

Document Title	Specification of Software Cluster Connection module
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	974

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R21-11

Document Change History			
Date	Release	Changed by	Description
2021-11-25	R21-11	AUTOSAR Release Management	added support for <ul style="list-style-type: none"> • signal and SOME/IP based communication from Software Components located in an Applicative Software Cluster • connection to diagnostic services as Dem, FiM, Dcm. • Software Cluster Service Resources for connection of WdgM instances in different Software Clusters
2020-11-30	R20-11	AUTOSAR Release Management	Initial release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1	Introduction and functional overview	15
2	Acronyms and abbreviations	15
3	Related documentation	19
3.1	Input documents & related standards and norms	19
3.2	Related specification	20
4	Constraints and assumptions	20
4.1	Limitations	20
4.2	Applicability to car domains	20
5	Dependencies to other modules	21
5.1	Binary Manifest	21
5.1.1	Header File structure	21
6	Requirements Tracing	21
7	Functional specification	35
7.1	Binary Manifest	35
7.1.1	Overview	35
7.1.2	Logical structure of a Binary Manifest	36
7.1.3	Mapping between Logical structure and Configuration structure	38
7.1.4	Implementation structure of a Binary Manifest	40
7.1.4.1	Implementation Overview	42
7.1.4.2	Multiple Notifier Sets Introduction	43
7.1.4.3	Binary Manifest Header	44
7.1.4.4	Interface Descriptor Table	46
7.1.4.5	Offered Interface Table	51
7.1.4.6	Subscribed Interface Table	51
7.1.4.7	Administrative Data	52
7.1.4.8	Memory Mapping	53
7.1.5	Connecting Software Clusters	55
7.1.5.1	Overview	55
7.1.5.2	Connecting Resources	56
7.1.5.3	Handling of the Substitution Software Cluster	57
7.1.5.4	Multiple Notifier Sets	58
7.1.5.5	Unconnected Resources	59
7.1.5.6	Disabling of the On-board Software Cluster Connection	60
7.1.5.7	Errors during software cluster connection	60
7.1.6	Software Cluster Binary Manifest Descriptor	61
7.1.7	Error Classification	62
7.1.7.1	Development Errors	62
7.1.7.2	Runtime Errors	62

7.1.7.3	Transient Faults	62
7.1.7.4	Production Errors	62
7.1.7.5	Extended Production Errors	63
7.2	Software Cluster Base Configuration Check	63
7.3	Cross Cluster Communication	64
7.3.1	Overview	64
7.3.2	General Requirements	65
7.3.2.1	Cross Software Cluster Communication Plug-In	65
7.3.2.2	Cross Cluster Communication and Binary Manifest	66
7.3.2.3	Cross Cluster Communication Base Socket	79
7.3.2.4	Cross Cluster Communication and McSupport Data	79
7.3.3	Sender Receiver Communication	81
7.3.3.1	Restrictions on VFB communication features	81
7.3.3.2	Transmission	81
7.3.3.3	Reception	83
7.3.4	Client Server Communication	87
7.3.4.1	General	87
7.3.4.2	Timeout	87
7.3.4.3	Buffering	87
7.3.4.4	Response to Request Mapping	89
7.3.4.5	Client Side	90
7.3.4.6	Server Side	91
7.3.5	Modes Communication	91
7.3.5.1	General principles	91
7.3.5.2	Software Cluster providing a mode	92
7.3.5.3	Host Software Cluster requiring a mode	97
7.3.5.4	Applicative Software Cluster requiring a mode	97
7.3.5.5	Initialization	99
7.3.6	Trigger Communication	100
7.3.6.1	General principles	100
7.3.6.2	Software Cluster providing a mode	100
7.3.6.3	Host Software Cluster requiring a trigger	104
7.3.6.4	Applicative Software Cluster requiring a trigger	104
7.3.7	Parameter Communication	106
7.3.8	Error Classification	107
7.3.8.1	Development Errors	107
7.3.8.2	Runtime Errors	107
7.3.8.3	Transient Faults	107
7.3.8.4	Production Errors	107
7.3.8.5	Extended Production Errors	107
7.4	Proxy Modules	108
7.4.1	Overview	108
7.4.2	Abstract Proxy Module Pattern	110
7.4.2.1	General Proxy functionality	110
7.4.2.2	Partitions	111
7.4.2.3	Unconnected Service Resources	113

7.4.2.4	Ecu Configuration Principles	113
7.4.2.5	Proxy Modules and Binary Manifest	114
7.4.3	Specific Proxy Module Requirements	115
7.4.3.1	OS Proxy	115
7.4.3.2	NvM Proxy	126
7.4.3.3	Com Proxy	134
7.4.3.4	LdCom Proxy	149
7.4.3.5	Dcm Proxy	162
7.4.3.6	Dem Proxy	182
7.4.3.7	FiM Proxy	190
7.4.4	Error Classification	193
7.4.4.1	Development Errors	193
7.4.4.2	Runtime Errors	195
7.4.4.3	Transient Faults	195
7.4.4.4	Production Errors	196
7.4.4.5	Extended Production Errors	196
7.5	Standardized Service Resources	196
7.5.1	Software Cluster Base Configuration Check	197
7.5.2	Cross Cluster Communication	197
7.5.3	OS Proxy	198
7.5.4	NvM Proxy	200
7.5.5	Com Proxy	201
7.5.6	LdCom Proxy	202
7.5.7	Dcm Proxy	202
7.5.8	Dem Proxy	203
7.5.9	FiM Proxy	204
7.5.10	WdgM	205
8	API specification	206
8.1	Imported types	206
8.2	Type definitions	206
8.2.1	Binary Manifest	206
8.2.1.1	SwCluC_BManif_SwClusterIdType	206
8.2.1.2	SwCluC_BManif_MachineIdType	206
8.2.1.3	SwCluC_BManif_ConCtrlType	207
8.2.1.4	SwCluC_BManif_ResourcePropertiesType	207
8.2.1.5	SwCluC_BManif_ResourceTypeIdType	208
8.2.1.6	SwCluC_BManif_GlobalResourceIdType	208
8.2.1.7	SwCluC_BManif_ResourceGuardValueType	208
8.2.1.8	SwCluC_BManif_TableIndexType	208
8.2.1.9	SwCluC_BManif_HandleIndexType	209
8.2.1.10	SwCluC_BManif_VoidFncPtrType	209
8.2.1.11	SwCluC_BManif_HandleType	210
8.2.1.12	SwCluC_BManif_HeaderType	210
8.2.2	Cross Cluster Connection	212
8.2.3	ProxyModules	212

