Symbolic Name generated for this parameter) |

#### Range

| 0 .. 65535 |

#### Default value

| -- |

#### Post-Build Variant Value

| false |

#### Value Configuration Class

| Pre-compile time | X All Variants |
| Link time | -- |
| Post-build time | -- |

#### Scope / Dependency

| scope: ECU |

#### SWS Item

| ECUC_J1939Nm_00012 : |
## 10.1.7 J1939NmRxPdu

**SWS Item** ECUC_J1939Nm_00010 :

<table>
<thead>
<tr>
<th>Name</th>
<th>J1939NmRxPdu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Container</td>
<td>J1939NmRxPdu</td>
</tr>
<tr>
<td>Description</td>
<td>Contains the configuration of the PDU used to receive the AddressClaimed PG. This PDU consumes a meta data item of type CAN_ID_32.</td>
</tr>
</tbody>
</table>

### Configuration Parameters

<table>
<thead>
<tr>
<th>SWS Item</th>
<th>ECUC_J1939Nm_00014 :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>J1939NmRxPduId</td>
</tr>
<tr>
<td>Parent Container</td>
<td>J1939NmRxPdu</td>
</tr>
<tr>
<td>Description</td>
<td>The PDU identifier used for RxIndication from CanIf.</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>1</td>
</tr>
<tr>
<td>Type</td>
<td>EcuIntegerParamDef (Symbolic Name generated for this parameter)</td>
</tr>
<tr>
<td>Range</td>
<td>0 .. 65535</td>
</tr>
<tr>
<td>Default value</td>
<td>--</td>
</tr>
<tr>
<td>Post-Build Variant Value</td>
<td>false</td>
</tr>
<tr>
<td>Value Configuration Class</td>
<td>Pre-compile time</td>
</tr>
<tr>
<td></td>
<td>Link time</td>
</tr>
<tr>
<td></td>
<td>Post-build time</td>
</tr>
<tr>
<td>Scope / Dependency</td>
<td>scope: ECU</td>
</tr>
</tbody>
</table>

**SWS Item** ECUC_J1939Nm_00013 :

<table>
<thead>
<tr>
<th>Name</th>
<th>J1939NmRxPduRef</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Container</td>
<td>J1939NmRxPdu</td>
</tr>
<tr>
<td>Description</td>
<td>Reference to the Pdu object representing the PDU.</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>1</td>
</tr>
<tr>
<td>Type</td>
<td>Reference to [ Pdu ]</td>
</tr>
<tr>
<td>Post-Build Variant Value</td>
<td>false</td>
</tr>
<tr>
<td>Value Configuration Class</td>
<td>Pre-compile time</td>
</tr>
<tr>
<td></td>
<td>Link time</td>
</tr>
<tr>
<td></td>
<td>Post-build time</td>
</tr>
<tr>
<td>Scope / Dependency</td>
<td>scope: local</td>
</tr>
</tbody>
</table>

No Included Containers
### 10.1.8 J1939NmNodeSpecificDemEventParameterRefs

<table>
<thead>
<tr>
<th>SWS Item</th>
<th>ECUC_J1939Nm_00006 :</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Container Name</strong></td>
<td>J1939NmNodeSpecificDemEventParameterRefs</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Container for the references to DemEventParameter elements related to one J1939NmNode which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.</td>
</tr>
<tr>
<td><strong>Post-Build Variant Multiplicity</strong></td>
<td>true</td>
</tr>
<tr>
<td><strong>Multiplicity Configuration</strong></td>
<td>Pre-compile time: X VARIANT-PRE-COMPILE, Link time: X VARIANT-LINK-TIME, Post-build time: X VARIANT-POST-BUILD</td>
</tr>
</tbody>
</table>

### SWS Item ECUC_J1939Nm_00007 :

| **Name** | J1939NM_E_ADDRESS_LOST |
| **Parent Container** | J1939NmNodeSpecificDemEventParameterRefs |
| **Description** | Reference to the DemEventParameter which shall be issued when the ECU failed to claim one of its addresses. |
| **Multiplicity** | 1 |
| **Type** | Symbolic name reference to [ DemEventParameter ] |
| **Post-Build Variant Value** | false |
| **Value Configuration Class** | Pre-compile time: X All Variants, Link time: --, Post-build time: -- |
| **Scope / Dependency** | scope: local |

