Document Title	Specification of Communication Stack Types
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	050
Document Classification	Standard

Document Version	2.6.0
Document Status	Final
Part of Release	3.2
Revision	3

Document Change History

Date	Version	Changed by	Change Description
28.02.2014	2.6.0	AUTOSAR Release Management	• Added Enum values for TPParameterType
29.05.2012	2.5.1	AUTOSAR Administration	• Added PNCHandleType • Removed NTFRSLT_E_CANCELATION_OK and NTFRSLT_E_CANCELATION_NOT_OK from NotifResultType
27.04.2011	2.4.0	AUTOSAR Administration	• Updated Chapter 8 and 10 • Legal disclaimer revised
10.09.2010	2.3.0	AUTOSAR Administration	• Published information of the document is updated • Legal disclaimer revised
23.06.2008	2.2.1	AUTOSAR Administration	• Legal disclaimer revised
27.11.2007	2.2.0	AUTOSAR Administration	• Chapter numbers in chapter 8.1 corrected • New data type NetworkHandleType created, according item COMTYPE026 established • Syntax correction in PduInfoType • Document meta information extended • Small layout adaptations made
24.01.2007	2.1.1	AUTOSAR Administration	• “Advice for users” revised • “Revision Information” added • Changed “sender” to “receiver” at NTFRSLT_E_WFT_OVRN

Document Change History			
14.12.2006	2.1.0	AUTOSAR Administration	• NTFRSLT_E_TIMEOUT_Bs changed NTFRSLT_E_TIMEOUT_BS • NTFRSLT_E_TIMEOUT_Cr changed to NTFRSLT_E_TIMEOUT_CR • Definitions according to compiler abstraction added • Legal disclaimer revised
21.03.2006	2.0.0	AUTOSAR Administration	• Initial release (The V1.0.0 was only as Pre-Release available within Release 1.0)

Disclaimer

This specification 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 specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification 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 specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1 Introduction and functional overview 5

2 Acronyms and abbreviations 6

3 Related documentation... 7
3.1 Input documents... 7
3.2 Related standards and norms 7

4 Constraints and assumptions 8
4.1 Limitations 8
4.2 Applicability to car domains... 8
4.3 Applicability to safety related environments 8

5 Software Architecture 9
5.1 Dependencies to other modules... 9
5.2 File structure 9

6 Requirements traceability 10

7 Functional specification 16
7.1 General issues 16

8 API specification... 17
8.1 Type definitions 17
8.1.1 PduIdType 17
8.1.2 PduLengthType 17
8.1.3 PduInfoType 18
8.1.4 TPParameterType 18
8.1.5 BufReq_ReturnType 18
8.1.6 NotifResultType 18
8.1.7 BusTrcvErrorType... 20
8.1.8 NetworkHandleType 21
8.1.9 PNCHandleType... 22
8.2 Function definitions 22

9 Sequence diagrams 23

10 Configuration specification 24
10.1 Published parameters 24

1 Introduction and functional overview

This document specifies the AUTOSAR communication stack type header file. It contains all types that are used across several modules of the communication stack of the basic software and all types of all basic software modules that are platform and compiler independent.
It is strongly recommended that those communication stack type files are unique within the AUTOSAR community to guarantee unique types and to avoid type changes when changing from supplier A to B.

2 Acronyms and abbreviations

Acronyms and abbreviations that have a local scope are not contained in the AUTOSAR glossary. These must appear in a local glossary.