8.3	Function definitions	213
8.3.1	BinaryManifest	213
8.3.1.1	API Principles	213
8.3.1.2	SwCluC_BManif_GetHandle	215
8.3.1.3	SwCluC_BManif_GetHandle	216
8.3.1.4	SwCluC_BManif_GetConSwClusterId	216
8.3.1.5	SwCluC_BManif_GetConSwClusterId	217
8.3.1.6	SwCluC_BManif_GetNoOfHandleSets	218
8.3.1.7	SwCluC_BManif_GetNoOfHandleSets	219
8.3.1.8	SwCluC_BManif_GetValidityMarker	220
8.3.1.9	SWCLUC_BMANIF_NO_OF_ENTRIES	220
8.3.1.10	SWCLUC_BMANIF_NO_OF_NOTIFIER_SETS	220
8.3.2	Cross Cluster Communication	221
8.3.2.1	SwCluC_Xcc_Init1	221
8.3.2.2	SwCluC_Xcc_Init2	221
8.3.3	Proxy Modules	222
8.3.3.1	SwCluC_OsProxy_Init	222
8.3.3.2	SwCluC_NvMProxy_Init	223
8.3.3.3	SwCluC_LdComProxy_Init	223
8.3.3.4	SwCluC_ComProxy_Init	224
8.3.3.5	SwCluC_DcmProxy_Init	224
8.3.3.6	SwCluC_DemProxy_Init	225
8.3.3.7	SwCluC_FiMProxy_Init	225
8.4	Callback notifications	226
8.4.1	Binary Manifest	226
8.4.2	Cross Cluster Connection	226
8.4.3	Proxy Modules	226
8.5	Scheduled functions	235
8.5.1	Binary Manifest	235
8.5.2	Cross Cluster Communication Scheduled functions	235
8.5.3	Proxy Modules Scheduled functions	235
8.6	Expected interfaces	236
8.6.1	Mandatory interfaces	236
8.6.1.1	Binary Manifest	236
8.6.1.2	Cross Cluster Connection	236
8.6.1.3	Proxy Modules	236
8.6.2	Optional interfaces	236
8.6.2.1	Binary Manifest	236
8.6.2.2	Cross Cluster Connection	236
8.6.2.3	Proxy Modules	237
8.6.3	Configurable interfaces	237
8.6.3.1	Binary Manifest	237
8.6.3.2	Cross Cluster Connection	237
8.6.3.3	Proxy Modules	237
8.7	Service Interfaces	237
8.7.1	Binary Manifest	237

8.7.2	Cross Cluster Connection	237
8.7.3	Proxy Modules	237
9	Sequence diagrams	238
10	Configuration specification	238
10.1	How to read this chapter	238
10.2	Containers and configuration parameters	238
10.2.1	Module Configuration	239
10.2.2	General configuration parameters	240
10.2.3	Software Cluster Definition	242
10.2.4	Software Cluster Base Configuration Check	247
10.2.5	Binary Manifest	249
10.2.5.1	Provide Resource Entry Group	252
10.2.5.2	Provided Resource Entry	254
10.2.5.3	Require Resource Entry Group	257
10.2.5.4	Require Resource Entry	259
10.2.5.5	Resource Type	262
10.2.6	Cross Cluster Communication	267
10.2.6.1	Cross Cluster Communication Base Socket	269
10.2.6.2	Cross Cluster Communication Rte Plug-in Props	270
10.2.7	Proxy Modules	272
10.2.7.1	NvM Proxy	278
10.2.7.2	Os Proxy	287
10.2.7.3	Com Proxy	309
10.2.7.4	LdCom Proxy	322
10.2.7.5	Dcm Proxy	335
10.2.7.6	Dem Proxy	347
10.2.7.7	FiM Proxy	354
10.3	Published Information	361
A	Not applicable requirements	361
B	Referenced Meta Classes	362
C	Referenced ECUC Configuration Parameters	434
C.1	EcuC	434
C.1.1	EcucPartition	434
C.1.2	Pdu	437
C.1.3	EcucPduDedicatedPartition	440
C.2	RTE	441
C.2.1	RteEventToTaskMapping	441
C.3	Os	450
C.3.1	OsAlarm	450
C.3.2	OsApplication	452
C.3.3	OsCounter	457
C.3.4	OsEvent	460

C.3.5	OsResource	460
C.3.6	OsScheduleTable	462
C.3.7	OsScheduleTableExpiryPoint	464
C.3.8	OsSpinlock	465
C.3.9	OsTask	466
C.4	NvM	470
C.4.1	NvMBlockDescriptor	470
C.4.2	NvMInitBlockCallback	488
C.4.3	NvMSingleBlockCallback	489
C.5	Dcm	490
C.5.1	DcmDspDid	490
C.5.2	DcmDspDidInfo	493
C.5.3	DcmDspData	494
C.5.4	DcmDspPidService01	506
C.5.5	DcmDspDidRead	508
C.5.6	DcmDspDidWrite	510
C.5.7	DcmDspDidControl	512
C.6	Dem	516
C.6.1	DemEventParameter	516
C.6.2	DemDataElementClass	527
C.7	Com	527
C.7.1	ComMainFunctionTx	527
C.7.2	ComMainFunctionRx	529
C.7.3	ComGroupSignal	530
C.7.4	ComIPdu	538
C.7.5	ComSignal	544
C.7.6	ComUserModule	558
C.7.7	ComUserUriDefSet	558
C.7.8	ComSignalGroup	559
C.8	LdCom	567
C.8.1	LdComConfig	568
C.8.2	LdComIPdu	568
C.8.3	LdComUserModule	575
C.8.4	LdComUserUriDefSet	576
D	Referenced C-API	577
D.1	RTE	577
D.1.1	Rte_Rips_DatalsUpdated	577
D.1.2	Rte_Rips_DRead	577
D.1.3	Rte_Rips_DatalsUpdated_EventActivation	578
D.1.4	Rte_Rips_Feedback	578
D.1.5	Rte_Rips_Invoke	579
D.1.6	Rte_Rips_Prm	579
D.1.7	Rte_Rips_Read	580
D.1.8	Rte_Rips_ReturnResult	580
D.1.9	Rte_Rips_Start	581

D.1.10	Rte_Rips_Stop	581
D.1.11	Rte_Rips_SchM_Deinit	581
D.1.12	Rte_Rips_SchM_Init	582
D.1.13	Rte_Rips_SwitchNotificationStatusType	582
D.1.14	Rte_Rips_Switch	583
D.1.15	Rte_Rips_DequeueModeSwitch	583
D.1.16	Rte_Rips_Trigger	584
D.1.17	Rte_Rips_Write	584
D.2	OS	585
D.3	NvM	586
D.3.1	NvM_CancelJobs	586
D.3.2	NvM_EraseNvBlock	586
D.3.3	NvM_GetDataIndex	587
D.3.4	NvM_GetErrorStatus	587
D.3.5	NvM_InvalidateNvBlock	588
D.3.6	NvM_ReadBlock	588
D.3.7	NvM_ReadPRAMBlock	588
D.3.8	NvM_RestoreBlockDefaults	589
D.3.9	NvM_RestorePRAMBlockDefaults	589
D.3.10	NvM_SetBlockLockStatus	590
D.3.11	NvM_SetBlockProtection	590
D.3.12	NvM_SetDataIndex	591
D.3.13	NvM_SetRamBlockStatus	591
D.3.14	NvM_WriteBlock	592
D.3.15	NvM_WritePRAMBlock	592
D.3.16	NvM_SingleBlockCallbackFunction	593
D.3.17	NvM_InitBlockCallbackFunction	593
D.3.18	NvM_ReadRamBlockFromNvm	594
D.3.19	NvM_WriteRamBlockToNvm	594
D.4	Com	595
D.4.1	Com_SendSignal	595
D.4.2	Com_SendDynSignal	595
D.4.3	Com_SendSignalGroup	596
D.4.4	Com_SendSignalGroupArray	596
D.4.5	Com_SendSignalWithMetaData	597
D.4.6	Com_SendDynSignalWithMetaData	597
D.4.7	Com_SendSignalGroupWithMetaData	598
D.4.8	Com_SendSignalGroupArrayWithMetaData	598
D.4.9	Com_ReceiveSignal	599
D.4.10	Com_ReceiveDynSignal	599
D.4.11	Com_ReceiveSignalGroup	600
D.4.12	Com_ReceiveSignalGroupArray	601
D.4.13	Com_ReceiveSignalWithMetaData	601
D.4.14	Com_ReceiveDynSignalWithMetaData	602
D.4.15	Com_ReceiveSignalGroupWithMetaData	602
D.4.16	Com_ReceiveSignalGroupArrayWithMetaData	603