### SWS Item ECUC_J1939Nm_00053 :

| **Name** | J1939NmNodeRef |
| **Parent Container** | J1939NmNodeSpecificDemEventParameterRefs |
| **Description** | Reference to J1939NmNode. |
| **Multiplicity** | 1 |
| **Type** | Reference to [ J1939NmNode ] |
| **Post-Build Variant Value** | true |
| **Value Configuration Class** | Pre-compile time: X VARIANT-PRE-COMPILE, Link time: X VARIANT-LINK-TIME, Post-build time: X VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local |

### No Included Containers

### 10.1.9 J1939NmNode

| SWS Item | ECUC_J1939Nm_00015 :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Container Name</strong></td>
<td>J1939NmNode</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Logical node representing one function handled by J1939Nm.</td>
</tr>
<tr>
<td><strong>Post-Build Variant Multiplicity</strong></td>
<td>true</td>
</tr>
<tr>
<td><strong>Multiplicity Configuration Class</strong></td>
<td>Pre-compile time: X VARIANT-PRE-COMPILE, Link time: X VARIANT-LINK-TIME</td>
</tr>
</tbody>
</table>
### SWS Item ECUC_J1939Nm_00030:

**Name**: J1939NmNodeId

**Parent Container**: J1939NmNode

**Description**: Unique identifier of this node.

**Multiplicity**: 1

**Type**: EcuiIntegerParamDef (Symbolic Name generated for this parameter)

**Range**: 0 .. 255

**Default value**: --

**Post-Build Variant Value**: false

**Value Configuration Class**

| Post-build time | VARIANT-POST-BUILD |

**Scope / Dependency**: scope: ECU

### SWS Item ECUC_J1939Nm_00018:

**Name**: J1939NmNodeNameArbitraryAddressCapable

**Parent Container**: J1939NmNode

**Description**: Arbitrary Address Capable field of the NAME of this node.

**Multiplicity**: 1

**Type**: EcucBooleanParamDef

**Default value**: --

**Post-Build Variant Value**: false

**Value Configuration Class**

<table>
<thead>
<tr>
<th>Pre-compile time</th>
<th>X</th>
<th>All Variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link time</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Post-build time</td>
<td>--</td>
<td></td>
</tr>
</tbody>
</table>

**Scope / Dependency**: scope: local

### SWS Item ECUC_J1939Nm_00024:

**Name**: J1939NmNodeNameECUInstance

**Parent Container**: J1939NmNode

**Description**: ECU Instance field of the NAME of this node.

**Multiplicity**: 1

**Type**: EcuiIntegerParamDef

**Range**: 0 .. 7

**Default value**: --

**Post-Build Variant Value**: true

**Value Configuration Class**

<table>
<thead>
<tr>
<th>Pre-compile time</th>
<th>VARIANT-PRE-COMPILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link time</td>
<td>X</td>
</tr>
<tr>
<td>Post-build time</td>
<td>X</td>
</tr>
</tbody>
</table>

**Scope / Dependency**: scope: local

### SWS Item ECUC_J1939Nm_00022:

**Name**: J1939NmNodeNameFunction

**Parent Container**: J1939NmNode

**Description**: Function field of the NAME of this node.

**Multiplicity**: 1

**Type**: EcuiIntegerParamDef

**Range**: 0 .. 255

**Default value**: --

**Post-Build Variant Value**: true

**Value Configuration Class**

<table>
<thead>
<tr>
<th>Pre-compile time</th>
<th>VARIANT-PRE-COMPILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link time</td>
<td>X</td>
</tr>
</tbody>
</table>

**Scope / Dependency**: scope: local
### Scope / Dependency
- **Post-build time**: X VARIANT-POST-BUILD
- **scope**: local

### SWS Item
- **ECUC_J1939Nm_00023**
- **Name**: J1939NmNodeNameFunctionInstance
- **Parent Container**: J1939NmNode
- **Description**: Function Instance field of the NAME of this node.
- **Multiplicity**: 1
- **Type**: EcuIntegerParamDef
- **Range**: 0 .. 31
- **Default value**: --
- **Post-Build Variant Value**: true
- **Value Configuration Class**:
  - Pre-compile time: X VARIANT-PRE-COMPILE
  - Link time: X VARIANT-LINK-TIME
  - Post-build time: X VARIANT-POST-BUILD

### Scope / Dependency
- **scope**: local

### SWS Item
- **ECUC_J1939Nm_00026**
- **Name**: J1939NmNodeNameIdentityNumber
- **Parent Container**: J1939NmNode
- **Description**: Identity Number field of the NAME of this node.
- **Multiplicity**: 1
- **Type**: EcuIntegerParamDef
- **Range**: 0 .. 2097151
- **Default value**: --
- **Post-Build Variant Value**: true
- **Value Configuration Class**:
  - Pre-compile time: X VARIANT-PRE-COMPILE
  - Link time: X VARIANT-LINK-TIME
  - Post-build time: X VARIANT-POST-BUILD