Acronym:	Description:
API	Application Programming Interface
DCM	Diagnostic Communication Manager
I-PDU	Interaction Layer PDU. In AUTOSAR the Interaction Layer is equivalent to the Communication Services Layer.
L-PDU	Data Link Layer PDU. In AUTOSAR the Data Link Layer is equivalent to the Communication Hardware Abstraction and Microcontroller Abstraction Layer.
N-PDU	Network Layer PDU. In AUTOSAR the Network Layer is equivalent to the Transport Protocol.
OSEK/VDX	n May 1993 OSEK has been founded as a joint project in the German automotive industry aiming at an industry standard for an open-ended architecture for distributed control units in vehicles. OSEK is an abbreviation for the German term "Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug" (English: Open Systems and the Corresponding Interfaces for Automotive Electronics). Initial project partners were BMW, Bosch, DaimlerChrysler, Opel, Siemens, VW and the IIIT of the University of Karlsruhe as co-ordinator. The French car manufacturers PSA and Renault joined OSEK in 1994 introducing their VDX-approach (Vehicle Distributed eXecutive) which is a similar project within the French automotive industry. At the first workshop on October 1995 the OSEK/VDX group presented the results of the harmonised specification between OSEK and VDX. After the 2nd international OSEK/VDX Workshop in October 1997 the 2nd versions of the specifications were published.
PDU	Protocol Data Unit
SDU	Service Data Unit - Payload of PDU
TP	Transport Protocol

Abbreviation:	Description:
Com	Communication
e.g.	[lat.] exempli gratia = [eng.] for example
i.e.	[lat.] it est = [eng.] that is

3 Related documentation

3.1 Input documents

[GeneralSRS] General Requirements on Basic Software Modules
AUTOSAR_SRS_General.pdf

[SRSSPAL] General Requirements on SPAL
AUTOSAR_SRS_SPAL.pdf

[StdTypes] Specification of Standard Types
AUTOSAR_SWS_StandardTypes.pdf

[PltfTypes] Specification of Platform Types
AUTOSAR_SWS_PlatformTypes.pdf

[CompTypes] Specification of Compiler Abstraction
AUTOSAR_SWS_CompilerAbstraction.pdf

[CANTP] Specification of CAN Transport Layer
AUTOSAR_SWS_CAN_TP.pdf

[FlexRayTP] Specification of FlexRay Transport Layer
AUTOSAR_SWS_FlexRay_TP.pdf

[CANTRCV] Specification of CAN Transceiver Driver
AUTOSAR_SWS_CANTransceiverDriver.pdf

[FRTRCV] Specification of FlexRay Transceiver Driver
AUTOSAR_SWS_FlexRayTransceiver.pdf

AUTOSAR Basic Software Module Description Template,
AUTOSAR_BSW_Module_Description.pdf

3.2 Related standards and norms

[CProgLang] ISO/IEC 9899:1990 Programming Language – C
[ISONM] ISO/IEC 15765-2; 2003 Diagnostics on Controller Area Networks (CAN) – Network layer services

4 Constraints and assumptions

4.1 Limitations

No limitations.

4.2 Applicability to car domains

No limitations.

4.3 Applicability to safety related environments

No restrictions, because the subject of this specification is a header file specifying types. It does not include or implement any functionality.

5 Software Architecture

5.1 Dependencies to other modules

The communication stack types header file defines communication types based on the platform types [PltfTypes] (Platform_Types.h) and Compiler (Compiler.h) header file [CompTypes]. To prevent multiple includes of header files, the communication stack header file includes the standard types header file [StdTypes] which already includes both other files.

5.2 File structure

COMTYPE001: The include file structure shall be as follows:

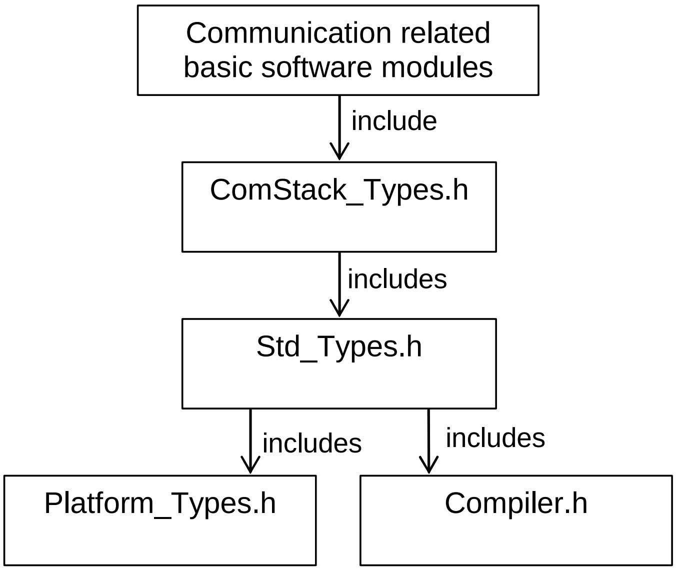

Figure 1: Include File Structure

`ComStack_Types.h` shall include `Std_Types.h`
`Std_Types.h` shall include `Platform_Types.h`
`Std_Types.h` shall include `Compiler.h`
Communication related basic software modules shall include `ComStack_Types.h`
Communication related basic software modules shall <u>not</u> include `Std_Types.h` directly.

6 Requirements traceability

Document: AUTOSAR general requirements on Basic Software Modules [GeneralSRS]

Requirement	***Satisfied by***
[BSW00344] Reference to link-time configuration	Not applicable (this is only a header file specification)
[BSW00404] Reference to post build time configuration	Not applicable (this is only a header file specification)
[BSW00405] Reference to multiple configuration sets	Not applicable (this is only a header file specification)
[BSW00345] Pre-compile-time configuration	Not applicable (this is only a header file specification)
[BSW159] Tool-based configuration	Not applicable (this is a tool requirement)
[BSW167] Static configuration checking	Not applicable (this is only a header file specification)
[BSW171] Configurability of optional functionality	Not applicable (this is only a header file specification)
[BSW170] Data for reconfiguration of AUTOSAR SW-Components	Not applicable (this is only a header file specification)
[BSW00380] Separate C-File for configuration parameters	Not applicable (this is only a header file specification)
[BSW00381] Separate configuration header file for pre-compile time parameters	Not applicable (this is only a header file specification)
[BSW00412] Separate H-File for configuration parameters [approved]	Not applicable (this is only a header file specification)
[BSW00383] List dependencies of configuration files	Not applicable (this is only a header file specification)
[BSW00384] List dependencies to other modules	COMTYPE001
[BSW00387] Specify the configuration class of callback function	Not applicable (this is only a header file specification)
[BSW00388] Introduce containers	Not applicable (this is only a header file specification)
[BSW00389] Containers shall have names	Not applicable (this is only a header file specification)
[BSW00390] Parameter content shall be unique within the module	Not applicable (this is only a header file specification)
[BSW00391] Parameter shall have unique names	Not applicable (this is only a header file specification)
[BSW00392] Parameters shall have a type	Not applicable (this is only a header file specification)
[BSW00393] Parameters shall have a range	Not applicable (this is only a header file specification)
[BSW00394] Specify the scope of the parameters	Not applicable (this is only a header file specification)
[BSW00395] List the required parameters (per parameter)	Not applicable (this is only a header file specification)
[BSW00396] Configuration classes	Not applicable (this is only a header file specification)
[BSW00397] Pre-compile-time parameters	Not applicable (this is only a header file specification)
[BSW00398] Link-time parameters	Not applicable (this is only a header file specification)
[BSW00399] Loadable Post-build time parameters	Not applicable