D.4.17	Com_InvalidateSignal	603
D.4.18	Com_InvalidateSignalGroup	604
D.5	LdCom	604
D.5.1	LdCom_Transmit	604
D.6	Dcm	605
D.6.1	DataServices	605
D.6.1.1	ReadData	605
D.6.1.2	WriteData	606
D.6.1.3	ConditionCheckRead	607
D.6.1.4	ReadDataLength	608
D.6.1.5	GetScalingInformation	609
D.6.1.6	ReturnControlToECU	610
D.6.1.7	ResetToDefault	611
D.6.1.8	FreezeCurrentState	612
D.6.1.9	ShortTermAdjustment	614
D.6.2	RoutineServices	615
D.6.2.1	Xxx_Start Operation	615
D.6.2.2	Xxx_StartConfirmation Operation	616
D.6.2.3	Xxx_Stop Operation	616
D.6.2.4	Xxx_StopConfirmation Operation	617
D.6.2.5	Xxx_RequestResults Operation	618
D.6.2.6	Xxx_RequestResultsConfirmation Operation	618
D.6.3	ServiceRequestNotification	619
D.6.3.1	Indication	619
D.6.3.2	Confirmation	620
D.7	Dem	620
D.7.1	Dem_ClearPrestoredFreezeFrame	620
D.7.2	Dem_PrestoreFreezeFrame	621
D.7.3	Dem_ResetEventDebounceStatus	621
D.7.4	Dem_ResetEventStatus	622
D.7.5	Dem_SetEventDisabled	622
D.7.6	Dem_SetEventStatus	623
D.7.7	Dem_GetDTCOfEvent	623
D.7.8	Dem_GetDebouncingOfEvent	624
D.7.9	Dem_GetEventAvailable	624
D.7.10	Dem_GetEventExtendedDataRecordEx	625
D.7.11	Dem_GetEventFreezeFrameDataEx	625
D.7.12	Dem_GetEventUdsStatus	626
D.7.13	Dem_GetFaultDetectionCounter	626
D.7.14	Dem_GetMonitorStatus	627
D.7.15	Dem_SetEventStatusWithMonitorData	627
D.7.16	Dem_SetWIRStatus	628
D.7.17	<Module>_DemGetFaultDetectionCounter<ForEvent>	629
D.7.18	<Module>_DemInitMonitorFor<EventName>	629
D.7.19	<Module>_DemTriggerOnMonitorStatus	630
D.7.20	<Module>_DemClearEventAllowed<ForCondition>	630

D.7.21	<Module>_DemTriggerOnEventData	630
D.7.22	<Module>_DemTriggerOnEventUdsStatus	631
D.8	FiM	631
D.8.1	FiM_GetFunctionPermission	631
D.8.2	FiM_SetFunctionAvailable	632
E	Referenced Service Interfaces	633
E.1	Os	633
E.2	NvM	633
E.2.1	NvM_BlockIdType	633
E.2.2	NvM_BlockRequestType	633
E.2.3	NvM_InitBlockRequestType	634
E.2.4	NvM_RequestResultType	634
E.2.5	NvMService	635
E.2.6	NvMAdmin	639
E.2.7	NvMNotifyJobFinished	640
E.2.8	NvMNotifyInitBlock	640
E.2.9	NvMMirror	641
E.2.10	PS_{Block}	641
E.2.11	PAdmin_{Block}	642
E.2.12	PNJF_{Block}	642
E.2.13	PNIB_{Block}	643
E.2.14	PM_{Block}	643
E.3	Dcm	644
E.3.1	Implementation Data Types	644
E.3.1.1	Dcm_OpStatusType	644
E.3.1.2	Dcm_ConfirmationStatusType	644
E.3.1.3	Dcm_NegativeResponseCodeType	646
E.3.1.4	Dcm_DataElementType_{Data}Type	648
E.3.1.5	Dcm_DataElementType_{Data}ArrayType	649
E.3.1.6	Dcm_DataElementType_{Data}PrimitiveType	650
E.3.1.7	Dcm_DataArrayTypeUint8_{Data}Type	651
E.3.1.8	{DID}_Struct_DataType	651
E.3.1.9	Dcm_RangeArray_{Range}Type	652
E.3.1.10	Dcm_ScalingInfoArray_{Data}Type	652
E.3.1.11	Dcm_RequestDataOut_{Routine}_{Signal}PrimitiveType	653
E.3.1.12	Dcm_RequestDataIn_{Routine}_{Signal}PrimitiveType	653
E.3.1.13	Dcm_RequestDataOut_{Routine}_{Signal}Type	654
E.3.1.14	Dcm_RequestDataIn_{Routine}_{Signal}Type	655
E.3.1.15	Dcm_RequestDataOut_{Routine}_{Signal}ArrayType	655
E.3.1.16	Dcm_RequestDataIn_{Routine}_{Signal}ArrayType	656
E.3.1.17	Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type	657
E.3.1.18	Dcm_RequestFlexibleInArrayData_{Routine}_{Signal}Type	657
E.3.1.19	Dcm_StartDataIn_{Routine}_{Signal}PrimitiveType	657

E.3.1.20	Dcm_StartDataIn_{Routine}_{Signal}Type	658
E.3.1.21	Dcm_StartDataIn_{Routine}_{Signal}ArrayType	658
E.3.1.22	Dcm_StartDataOut_{Routine}_{Signal}PrimitivType	659
E.3.1.23	Dcm_StartDataOut_{Routine}_{Signal}Type	660
E.3.1.24	Dcm_StartDataOut_{Routine}_{Signal}ArrayType	660
E.3.1.25	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type	661
E.3.1.26	Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type	661
E.3.1.27	Dcm_StopDataIn_{Routine}_{Signal}PrimitivType	662
E.3.1.28	Dcm_StopDataIn_{Routine}_{Signal}Type	662
E.3.1.29	Dcm_StopDataIn_{Routine}_{Signal}ArrayType	663
E.3.1.30	Dcm_StopDataOut_{Routine}_{Signal}PrimitivType	664
E.3.1.31	Dcm_StopDataOut_{Routine}_{Signal}Type	664
E.3.1.32	Dcm_StopDataOut_{Routine}_{Signal}ArrayType	665
E.3.1.33	Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type	665
E.3.1.34	Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type	666
E.3.1.35	Dcm_KeyArray_{SecurityLevel}Type	666
E.3.1.36	Dcm_SeedArray_{SecurityLevel}Type	666
E.3.1.37	Dcm_SecurityAccessDataRecordArray_{SecurityLevel}Type	667
E.3.1.38	Dcm_RequestDataArrayType	667
E.3.1.39	Dcm_ControlMask_{DID}Type	667
E.3.1.40	Dcm_inputOutputControlParameterType	668
E.3.1.41	Dcm_IOOperationRequest_{DID}Type	668
E.3.1.42	Dcm_IOOperationResponseType	669
E.3.1.43	Dcm_ResponseDataArrayType	669
E.3.1.44	Dcm_ControlMask_{Data}ArrayType	670
E.3.1.45	Dcm_Cemr_{DID}Type	670
E.3.2	Sender-Receiver-Interfaces	670
E.3.2.1	DataServices_{DID}	671
E.3.2.2	DataServices_{Data}	671
E.3.2.3	IOControlRequest_{DID}	671
E.3.2.4	IOControlResponse_{DID}	672
E.3.3	NvDataInterface	672
E.3.3.1	DataServices_{DID}	672
E.3.4	Client-Server-Interfaces	673
E.3.4.1	DataServices_{Data}	673
E.3.4.2	RoutineServices_{RoutineName}	673
E.3.4.3	ServiceRequestNotification	687
E.3.5	Ports	689
E.3.5.1	DataServices_{DID} RPortProtoype	689
E.3.5.2	DataServices_{DID} PPortProtoype	689
E.3.5.3	DataServices_{DID} PRPortProtoype	689