### Scope / Dependency
- **scope**: local

### SWS Item
- **ECUC_J1939Nm_00019**
- **Name**: J1939NmNodeNameIndustryGroup
- **Parent Container**: J1939NmNode
- **Description**: Industry Group field of the NAME of this node.
- **Multiplicity**: 1
- **Type**: EcuIntegerParamDef
- **Range**: 0 .. 7
- **Default value**: --
- **Post-Build Variant Value**: true
- **Value Configuration Class**:
  - Pre-compile time: X VARIANT-PRE-COMPILE
  - Link time: X VARIANT-LINK-TIME
  - Post-build time: X VARIANT-POST-BUILD

### Scope / Dependency
- **scope**: local

### SWS Item
- **ECUC_J1939Nm_00025**
- **Name**: J1939NmNodeNameManufacturerCode
- **Parent Container**: J1939NmNode
- **Description**: Manufacturer Code field of the NAME of this node.
- **Multiplicity**: 1
- **Type**: EcuIntegerParamDef
- **Range**: 0 .. 2047
- **Default value**: --
- **Post-Build Variant Value**: true
- **Value Configuration Class**:
  - Pre-compile time: X VARIANT-PRE-COMPILE
**SWS Item** ECUC_J1939Nm_00021 :
**Name** J1939NmNodeNameVehicleSystem
**Parent Container** J1939NmNode
**Description** Vehicle System field of the NAME of this node.
**Multiplicity** 1
**Type** EcucIntegerParamDef
**Range** 0 .. 127
**Default value** --
**Value Configuration Class**
**Pre-compile time** X VARIANT-PRE-COMPILE
**Link time** X VARIANT-LINK-TIME
**Post-build time** X VARIANT-POST-BUILD

---

**SWS Item** ECUC_J1939Nm_00020 :
**Name** J1939NmNodeNameVehicleSystemInstance
**Parent Container** J1939NmNode
**Description** Vehicle System Instance field of the NAME of this node.
**Multiplicity** 1
**Type** EcucIntegerParamDef
**Range** 0 .. 15
**Default value** --
**Value Configuration Class**
**Pre-compile time** X VARIANT-PRE-COMPILE
**Link time** X VARIANT-LINK-TIME
**Post-build time** X VARIANT-POST-BUILD

---

**SWS Item** ECUC_J1939Nm_00016 :
**Name** J1939NmNodePreferredAddress
**Parent Container** J1939NmNode
**Description** Source address of this node used for address claiming.
**Multiplicity** 1
**Type** EcucIntegerParamDef
**Range** 0 .. 253
**Default value** --
**Value Configuration Class**
**Pre-compile time** X VARIANT-PRE-COMPILE
**Link time** X VARIANT-LINK-TIME
**Post-build time** X VARIANT-POST-BUILD

---

**SWS Item** ECUC_J1939Nm_00017 :
**Name** J1939NmNodeStartUpDelay
**Parent Container** J1939NmNode
**Description** If enabled, the node will start communication after a delay of 250ms after transmission of the initial AddressClaimed, depending on the configured J1939NmNodePreferredAddress. If disabled, the node will start communication immediately at network start-up.

Please note: According to J1939/81, the 250ms delay is not required for single address CAs with desired source addresses in the ranges 0..127 or 248..253.
Specification of Network Management for SAE J1939
AUTOSAR CP Release 4.4.0

| Multiplicity | 1          |
| Type         | EcucBooleanParamDef |
| Default value | true          |
| Post-Build Variant Value | false |

<table>
<thead>
<tr>
<th>Value Configuration Class</th>
<th>Pre-compile time</th>
<th>Link time</th>
<th>Post-build time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>VAR-O-COMP</td>
<td></td>
</tr>
</tbody>
</table>

| Scope / Dependency | scope: local |

SWS Item | ECUC_J1939Nm_00029 : |
Name | J1939NmNodeChannelRef |
Parent Container | J1939NmNode |
Description | Reference to the channels this node has access to. |
Multiplicity | 1..* |
Type | Reference to [ J1939NmChannel ] |
Post-Build Variant Multiplicity | false |
Post-Build Variant Value | false |