Requirement	*Satisfied by*
	(this is only a header file specification)
[BSW00400] Selectable Post-build time parameters	Not applicable (this is only a header file specification)
[BSW00402] Published information	Partly fulfilled by COMTYPE002. Vendor version number for this header file not necessary.
[BSW00375] Notification of wake-up reason	Not applicable (this is only a header file specification)
[BSW101] Initialization interface	Not applicable (this is only a header file specification)
[BSW00416] Sequence of Initialization	Not applicable (this is only a header file specification)
[BSW00406] Check module initialization	Not applicable (this is only a header file specification)
[BSW168] Diagnostic Interface of SW components	Not applicable (this is only a header file specification)
[BSW00407] Function to read out published parameters	Not applicable (this is only a header file specification)
[BSW00423] Usage of SW-C template to describe BSW modules with AUTOSAR Interfaces	Not applicable (this is only a header file specification)
[BSW00424] BSW main processing function task allocation	Not applicable (this is only a header file specification)
[BSW00425] Trigger conditions for schedulable objects	Not applicable (this is only a header file specification)
[BSW00426] Exclusive areas in BSW modules	Not applicable (this is only a header file specification)
[BSW00427] ISR description for BSW modules	Not applicable (this is only a header file specification)
[BSW00428] Execution order dependencies of main processing functions	Not applicable (not related to this specification)
[BSW00429] Restricted BSW OS functionality access	Not applicable (this is only a header file specification)
[BSW00431] The BSW Scheduler module implements task bodies	Not applicable (not related to this specification)
[BSW00432] Modules should have separate main processing functions for read/receive and write/transmit data path	Not applicable (this is only a header file specification)
[BSW00433] Calling of main processing functions	Not applicable (not related to this specification)
[BSW00434] The Schedule Module shall provide an API for exclusive areas	Not applicable (not related to this specification)
[BSW00336] Shutdown interface	Not applicable (this is only a header file specification)
[BSW00337] Classification of errors	Not applicable (this is only a header file specification)
[BSW00338] Detection and Reporting of development errors	Not applicable (this is only a header file specification)
[BSW00369] Do not return development error codes via API	Not applicable (this is only a header file specification)
[BSW00339] Reporting of production relevant error status	Not applicable (this is only a header file specification)
[BSW00421] Reporting of production relevant error events	Not applicable (this is only a header file specification)
[BSW00422] Debouncing of production relevant error status	Not applicable (not related to this specification)
[BSW00420] Production relevant error event rate detection	Not applicable (not related to this specification)
[BSW00417] Reporting of Error Events by Non-	Not applicable

Requirement	***Satisfied by***
Basic Software	(this is only a header file specification)
[BSW00323] API parameter checking	Not applicable (this is only a header file specification)
[BSW004] Version check	Check has to be done by a specific tool. Version numbers provided by COMTYPE002.
[BSW00409] Header files for production code error IDs	Not applicable (this is only a header file specification)
[BSW00385] List possible error notificatons	Not applicable (this is only a header file specification)
[BSW00386] Configuration for detecting an error	Not applicable (this is only a header file specification)
[BSW161] Microcontroller abstraction	Not applicable (this is only a header file specification)
[BSW162] ECU layout abstraction	Not applicable (requirement on AUTOSAR architecture, not a single module)
[BSW00324] Do not use HIS I/O Library	Not applicable (architecture decision)
[BSW005] No hard coded horizontal interfaces within MCAL	Not applicable (requirement on AUTOSAR architecture, not a single module)
[BSW00415] User dependent include files	Not applicable (only one user for this module)
[BSW164] Implementation of interrupt service routines	Not applicable (this module does not implement any ISRs)
[BSW00325] Runtime of interrupt service routines	Not applicable (this module does not implement any ISRs or callback routines)
[BSW00326] Transition from ISRs to OS tasks	Not applicable (requirement on implementation, not on specification)
[BSW00342] Usage of source code and object code	Not applicable (requirement on AUTOSAR architecture, not a single module)
[BSW00343] Specification and configuration of time	Not applicable (this module does not provide any timing configuration)
[BSW160] Human-readable configuration data	Not applicable (requirement on documentation, not on specification)
[BSW007] HIS MISRA C	Not applicable (requirement on implementation, not on specification)
[BSW00300] Module naming convention	Not applicable (requirement on implementation, not on specification)
[BSW00413] Accessing instances of BSW modules	Requirement can not be implemented in R2.0 timeframe.
[BSW00347] Naming separation of different instances of BSW drivers	Not applicable (requirement on the implementation, not on the specification)
[BSW00305] Self-defined data types naming convention	Chapter 8.1
[BSW00307] Global variables naming convention	Not applicable (requirement on the implementation, not on the specification)
[BSW00310] API naming convention	Not applicable