	E.3.5.4	DataServices_{Data} RPortProtoype	690
	E.3.5.5	DataServices_{Data} RPortProtoype	690
	E.3.5.6	DataServices_{Data} PPortProtoype	690
	E.3.5.7	DataServices_{Data} PRPortProtoype	691
	E.3.5.8	IOControlRequest_{DID}	691
	E.3.5.9	IOControlResponse_{DID}	691
	E.3.5.10	ServiceRequestManufacturerNotification_{Name}	691
	E.3.5.11	ServiceRequestSupplierNotification_{Name}	692
	E.3.5.12	RoutineServices_{RoutineName}	692
E.4	Dem		692
	E.4.1	Implementation Data Types	692
		E.4.1.1 Dem_DebouncingStateType	692
		E.4.1.2 Dem_DTCTOriginType	693
		E.4.1.3 Dem_EventIdType	693
		E.4.1.4 Dem_EventStatusType	694
		E.4.1.5 Dem_DebounceResetStatusType	694
		E.4.1.6 Dem_DTCTFormatType	694
		E.4.1.7 Dem_InitMonitorReasonType	695
		E.4.1.8 Dem_MaxDataValueType	695
		E.4.1.9 Dem_MonitorDataType	696
		E.4.1.10 Dem_MonitorStatusType	696
		E.4.1.11 Dem_UdsStatusByteType	696
	E.4.2	Client-Server-Interfaces	697
		E.4.2.1 CallbackClearEventAllowed	697
		E.4.2.2 CallbackEventDataChanged	697
		E.4.2.3 CallbackEventUdsStatusChanged	698
		E.4.2.4 CallbackGetFaultDetectCounter	698
		E.4.2.5 CallbackInitMonitorForEvent	699
		E.4.2.6 CallbackMonitorStatusChange	699
		E.4.2.7 DiagnosticInfo	699
		E.4.2.8 DiagnosticMonitor	703
		E.4.2.9 DiagnosticMonitor_MonitorData	704
		E.4.2.10 EventStatus	705
	E.4.3	Ports	705
		E.4.3.1 Event	705
		E.4.3.2 DiagnosticMonitor_MonitorData	705
		E.4.3.3 EventStatus	706
		E.4.3.4 EventInfo	706
		E.4.3.5 CBClrEvt	706
		E.4.3.6 CBDataEvt	706
		E.4.3.7 CBFaultDetectCtr	707
		E.4.3.8 CBInitEvt	707
		E.4.3.9 CBEventUdsStatusChanged	707
		E.4.3.10 CBMonitorStatusChanged	707
E.5	FiM		708
	E.5.1	Implementation Data Types	708

E.5.1.1	FiM_FunctionIdType	708
E.5.2	Client-Server-Interfaces	708
E.5.2.1	FiM_FunctionInhibition	708
E.5.2.2	FiM_ControlFunctionAvailable	709
E.5.3	Ports	709
E.5.3.1	FiM_FunctionInhibition	709
E.5.3.2	FiM_ControlFunctionAvailable	709
F	Changes History	710
F.1	Changes in R20-11 compared to R19-11	710
F.1.1	Added Traceables in R20-11	710
F.1.2	Changed Traceables in R20-11	712
F.1.3	Deleted Traceables in R20-11	713
F.2	Changes in R21-11 compared to R20-11	713
F.2.1	Added Traceables in R21-11	713
F.2.2	Changed Traceables in R21-11	715
F.2.3	Deleted Traceables in R21-11	715

Known Limitations

- Not all VFB communication features are supported
- The Proxy Modules are not specified for all BSW Services
- No run-time separation is defined

Additional limitations are described in document [1]

1 Introduction and functional overview

This specification describes the functionality, API and the configuration for the AUTOSAR Basic Software module [Software Cluster Connection](#).

2 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to Requirements on Software Cluster Connection that are not included in the AUTOSAR Glossary [2].

Abbreviation / Acronym:	Description:
SwCluC	Software Cluster Connection

Table 2.1: Acronyms and Abbreviations

Terms:	Description:
Software Cluster	A Software Cluster groups all AUTOSAR artifacts that are relevant to deploy software on a machine. The full definition is given in document [2]
Software Cluster Connection	The Software Cluster Connection is the BSW module that provides the features to <ul style="list-style-type: none"> • connect the Binary Objects deployed on the same machine • substitute not locally-available BSW modules in an Applicative Software Cluster, whose interfaces are required for the integrated SW, by so called Proxy Modules. • implement the VFB communication features between Software Clusters together with RTE by means of an RTE Implementation Plug-In

Terms:	Description:
machine	A machine consists of a set of computing resources - such as microcontroller cores, memory or peripheral (e.g. communication) devices - and has the ability to execute software applications. The representation of a machine in the AUTOSAR Classic Platform could be done with an EcuInstance , but note that this semantic is currently in clarification. Further information is given in document [2].
binary-identical	Bit for Bit identical
Binary Object	A set of files, which contains the binary executable code and data. This binary executable code and data will not be modified again, before programming it on the target ECU.
Binary Manifest	The Binary Manifest is the well-defined interface of the Software Cluster's Binary Object , providing the meta information of a resources and information - so called handles - to access such a resource.
Applicative Software Cluster	A Software Cluster that mainly contains software components, and only selected BSW modules (e.g. a Service module, transformers, e.t.c.)
Host Software Cluster	The single Software Cluster that contains the major part of the BSW, and especially the micro controller dependent lower layer BSW Modules, e.g. OS and MCAL.
Substitution Software Cluster	The single Software Cluster that can override the provided resources of other Software Clusters for bug fixing purpose.
Proxy Module	A Proxy Module substitutes a BSW module in an Applicative Software Cluster . A Proxy module itself is split into High Proxy Module and Low Proxy Module . The High Proxy Module provides dedicated interfaces for modules in higher layers or same layer, and the functionality to connect them via the Binary Manifest to the Low Proxy Module in the Host Software Cluster .
High Proxy Module	The part of the Proxy Module residing in an Applicative Software Cluster .
Low Proxy Module	The part of the Proxy Module residing in the Host Software Cluster .
Os High Proxy	A type of Proxy Module implementing Os APIs in the Applicative Software Cluster .
Os Low Proxy	A type of proxy Module implementing an Os abstraction in the Host Software Cluster .
NvM High Proxy	A type of Proxy Module substituting the NVRAM Manager in the Applicative Software Cluster .
NvM Low Proxy	A type of Proxy Module connecting the NvM High Proxy Modules to the NVRAM Manager in the Host Software Cluster .
LdCom High Proxy	A type of Proxy Module implementing default LdCom (Callback) APIs in the Applicative Software Cluster .
LdCom Low Proxy	A type of proxy Module connecting LdCom High Proxy Modules to the LdCom in the Host Software Cluster .
Com High Proxy	A type of Proxy Module implementing default Com (Callback) APIs in the Applicative Software Cluster .
Com Low Proxy	A type of proxy Module connecting Com High Proxy Modules to the Com in the Host Software Cluster .
Dcm High Proxy	A type of Proxy Module substituting the Dcm in the Applicative Software Cluster .