<table>
<thead>
<tr>
<th>Multiplicity Configuration Class</th>
<th>Pre-compile time</th>
<th>Link time</th>
<th>Post-build time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>VAR-O-COMP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value Configuration Class</th>
<th>Pre-compile time</th>
<th>Link time</th>
<th>Post-build time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>VAR-O-COMP</td>
<td></td>
</tr>
</tbody>
</table>

| Scope / Dependency | scope: local |

No Included Containers

10.1.10 J1939NmExternalNode

SWS Item | ECUC_J1939Nm_00039 : |
Container Name | J1939NmExternalNode |
Description | Logical node implemented in another ECU. Configures potential communication partners. If this container is connected to more than one channel, the external ECU is linked to the local ECU by each of these channels. |
Post-Build Variant Multiplicity | false |

<table>
<thead>
<tr>
<th>Multiplicity Configuration Class</th>
<th>Pre-compile time</th>
<th>Link time</th>
<th>Post-build time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>VAR-O-COMP</td>
<td></td>
</tr>
</tbody>
</table>

| Configuration Parameters |

SWS Item | ECUC_J1939Nm_00040 : |
Name | J1939NmExternalNodeId |
Parent Container | J1939NmExternalNode |
Description | Unique identifier of this external node. |
Multiplicity | 1 |
Type | EcucIntegerParamDef (Symbolic Name generated for this parameter) |
Range | 0 .. 65535 |
Default value | -- |
<table>
<thead>
<tr>
<th>SWS Item</th>
<th>Name</th>
<th>Description</th>
<th>Multiplicity</th>
<th>Type</th>
<th>Default value</th>
<th>Post-Build Variant Value</th>
<th>Value Configuration Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECUC_J1939Nm_00041:</td>
<td>J1939NmExternalNodeNameArbitraryAddressCapable</td>
<td>Arbitrary Address Capable field of the NAME of this external node.</td>
<td>1</td>
<td>EcucBooleanParamDef</td>
<td>--</td>
<td>true</td>
<td>Pre-compile time: X VARIANT-PRE-COMPILE, Link time: X VARIANT-LINK-TIME, Post-build time: X VARIANT-POST-BUILD</td>
</tr>
<tr>
<td>ECUC_J1939Nm_00042:</td>
<td>J1939NmExternalNodeNameECUInstance</td>
<td>ECU Instance field of the NAME of this external node.</td>
<td>1</td>
<td>EcuIntegerParamDef</td>
<td>--</td>
<td>true</td>
<td>Pre-compile time: X VARIANT-PRE-COMPILE, Link time: X VARIANT-LINK-TIME, Post-build time: X VARIANT-POST-BUILD</td>
</tr>
<tr>
<td>ECUC_J1939Nm_00043:</td>
<td>J1939NmExternalNodeNameFunction</td>
<td>Function field of the NAME of this external node.</td>
<td>1</td>
<td>EcuIntegerParamDef</td>
<td>0 .. 255</td>
<td>true</td>
<td>Pre-compile time: X VARIANT-PRE-COMPILE, Link time: X VARIANT-LINK-TIME, Post-build time: X VARIANT-POST-BUILD</td>
</tr>
<tr>
<td>ECUC_J1939Nm_00044:</td>
<td>J1939NmExternalNodeNameFunctionInstance</td>
<td>Function Instance field of the NAME of this external node.</td>
<td>1</td>
<td>EcuIntegerParamDef</td>
<td>0 .. 31</td>
<td>false</td>
<td>Pre-compile time: X VARIANT-PRE-COMPILE, Link time: X VARIANT-LINK-TIME, Post-build time: X VARIANT-POST-BUILD</td>
</tr>
</tbody>
</table>
SWS Item  ECUC_J1939Nm_00045:
Name  J1939NmExternalNodeNameIdentityNumber
Parent Container  J1939NmExternalNode
Description  Identity Number field of the NAME of this external node.
Multiplicity 1
Type  EcucIntegerParamDef
Range 0 .. 2097151
Default value --
Post-Build Variant Value true
Value Configuration Class
Pre-compile time X VARIANT-PRE-COMPILE
Link time X VARIANT-LINK-TIME
Post-build time X VARIANT-POST-BUILD
Scope / Dependency scope: local

SWS Item  ECUC_J1939Nm_00046:
Name  J1939NmExternalNodeNameIndustryGroup
Parent Container  J1939NmExternalNode
Description  Industry Group field of the NAME of this external node.
Multiplicity 1
Type  EcucIntegerParamDef
Range 0 .. 7
Default value --
Post-Build Variant Value true
Value Configuration Class
Pre-compile time X VARIANT-PRE-COMPILE
Link time X VARIANT-LINK-TIME
Post-build time X VARIANT-POST-BUILD
Scope / Dependency scope: local