Requirement	***Satisfied by***
	(this is only a header file specification)
[BSW00373] Main processing function naming convention	Not applicable (this module does not provide a scheduled function)
[BSW00327] Error values naming convention	Not applicable (this is only a header file specification)
[BSW00335] Status values naming convention	Not applicable (this is only a header file specification)
[BSW00350] Development error detection keyword	Not applicable (this is only a header file specification)
[BSW00408] Configuration parameter naming convention	Not applicable (this is only a header file specification)
[BSW00410] Compiler switches shall have defined values	Not applicable (this is only a header file specification)
[BSW00411] Get version info keyword	Not applicable (this is only a header file specification)
[BSW00346] Basic set of module files	Not applicable (this is only a header file specification)
[BSW158] Separation of configuration from implementation	Not applicable (this is only a header file specification)
[BSW00314] Separation of interrupt frames and service routines	Not applicable (this is only a header file specification)
[BSW00370] Separation of callback interface from API	Not applicable (this module does not implement any callback routines)
[BSW00348] Standard type header	Not applicable (requirement on the standard header file)
[BSW00353] Platform specific type header	Not applicable (requirement on the platform specific header file)
[BSW00361] Compiler specific language extension header	Not applicable (requirement on the compiler specific header file)
[BSW00301] Limit imported information	Not applicable (this is only a header file specification)
[BSW00302] Limit exported information	Not applicable (requirement on the implementation, not on the specification)
[BSW00328] Avoid duplication of code	Not applicable (requirement on the implementation, not on the specification)
[BSW00312] Shared code shall be reentrant	Not applicable (requirement on the implementation, not on the specification)
[BSW006] Platform independency	Not applicable (this is a module of the microcontroller abstraction layer)
[BSW00357] Standard API return type	Not applicable (this is only a header file specification)
[BSW00377] Module specific API return types	Not applicable (this is only a header file specification)
[BSW00304] AUTOSAR integer data types	Not applicable (requirement on implementation, not for specification)
[BSW00355] Do not redefine AUTOSAR integer data types	Not applicable (requirement on implementation, not for specification)
[BSW00378] AUTOSAR boolean type	Not applicable (requirement on implementation, not for

Requirement	*Satisfied by*
	specification)
[BSW00306] Avoid direct use of compiler and platform specific keywords	Not applicable (requirement on implementation, not for specification)
[BSW00308] Definition of global data	Not applicable (requirement on implementation, not for specification)
[BSW00309] Global data with read-only constraint	Not applicable (requirement on implementation, not for specification)
[BSW00371] Do not pass function pointers via API	Not applicable (no function pointers in this specification)
[BSW00358] Return type of init() functions	Not applicable (this module does not provide an initialization function)
[BSW00414] Parameter of init function	Not applicable (this module does not provide an initialization function)
[BSW00376] Return type and parameters of main processing functions	Not applicable (this module does not provide a scheduled function)
[BSW00359] Return type of callback functions	Not applicable (this module does not provide any callback routines)
[BSW00360] Parameters of callback functions	Not applicable (this module does not provide any callback routines)
[BSW00329] Avoidance of generic interfaces	Not applicable (this is only a header file specification)
[BSW00330] Usage of macros / inline functions instead of functions	Not applicable (requirement on implementation, not for specification)
[BSW00331] Separation of error and status values	Not applicable (this is only a header file specification)
[BSW009] Module User Documentation	Not applicable (requirement on documentation, not on specification)
[BSW00401] Documentation of multiple instances of configuration parameters	Not applicable (all configuration parameters are single instance only)
[BSW172] Compatibility and documentation of scheduling strategy	Not applicable (no internal scheduling policy)
[BSW010] Memory resource documentation	Not applicable (requirement on documentation, not on specification)
[BSW00333] Documentation of callback function context	Not applicable (requirement on documentation, not for specification)
[BSW00374] Module vendor identification	Not applicable (this module is a standardized module)
[BSW00379] Module identification	Not applicable (this is only a header file specification)
[BSW003] Version identification	COMTYPE002
[BSW00318] Format of module version numbers	COMTYPE002
[BSW00321] Enumeration of module version numbers	Not applicable (requirement on implementation, not for specification)