Terms:	Description:
Dcm Low Proxy	A type of proxy Module connecting Dcm High Proxy Modules to the Dcm in the Host Software Cluster .
Dem High Proxy	A type of Proxy Module substituting the Dem in the Applicative Software Cluster .
Dem Low Proxy	A type of proxy Module connecting Dem High Proxy Modules to the Dem in the Host Software Cluster .
FiM High Proxy	A type of Proxy Module substituting the FiM in the Applicative Software Cluster .
FiM Low Proxy	A type of proxy Module connecting FiM High Proxy Modules to the FiM in the Host Software Cluster .
RTE Implementation Plug-In	A RTE Implementation Plug-In is a part of the overall RTE implementation, which is not provided by the RTE Generator, but from an additional source (e.g. a Plug-In Generator or a manually implemented source code).
RTE Implementation Plug-In Service	A RTE Implementation Plug-In Service is a single entry point into the RTE Implementation Plug-In implementing a low level service for the RTE. For instance access to a specific buffer.
Local Software Cluster Communication Plug-In	A Local Software Cluster Communication Plug-In is an RTE Implementation Plug-In , which handles the communication locally inside a Software Cluster . This includes the Transformer handling, if a DataMapping exist for the according Communication Graph
Cross Software Cluster Communication Plug-In	A Cross Software Cluster Communication Plug-In is an RTE Implementation Plug-In that handles the communication towards other Software Clusters . This includes the Transformer handling, if intra ECU transformation is configured.
Communication Graph	The sum of all AbstractAccessPoints to elements of Port-Interfaces , instantiated in PortPrototypes which are connected to each other; or the sum of all accesses from BswModuleEntitys to interface elements in a BswModuleDescriptions connected to each other.
Data Communication Graph	The sum of all VariableAccesses to VariableDataPrototypes instantiated in PortPrototypes , which are connected to each other; or the sum of all VariableAccesses to VariableDataPrototypes in the InternalBehavior ; or the sum of all BswVariableAccesses to VariableDataPrototypes in BswModuleDescriptions connected to each other.
Parameter Communication Graph	The sum of all ParameterAccesses to ParameterDataPrototypes instantiated in PortPrototypes , which are connected to each other; or the sum of all ParameterAccesses to ParameterDataPrototypes in the InternalBehavior .
Client Server Communication Graph	The sum of all ServerCallPoints to operations instantiated in PortPrototypes , which are connected to each other, including the associated server runnable .
Trigger Communication Graph	The sum of all ExternalTriggeringPoints for triggers instantiated in PortPrototypes , which are connected to each other, including the associated triggered runnable .
Mode Communication Graph	The sum of all ModeAccessPoints and ModeSwitchPoints to ModeDeclarationGroupPrototypes instantiated in Port-Prototypes , which are connected to each other; or the sum of all managedModeGroups and accessedModeGroups to ModeDeclarationGroupPrototypes in BswModuleDescriptions connected to each other.

Terms:	Description:
mode manager	Entering and leaving modes is initiated by a <i>mode manager</i> . A <i>mode manager</i> is either a software component that provides a p-port typed by a <code>ModeSwitchInterface</code> , or a BSW module that defines in its <code>BswModuleDescription</code> a <code>ModeDeclarationGroupPrototype</code> in the role <code>providedModeGroup</code> .
mode switch notification	The communication of a mode switch from the <code>mode manager</code> to the <code>mode user</code> , using either the <code>ModeSwitchInterface</code> or <code>providedModeGroup</code> and <code>requiredModeGroup ModeDeclarationGroupPrototypes</code> .
mode switch port	The port for receiving (or sending) a mode switch notification. For this purpose, a <code>mode switch port</code> is typed by a <code>ModeSwitchInterface</code> .
mode user	An <i>AUTOSAR SW-C</i> or <i>AUTOSAR Basic Software Module</i> that depends on modes, is called a mode user. The dependency can occur through a <code>SwcModeSwitchEvent/BswModeSwitchEvent</code> , a <code>ModeAccessPoint</code> for a provided/required <code>mode switch port</code> , or a <code>accessedModeGroup</code> for a <code>providedModeGroup/requiredModeGroup ModeDeclarationGroupPrototype</code> .
on-entry ExecutableEntity	A <code>RunnableEntity</code> that is triggered by a <code>SwcModeSwitchEvent</code> with <code>ModeActivationKind</code> 'entry'; or a <code>BswSchedulableEntity</code> that is triggered by a <code>BswModeSwitchEvent</code> with <code>ModeActivationKind</code> 'entry'.
on-exit ExecutableEntity	A <code>RunnableEntity</code> that is triggered by a <code>SwcModeSwitchEvent</code> with <code>ModeActivationKind</code> 'exit'; or a <code>BswSchedulableEntity</code> that is triggered by a <code>BswModeSwitchEvent</code> with <code>ModeActivationKind</code> 'exit'.
on-transition ExecutableEntity	A <code>RunnableEntity</code> that is triggered by a <code>SwcModeSwitchEvent</code> with <code>ModeActivationKind</code> 'transition'; or a <code>BswSchedulableEntity</code> that is triggered by a <code>BswModeSwitchEvent</code> with <code>ModeActivationKind</code> 'transition'.
trigger port	A <code>PortPrototype</code> , which is typed by an <code>TriggerInterface</code>
trigger sink	A <i>trigger sink</i> relies on the activation of <code>RunnableEntity</code> or a <code>BswSchedulableEntity</code> , if a particular <code>Trigger</code> is raised. A <i>trigger sink</i> has a dedicated require <code>trigger port(s)</code> and / or <code>requiredTrigger Trigger(s)</code> to communicate to the <code>trigger source(s)</code> .
trigger source	A <i>trigger source</i> administrates the particular <code>Trigger</code> , and informs the RTE or <i>Basic Software Scheduler</i> if the <code>Trigger</code> is raised. A <i>trigger source</i> has dedicated provide <code>trigger port(s)</code> and / or <code>releasedTrigger Trigger(s)</code> to communicate to the <code>trigger sink(s)</code> .
triggered BswSchedulableEntity	A <code>BswSchedulableEntity</code> that is triggered at least by one <code>BswExternalTriggerOccurredEvent</code> or <code>BswInternalTriggerOccurredEvent</code> . In particular cases, the <i>Trigger Event Communication</i> or the <i>Inter Basic Software Schedulable Entity Triggering</i> is implemented by the <i>Basic Software Scheduler</i> as a direct or trusted function call of the <code>triggered ExecutableEntity</code> , by the triggering <code>ExecutableEntity</code> .