SWS Item  ECUC_J1939Nm_00047:
Name  J1939NmExternalNodeNameManufacturerCode
Parent Container  J1939NmExternalNode
Description  Manufacturer Code field of the NAME of this external node.
Multiplicity 1
Type  EcucIntegerParamDef
Range 0 .. 2047
Default value --
Post-Build Variant Value true
Value Configuration Class
Pre-compile time X VARIANT-PRE-COMPILE
Link time X VARIANT-LINK-TIME
Post-build time X VARIANT-POST-BUILD
Scope / Dependency scope: local

SWS Item  ECUC_J1939Nm_00048:
Name  J1939NmExternalNodeNameVehicleSystem
Parent Container  J1939NmExternalNode
Description  Vehicle System field of the NAME of this external node.
Multiplicity 1
Type  EcucIntegerParamDef
Range 0 .. 127
### SWS Item ECUC_J1939Nm_00050:
- **Name**: J1939NmExternalNodeNameVehicleSystemInstance
- **Parent Container**: J1939NmExternalNode
- **Description**: Vehicle System Instance field of the NAME of this external node.
- **Multiplicity**: 1
- **Type**: EcucIntegerParamDef
- **Range**: 0 .. 15
- **Default value**: --
- **Post-Build Variant Value**: true

### SWS Item ECUC_J1939Nm_00049:
- **Name**: J1939NmExternalNodePreferredAddress
- **Parent Container**: J1939NmExternalNode
- **Description**: Source address of this external node.
- **Multiplicity**: 1
- **Type**: EcucIntegerParamDef
- **Range**: 0 .. 253
- **Default value**: --
- **Post-Build Variant Value**: true

### SWS Item ECUC_J1939Nm_00051:
- **Name**: J1939NmExternalNodeChannelRef
- **Parent Container**: J1939NmExternalNode
- **Description**: Reference to the channels of the local ECU this external node has access to.
- **Multiplicity**: 1..*
- **Type**: Reference to [ J1939NmChannel ]
- **Post-Build Variant Value**: true

### SWS Item ECUC_J1939Nm_00052:
- **Name**: J1939NmExternalNodeGatewayedChannelRef
- **Parent Container**: J1939NmExternalNode
<table>
<thead>
<tr>
<th>Description</th>
<th>Reference to the channels on which messages to/from this external node shall be gatewayed. The address claim from the external node will be replicated on these channels.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplicity</td>
<td>0..*</td>
</tr>
<tr>
<td>Type</td>
<td>Reference to [ J1939NmChannel ]</td>
</tr>
<tr>
<td>Post-Build Variant</td>
<td>true</td>
</tr>
<tr>
<td>Post-Build Variant Value</td>
<td>true</td>
</tr>
<tr>
<td>Multiplicity Configuration Class</td>
<td>Pre-compile time X VARIANT-PRE-COMPILE</td>
</tr>
<tr>
<td></td>
<td>Link time X VARIANT-LINK-TIME</td>
</tr>
<tr>
<td></td>
<td>Post-build time X VARIANT-POST-BUILD</td>
</tr>
<tr>
<td>Value Configuration Class</td>
<td>Pre-compile time X VARIANT-PRE-COMPILE</td>
</tr>
<tr>
<td></td>
<td>Link time X VARIANT-LINK-TIME</td>
</tr>
<tr>
<td></td>
<td>Post-build time X VARIANT-POST-BUILD</td>
</tr>
<tr>
<td>Scope / Dependency</td>
<td>scope: local</td>
</tr>
</tbody>
</table>

### 10.2 Configuration of NM Interface

The J1939 Network Management module relies on the following channel configuration in the NM Interface to be operational:

- NmActiveCoordinator: False
- NmBusSynchronizationEnabled: False
- NmChannelSleepMaster: True
- NmComControlEnabled: False
- NmCoordClusterIndex: <undefined>
- NmCoordinatorSyncSupport: False
- NmNodeDetectionEnabled: False
- NmNodeIdEnabled: False
- NmPassiveModeEnabled: False
- NmRemoteSleepIndEnabled: False
- NmShutdownDelayTimer: 0.0
- NmStateReportEnabled: False
- NmStateReportSignalRef: <undefined>
- NmSynchronizingNetwork: False
- NmUserDataEnabled: False

### 10.3 Published Information

For details, refer to the chapter 10.3 “Published Information” in the SWS BSW General [4].