Requirement	***Satisfied by***
[BSW00341] Microcontroller compatibility documentation	Not applicable (requirement on documentation, not on specification)
[BSW00334] Provision of XML file	Not applicable (requirement on documentation, not on specification)

7 Functional specification

7.1 General issues

COMTYPE003: The file name of the communication stack types header file shall be 'ComStack_Types.h'.

COMTYPE004: It is not allowed to add any project or supplier specific extension to this file. Any extension invalidates the AUTOSAR conformity.

COMTYPE015: Because many of the communication stack types are depending on the appropriate ECU, this file shall be generated dependent on the specific ECU configuration for each ECU independently.

COMTYPE016: The communication stack types header file shall be protected against multiple inclusion:

```
#ifndef COMSTACK_TYPES_H
#define COMSTACK_TYPES_H
..
/*
 * Contents of file
 */
..
#endif /* COMSTACK_TYPES_H */
```

8 API specification

8.1 Type definitions

8.1.1 PduIdType

Name:	`PduIdType`		
Type:	`uint8, uint16`		
Range:	`0..<PduIdmax>`	--	Zero-based integer number The size of this global type depends on the maximum number of PDUs used within one software module. Example: If no software module deals with more PDUs that 256, this type can be set to uint8. If at least one software module handles more than 256 PDUs, this type must globally be set to uint16.
Description:	COMTYPE005: This type is used within the entire AUTOSAR Com Stack except for bus drivers. COMTYPE006: Variables of this type serve as a unique identifier of a PDU within a software module or a set thereof, and also for interaction of two software modules where the PduId of the corresponding target module is being used for referencing. COMTYPE007: In order to be able to perform table-indexing within a software module, variables of this type shall be zero-based and consecutive. There might be several ranges of PduIds in a module, one for each type of operation performed within that module (e.g. sending and receiving). COMTYPE014: PduIdmax, the maximum number of a PduId range, is the number -1 of PDUs dealt with in the corresponding type of operation within that module.		

8.1.2 PduLengthType

Name:	`PduLengthType`		
Type:	`uint8, uint16, uint32`		
Range:	`0..<PduLengthmax>`	--	Zero-based integer number The size of this global type depends on the maximum length of PDUs to be sent by an ECU. Example: If no segmentation is used the length depends on the maximum payload size of a frame of the underlying communication system (for FlexRay maximum size is 255, therefore uint8). If segementation is used it depends on the maximum length of a segmeneted N-PDU (in general uint16 is used)
Description:	COMTYPE008: This type shall be used within the entire AUTOSAR Com Stack of an ECU except for bus drivers. COMTYPE010: Variables of this type serve as length information of a PDU. The length information is provided in number of bytes. COMTYPE017: PduLengthmax, the maximum length of a Pdu, is the length of the largest (possibly segmented) PDU to be sent by the ECU.		

8.1.3 PduInfoType

Name:	PduInfoType		
Type:	Structure		
Element:	uint8*	SduDataPtr	uint8-pointer to the SDU (i.e. payload data) of the PDU. The type of this pointer depends on the memory model being used at compile time. P2VAR(uint8,AUTOMATIC,AUTOSAR_COMSTACKDATA)
	PduLengthType	SduLength	length of the SDU in bytes
Description:	COMTYPE011: Variables of this type shall be used to store the basic information about a PDU of any type, namely a pointer variable pointing to it's SDU (payload), and the corresponding length of the SDU in bytes.		