Terms:	Description:
triggered ExecutableEntity	A RunnableEntity that is triggered by at least one ExternalTriggerOccurredEvent / InternalTriggerOccurredEvent ; or a BswSchedulableEntity that is triggered by at least one BswExternalTriggerOccurredEvent / BswInternalTriggerOccurredEvent . In particular cases, the Trigger Event Communication or the <i>Inter Runnable Triggering</i> is implemented by RTE or <i>Basic Software Scheduler</i> as a direct or trusted function call of the triggered ExecutableEntity , by the triggering ExecutableEntity .
triggered runnable	A RunnableEntity that is triggered at least by one ExternalTriggerOccurredEvent or InternalTriggerOccurredEvent . In particular cases, the Trigger Event Communication or the <i>Inter Runnable Triggering</i> is implemented by RTE as a direct or trusted function call of the <i>triggered runnable</i> , by the triggering runnable.

Table 2.2: Terms

3 Related documentation

3.1 Input documents & related standards and norms

- [1] Explanation of CP Software Cluster Design And Integration Guideline
AUTOSAR_EXP_CPSwClusterDesignAndIntegrationGuideline
- [2] Glossary
AUTOSAR_TR_Glossary
- [3] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral
- [4] Requirements on Software Cluster Connection module
AUTOSAR_SRS_SoftwareClusterConnection
- [5] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral
- [6] System Template
AUTOSAR_TPS_SystemTemplate
- [7] Specification of CRC Routines
AUTOSAR_SWS_CRCLibrary
- [8] Specification of Memory Mapping
AUTOSAR_SWS_MemoryMapping
- [9] Specification of RTE Software
AUTOSAR_SWS_RTE

- [10] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration
- [11] Software Component Template
AUTOSAR_TPS_SoftwareComponentTemplate
- [12] Specification of Operating System
AUTOSAR_SWS_OS
- [13] Specification of NVRAM Manager
AUTOSAR_SWS_NVRAMManager
- [14] Specification of Watchdog Manager
AUTOSAR_SWS_WatchdogManager
- [15] Specification of Platform Types
AUTOSAR_SWS_PlatformTypes
- [16] Specification of Standard Types
AUTOSAR_SWS_StandardTypes
- [17] ISO 17356-3: Road vehicles – Open interface for embedded automotive applications – Part 3: OSEK/VDX Operating System (OS)

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [3, SWS BSW General], which is also valid for [Software Cluster Connection](#).

Thus, the specification SWS BSW General shall be considered as additional and required specification for [Software Cluster Connection](#).

4 Constraints and assumptions

4.1 Limitations

The specification currently supports a limited number of BSW modules. Furthermore, for the supported BSW modules, the functionality and APIs available for Applicative Software Clusters is only subset of the overall functionality. In addition, the available VFB communication features are restricted.

4.2 Applicability to car domains

The specification focus on larger ECUs centralizing software functionality - so called domain or zone controllers. It assumes that the software components which are

mapped to different [Software Clusters](#) are rather loosely coupled w.r.t. to interfaces and time domain. The software components supposed to implement mainly control loop software - usually time driven but may also react on limited numbers of sporadic events. Further information can be found in document [1].

5 Dependencies to other modules

5.1 Binary Manifest

5.1.1 Header File structure

[SWS_SwCluC_00013]{DRAFT} [The [Binary Manifest](#) of the [Software Cluster Connection](#) shall provide the header file `SwCluC_BManif.h`.

]()

[SWS_SwCluC_00014]{DRAFT} [The header file `SwCluC_BManif.h` shall include all header files configured in [SwCluCBManifHeaderIncludes](#) in the containers [SwCluCBManifProvideResourceEntryGroup](#) and [SwCluCBManifRequireResourceEntryGroup](#).]()

6 Requirements Tracing

The following tables reference the requirements specified in [4], `SRS_SoftwareClusterConnection` and [5], `SRS_BSWGeneral` and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[SRS_BSW_00005]	Modules of the μ C Abstraction Layer (MCAL) may not have hard coded horizontal interfaces	[SWS_SwCluC_00999]
[SRS_BSW_00006]	The source code of software modules above the μ C Abstraction Layer (MCAL) shall not be processor and compiler dependent.	[SWS_SwCluC_00999]
[SRS_BSW_00007]	All Basic SW Modules written in C language shall conform to the MISRA C 2012 Standard.	[SWS_SwCluC_00999]





Requirement	Description	Satisfied by
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_SwCluC_11000] [SWS_SwCluC_11001] [SWS_SwCluC_12000] [SWS_SwCluC_12100] [SWS_SwCluC_12200] [SWS_SwCluC_12300] [SWS_SwCluC_12400] [SWS_SwCluC_13000] [SWS_SwCluC_13001]
[SRS_BSW_00160]	Configuration files of AUTOSAR Basic SW module shall be readable for human beings	[SWS_SwCluC_00999]
[SRS_BSW_00161]	The AUTOSAR Basic Software shall provide a microcontroller abstraction layer which provides a standardized interface to higher software layers	[SWS_SwCluC_00999]
[SRS_BSW_00162]	The AUTOSAR Basic Software shall provide a hardware abstraction layer	[SWS_SwCluC_00999]
[SRS_BSW_00164]	The Implementation of interrupt service routines shall be done by the Operating System, complex drivers or modules	[SWS_SwCluC_00999]
[SRS_BSW_00168]	SW components shall be tested by a function defined in a common API in the Basis-SW	[SWS_SwCluC_00999]
[SRS_BSW_00170]	The AUTOSAR SW Components shall provide information about their dependency from faults, signal qualities, driver demands	[SWS_SwCluC_00999]
[SRS_BSW_00301]	All AUTOSAR Basic Software Modules shall only import the necessary information	[SWS_SwCluC_00999]
[SRS_BSW_00302]	All AUTOSAR Basic Software Modules shall only export information needed by other modules	[SWS_SwCluC_00999]
[SRS_BSW_00304]	All AUTOSAR Basic Software Modules shall use only AUTOSAR data types instead of native C data types	[SWS_SwCluC_00999]
[SRS_BSW_00306]	AUTOSAR Basic Software Modules shall be compiler and platform independent	[SWS_SwCluC_00999]
[SRS_BSW_00307]	Global variables naming convention	[SWS_SwCluC_00999]





Requirement	Description	Satisfied by
[SRS_BSW_00308]	AUTOSAR Basic Software Modules shall not define global data in their header files, but in the C file	[SWS_SwCluC_00999]
[SRS_BSW_00309]	All AUTOSAR Basic Software Modules shall indicate all global data with read-only purposes by explicitly assigning the const keyword	[SWS_SwCluC_00999]
[SRS_BSW_00310]	API naming convention	[SWS_SwCluC_10000] [SWS_SwCluC_10001] [SWS_SwCluC_10002] [SWS_SwCluC_10003] [SWS_SwCluC_10004] [SWS_SwCluC_10005] [SWS_SwCluC_10006] [SWS_SwCluC_10007] [SWS_SwCluC_10008] [SWS_SwCluC_10011] [SWS_SwCluC_10020] [SWS_SwCluC_10021] [SWS_SwCluC_10022] [SWS_SwCluC_10023] [SWS_SwCluC_10032] [SWS_SwCluC_10033] [SWS_SwCluC_10034] [SWS_SwCluC_11000] [SWS_SwCluC_11001] [SWS_SwCluC_12000] [SWS_SwCluC_12100] [SWS_SwCluC_12200] [SWS_SwCluC_12300] [SWS_SwCluC_12400] [SWS_SwCluC_13000] [SWS_SwCluC_13001]
[SRS_BSW_00312]	Shared code shall be reentrant	[SWS_SwCluC_00999]
[SRS_BSW_00314]	All internal driver modules shall separate the interrupt frame definition from the service routine	[SWS_SwCluC_00999]
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_SwCluC_02137] [SWS_SwCluC_02138] [SWS_SwCluC_02139] [SWS_SwCluC_02533] [SWS_SwCluC_02534] [SWS_SwCluC_02636] [SWS_SwCluC_02637] [SWS_SwCluC_02638] [SWS_SwCluC_02751] [SWS_SwCluC_03217] [SWS_SwCluC_03218]
[SRS_BSW_00325]	The runtime of interrupt service routines and functions that are running in interrupt context shall be kept short	[SWS_SwCluC_00999]
[SRS_BSW_00327]	Error values naming convention	[SWS_SwCluC_02140] [SWS_SwCluC_02426] [SWS_SwCluC_02545] [SWS_SwCluC_02650] [SWS_SwCluC_02750] [SWS_SwCluC_03216]
[SRS_BSW_00328]	All AUTOSAR Basic Software Modules shall avoid the duplication of code	[SWS_SwCluC_00999]
[SRS_BSW_00330]	It shall be allowed to use macros instead of functions where source code is used and runtime is critical	[SWS_SwCluC_00999]
[SRS_BSW_00331]	All Basic Software Modules shall strictly separate error and status information	[SWS_SwCluC_00999]
[SRS_BSW_00335]	Status values naming convention	[SWS_SwCluC_00999]
[SRS_BSW_00337]	Classification of development errors	[SWS_SwCluC_02140] [SWS_SwCluC_02426] [SWS_SwCluC_02545] [SWS_SwCluC_02650] [SWS_SwCluC_02750] [SWS_SwCluC_03216]
[SRS_BSW_00339]	Reporting of production relevant error status	[SWS_SwCluC_00999]





Requirement	Description	Satisfied by
[SRS_BSW_00342]	It shall be possible to create an AUTOSAR ECU out of modules provided as source code and modules provided as object code, even mixed	[SWS_SwCluC_00999]
[SRS_BSW_00343]	The unit of time for specification and configuration of Basic SW modules shall be preferably in physical time unit	[SWS_SwCluC_00999]
[SRS_BSW_00344]	BSW Modules shall support link-time configuration	[SWS_SwCluC_00999]
[SRS_BSW_00346]	All AUTOSAR Basic Software Modules shall provide at least a basic set of module files	[SWS_SwCluC_00999]
[SRS_BSW_00347]	A Naming separation of different instances of BSW drivers shall be in place	[SWS_SwCluC_00999]
[SRS_BSW_00348]	All AUTOSAR standard types and constants shall be placed and organized in a standard type header file	[SWS_SwCluC_00999]
[SRS_BSW_00353]	All integer type definitions of target and compiler specific scope shall be placed and organized in a single type header	[SWS_SwCluC_00999]
[SRS_BSW_00357]	For success/failure of an API call a standard return type shall be defined	[SWS_SwCluC_00999]
[SRS_BSW_00358]	The return type of init() functions implemented by AUTOSAR Basic Software Modules shall be void	[SWS_SwCluC_11000] [SWS_SwCluC_11001] [SWS_SwCluC_12000] [SWS_SwCluC_12100] [SWS_SwCluC_12200] [SWS_SwCluC_12300] [SWS_SwCluC_12400] [SWS_SwCluC_13000] [SWS_SwCluC_13001]
[SRS_BSW_00359]	All AUTOSAR Basic Software Modules callback functions shall avoid return types other than void if possible	[SWS_SwCluC_00999]
[SRS_BSW_00360]	AUTOSAR Basic Software Modules callback functions are allowed to have parameters	[SWS_SwCluC_00999]





Requirement	Description	Satisfied by
[SRS_BSW_00361]	All mappings of not standardized keywords of compiler specific scope shall be placed and organized in a compiler specific type and keyword header	[SWS_SwCluC_00999]
[SRS_BSW_00369]	All AUTOSAR Basic Software Modules shall not return specific development error codes via the API	[SWS_SwCluC_02137] [SWS_SwCluC_02138] [SWS_SwCluC_02139] [SWS_SwCluC_02533] [SWS_SwCluC_02534] [SWS_SwCluC_02636] [SWS_SwCluC_02637] [SWS_SwCluC_02638] [SWS_SwCluC_02751] [SWS_SwCluC_03217] [SWS_SwCluC_03218]
[SRS_BSW_00373]	The main processing function of each AUTOSAR Basic Software Module shall be named according the defined convention	[SWS_SwCluC_00999]
[SRS_BSW_00375]	Basic Software Modules shall report wake-up reasons	[SWS_SwCluC_00999]
[SRS_BSW_00377]	A Basic Software Module can return a module specific types	[SWS_SwCluC_00999]
[SRS_BSW_00378]	AUTOSAR shall provide a boolean type	[SWS_SwCluC_00999]
[SRS_BSW_00380]	Configuration parameters being stored in memory shall be placed into separate c-files	[SWS_SwCluC_00999]
[SRS_BSW_00385]	List possible error notifications	[SWS_SwCluC_02140] [SWS_SwCluC_02426] [SWS_SwCluC_02545] [SWS_SwCluC_02650] [SWS_SwCluC_02750] [SWS_SwCluC_03216]
[SRS_BSW_00404]	BSW Modules shall support post-build configuration	[SWS_SwCluC_00999]
[SRS_BSW_00405]	BSW Modules shall support multiple configuration sets	[SWS_SwCluC_00999]
[SRS_BSW_00406]	A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called	[SWS_SwCluC_00999]
[SRS_BSW_00409]	All production code error ID symbols are defined by the Dem module and shall be retrieved by the other BSW modules from Dem configuration	[SWS_SwCluC_00999]
[SRS_BSW_00410]	Compiler switches shall have defined values	[SWS_SwCluC_00999]





Requirement	Description	Satisfied by
[SRS_BSW_00411]	All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API	[SWS_SwCluC_00999]
[SRS_BSW_00413]	An index-based accessing of the instances of BSW modules shall be done	[SWS_SwCluC_00999]
[SRS_BSW_00414]	Init functions shall have a pointer to a configuration structure as single parameter	[SWS_SwCluC_00999]
[SRS_BSW_00415]	Interfaces which are provided exclusively for one module shall be separated into a dedicated header file	[SWS_SwCluC_00999]
[SRS_BSW_00416]	The sequence of modules to be initialized shall be configurable	[SWS_SwCluC_00999]
[SRS_BSW_00417]	Software which is not part of the SW-C shall report error events only after the DEM is fully operational.	[SWS_SwCluC_00999]
[SRS_BSW_00419]	If a pre-compile time configuration parameter is implemented as "const" it should be placed into a separate c-file	[SWS_SwCluC_00999]
[SRS_BSW_00422]	Pre-de-bouncing of error status information is done within the DEM	[SWS_SwCluC_00999]
[SRS_BSW_00423]	BSW modules with AUTOSAR interfaces shall be describable with the means of the SW-C Template	[SWS_SwCluC_00999]
[SRS_BSW_00424]	BSW module main processing functions shall not be allowed to enter a wait state	[SWS_SwCluC_00999]
[SRS_BSW_00425]	The BSW module description template shall provide means to model the defined trigger conditions of schedulable objects	[SWS_SwCluC_00999]
[SRS_BSW_00426]	BSW Modules shall ensure data consistency of data which is shared between BSW modules	[SWS_SwCluC_00999]