8.1.4 TPParameterType

Name:	TPParameterType	
Type:	Enumeration	
Range:	TP_STMIN	0x00: Separation Time
	TP_BS	0x01: Block Size
	TP_BC	0x02: The Band width control parameter used in FlexRay transport protocol module.
Description:	COMTYPE028: Specify the parameter to which the value has to be changed (BS or STmin).	

8.1.5 BufReq_ReturnType

Name:	BufReq_ReturnType	
Type:	Enumeration	
Range:	BUFREQ_OK	Buffer request accomplished successful. This status shall have the value 0.
	BUFREQ_E_NOT_OK	Buffer request not successful. Buffer cannot be accessed. This status shall have the value 1.
	BUFREQ_E_BUSY	Temporarily no buffer available. It's up the requestor to retry request for a certain time. This status shall have the value 2.
	BUFREQ_E_OVFL	No Buffer of the required length can be provided. This status shall have the value 3.
Description:	COMTYPE012: Variables of this type shall be used to store the result of a buffer request.	

8.1.6 NotifResultType

Name:	NotifResultType		
Type:	uint8		
Range:	0x00 - 0x1E	--	General return codes. A detailed specification is listed below.
	0x1f - 0xff	--	Currently values in this range are invalid.
Description:	COMTYPE013: Variables of this type shall be used to store the result status of a notification (confirmation or indication).		

	(Currently this type is only used for communication between DCM and TP to enable the notification that an error has occurred and a dedicated buffer can be unlocked.)

Return code specification:

COMTYPE018: General Codes

Return code	***Value***	***Description***
`NTFRSLT_OK`	`0x00`	Action has been successfully finished: • message sent out (in case of confirmation), • message received (in case of indication)
`NTFRSLT_E_NOT_OK`	`0x01`	Error notification: • message not successfully sent out (in case of confirmation), • message not successfully received (in case of indication)
`NTFRSLT_E_TIMEOUT_A`	`0x02`	Error notification: • timer N_Ar/N_As (according to ISO specification [ISONM]) has passed its time-out value N_Asmax/N_Armax. This value can be issued to service user on both the sender and receiver side.
`NTFRSLT_E_TIMEOUT_BS`	`0x03`	Error notification: timer N_Bs has passed its time-out value N_Bsmax (according to ISO specification [ISONM]). This value can be issued to the service user on the sender side only.
`NTFRSLT_E_TIMEOUT_CR`	`0x04`	Error notification: timer N_Cr has passed its time-out value N_Crmax. This value can be issued to the service user on the receiver side only.
`NTFRSLT_E_WRONG_SN`	`0x05`	Error notification: unexpected sequence number (PCI.SN) value received. This value can be issued to the service user on the receiver side only.
`NTFRSLT_E_INVALID_FS`	`0x06`	Error notification: invalid or unknown FlowStatus value has been received in a flow control (FC) N_PDU. This value can be issued to the service user on the sender side only.
`NTFRSLT_E_UNEXP_PDU`	`0x07`	Error notification: unexpected protocol data unit received.

Return code	***Value***	***Description***
		This value can be issued to the service user on both the sender and receiver side.
NTFRSLT_E_WFT_OVRN	0x08	Error notification: flow control WAIT frame that exceeds the maximum counter N_WFTmax received. This value can be issued to the service user on the receiver side.
NTFRSLT_E_NO_BUFFER	0x09	Error notification: flow control (FC) N_PDU with FlowStatus = OVFLW received. It indicates that the buffer on the receiver side of a segmented message transmission cannot store the number of bytes specified by the FirstFrame DataLength (FF_DL) parameter in the FirstFrame and therefore the transmission of the segmented message was aborted. no buffer within the TP available to transmit the segmented I-PDU. This value can be issued to the service user on both the sender and receiver side.
	0x0C-0x1E	Reserved values for future usage.

COMTYPE019: The Communication System dependent Return codes shall be named as follows:
NTFRSLT_E_<Communication System Abbreviation>_<Error Code Name>.
Communication System Abbreviation:
CAN: for Controller area network
LIN: for Local Interconnect Network
FR: for FlexRay
Error Code Name: self explaining name of error return code.

Example for a FlexRay specific return value:
NTFRSLT_E_FR_NEG_ACK: Negative acknowledgement on received

8.1.7 BusTrcvErrorType

Name:	BusTrcvErrorType		
Type:	uint8		
Range:	0x00 – 0x1E	--	General return codes. A detailed specification is listed below.
	0x1F – 0x3C	--	Error notification: Error notification codes specific for the communication system CAN. For a detailed definition please refer to the AUTOSAR specification of CAN Transceiver Driver [CANTRCV].
	0x3D – 0x5A	--	Error notification: Error notification codes specific for the communication

			system LIN. A detailed definition is still open, because currently there is not AUTOSAR specification of Lin Interface.
	0x5B – 0x78	--	Error notification: Error notification codes specific for the communication system FlexRay. For a detailed definition please refer to the AUTOSAR specification of FlexRay Transceiver Driver [FRTRCV].
	> 0x78	--	Currently values in this range are invalid. In future it might be possible that further return codes are specified for other communication systems.
Description:	COMTYPE020: Variables of this type shall be used to return the bus status evaluated by a transceiver.		

Return code specification:
COMTYPE021: General Codes

Return code	***Value***	***Description***
BUSTRCV_OK	0x00	There is no bus transceiver error seen by the driver or transceiver does not support the detection of bus errors.
BUSTRCV_E_ERROR	0x01	Bus transceiver detected an unclassified error.
	0x02-0x1E	Reserved values for future usage.

COMTYPE022: The Communication System dependent Return codes shall be named as follows:
BUSTRCV_E_<Communication System Abbreviation>_<Error Code Name>.
Communication System Abbreviation:
CAN: for Controller area network
LIN: for Local Interconnect Network
FR: for FlexRay
Error Code Name: self explaining name of error return code.

Example for a CAN specific return value:
BUSTRCV_E_CAN_SINGLE: CAN bus transceiver has detected that the fault tolerant bus is in single wire mode.

8.1.8 NetworkHandleType

Name:	NetworkHandleType
Type:	uint8
Description:	Zero-based integer number. COMTYPE026: Variables of the type NetworkHandleType shall be used to store the identifier of a communication channel.

Name:	NetworkHandleType
Type:	uint8
Description:	Zero-based integer number. COMTYPE026: Variables of the type NetworkHandleType shall be used to store the identifier of a communication channel.

8.1.9 PNCHandleType

Name:	PNCHandleType
Type:	uint8
Description:	Used to store the identifier of a partial network cluster.

8.2 Function definitions

Not applicable.

9 Sequence diagrams

Not applicable.

10 Configuration specification

10.1 Published parameters

COMTYPE002: The following table specifies parameters that shall be published within the communication types header file (“`ComStack_Types.h`”).

The standard common published information like

vendorId (COMSTACKTYPE_VENDOR_ID),
moduleId (COMSTACKTYPE _MODULE_ID),
arMajorVersion (COMSTACKTYPE_AR_MAJOR_VERSION),
arMinorVersion (COMSTACKTYPE_AR_MINOR_VERSION),
arPatchVersion (COMSTACKTYPE_AR_PATCH_VERSION),
swMajorVersion (COMSTACKTYPE_SW_MAJOR_VERSION),
swMinorVersion (COMSTACKTYPE_SW_MINOR_VERSION),
swPatchVersion (COMSTACKTYPE_SW_PATCH_VERSION),
vendorApiInfix (COMSTACKTYPE_VENDOR_API_INFIX)

is provided in the BSW Module Description Template (see 3.1 Figure 4.1 and Figure 7.1).

Additional published parameters are listed below if applicable for this module.