Requirement	Description	Satisfied by
[SRS_BSW_00427]	ISR functions shall be defined and documented in the BSW module description template	[SWS_SwCluC_00999]
[SRS_BSW_00432]	Modules should have separate main processing functions for read/receive and write/transmit data path	[SWS_SwCluC_00999]
[SRS_BSW_00433]	Main processing functions are only allowed to be called from task bodies provided by the BSW Scheduler	[SWS_SwCluC_00999]
[SRS_BSW_00437]	Memory mapping shall provide the possibility to define RAM segments which are not to be initialized during startup	[SWS_SwCluC_00999]
[SRS_BSW_00438]	Configuration data shall be defined in a structure	[SWS_SwCluC_00999]
[SRS_BSW_00439]	Enable BSW modules to handle interrupts	[SWS_SwCluC_00999]
[SRS_BSW_00440]	The callback function invocation by the BSW module shall follow the signature provided by RTE to invoke servers via Rte_Call API	[SWS_SwCluC_00999]
[SRS_BSW_00441]	Naming convention for type, macro and function	[SWS_SwCluC_01000] [SWS_SwCluC_01002]
[SRS_BSW_00447]	Standardizing Include file structure of BSW Modules Implementing Autosar Service	[SWS_SwCluC_00999]
[SRS_BSW_00448]	Module SWS shall not contain requirements from Other Modules	[SWS_SwCluC_00999]
[SRS_BSW_00449]	BSW Service APIs used by Autosar Application Software shall return a Std_ReturnType	[SWS_SwCluC_00999]
[SRS_BSW_00450]	A Main function of a un-initialized module shall return immediately	[SWS_SwCluC_00999]
[SRS_BSW_00451]	Hardware registers shall be protected if concurrent access to these registers occur	[SWS_SwCluC_00999]
[SRS_BSW_00452]	Classification of runtime errors	[SWS_SwCluC_00999]
[SRS_BSW_00453]	BSW Modules shall be harmonized	[SWS_SwCluC_00999]





Requirement	Description	Satisfied by
[SRS_BSW_00454]	An alternative interface without a parameter of category DATA_REFERENCE shall be available.	[SWS_SwCluC_00999]
[SRS_BSW_00456]	A Header file shall be defined in order to harmonize BSW Modules	[SWS_SwCluC_00999]
[SRS_BSW_00457]	Callback functions of Application software components shall be invoked by the Basis SW	[SWS_SwCluC_00999]
[SRS_BSW_00458]	Classification of production errors	[SWS_SwCluC_00999]
[SRS_BSW_00459]	It shall be possible to concurrently execute a service offered by a BSW module in different partitions	[SWS_SwCluC_00999]
[SRS_BSW_00460]	Reentrancy Levels	[SWS_SwCluC_00999]
[SRS_BSW_00461]	Modules called by generic modules shall satisfy all interfaces requested by the generic module	[SWS_SwCluC_00999]
[SRS_BSW_00463]	Naming convention of callout prototypes	[SWS_SwCluC_00999]
[SRS_BSW_00466]	Classification of extended production errors	[SWS_SwCluC_00999]
[SRS_BSW_00467]	The init / deinit services shall only be called by BswM or EcuM	[SWS_SwCluC_00999]
[SRS_BSW_00469]	Fault detection and healing of production errors and extended production errors	[SWS_SwCluC_00999]
[SRS_BSW_00470]	Execution frequency of production error detection	[SWS_SwCluC_00999]
[SRS_BSW_00471]	Do not cause dead-locks on detection of production errors - the ability to heal from previously detected production errors	[SWS_SwCluC_00999]
[SRS_BSW_00472]	Avoid detection of two production errors with the same root cause.	[SWS_SwCluC_00999]
[SRS_BSW_00473]	Classification of transient faults	[SWS_SwCluC_00999]
[SRS_BSW_00477]	The functional interfaces of AUTOSAR BSW modules shall be specified in C99	[SWS_SwCluC_00999]





Requirement	Description	Satisfied by
[SRS_BSW_00478]	Timing limits of main functions	[SWS_SwCluC_00999]
[SRS_BSW_00479]	Interfaces for handling request from external devices	[SWS_SwCluC_00999]
[SRS_BSW_00480]	NullPointer Errors shall follow a naming rule	[SWS_SwCluC_02140] [SWS_SwCluC_02426] [SWS_SwCluC_02545] [SWS_SwCluC_02650] [SWS_SwCluC_02750] [SWS_SwCluC_03216]
[SRS_BSW_00481]	Invalid configuration set selection errors shall follow a naming rule	[SWS_SwCluC_00999]
[SRS_BSW_00483]	BSW Modules shall handle buffer alignments internally	[SWS_SwCluC_00999]
[SRS_BSW_00484]	Input parameters of scalar and enum types shall be passed as a value.	[SWS_SwCluC_00999]
[SRS_BSW_00485]	Input parameters of structure type shall be passed as a reference to a constant structure	[SWS_SwCluC_00999]
[SRS_BSW_00486]	Input parameters of array type shall be passed as a reference to the constant array base type	[SWS_SwCluC_00999]
[SRS_BSW_00487]	Errors for module initialization shall follow a naming rule	[SWS_SwCluC_02140] [SWS_SwCluC_02426] [SWS_SwCluC_02545] [SWS_SwCluC_02650] [SWS_SwCluC_02750] [SWS_SwCluC_03216]
[SRS_BSW_00488]	Classification of security events	[SWS_SwCluC_00999]
[SRS_SwCluC_00001]	Easy target machine interpretation	[SWS_SwCluC_00001] [SWS_SwCluC_00003] [SWS_SwCluC_00004] [SWS_SwCluC_00005] [SWS_SwCluC_00006] [SWS_SwCluC_00007] [SWS_SwCluC_00008] [SWS_SwCluC_00009] [SWS_SwCluC_00010] [SWS_SwCluC_00011] [SWS_SwCluC_00012] [SWS_SwCluC_00015] [SWS_SwCluC_00016] [SWS_SwCluC_00017] [SWS_SwCluC_00018] [SWS_SwCluC_00019] [SWS_SwCluC_00020] [SWS_SwCluC_00021] [SWS_SwCluC_00022] [SWS_SwCluC_00023] [SWS_SwCluC_00024] [SWS_SwCluC_00025] [SWS_SwCluC_00026] [SWS_SwCluC_00027] [SWS_SwCluC_00028] [SWS_SwCluC_00029] [SWS_SwCluC_00030] [SWS_SwCluC_00031] [SWS_SwCluC_00032] [SWS_SwCluC_00033] [SWS_SwCluC_00034] [SWS_SwCluC_00035] [SWS_SwCluC_00036] [SWS_SwCluC_00037] [SWS_SwCluC_00040] [SWS_SwCluC_00041] [SWS_SwCluC_00042] [SWS_SwCluC_00070] [SWS_SwCluC_00072] [SWS_SwCluC_00079] [SWS_SwCluC_00080] [SWS_SwCluC_00081] [SWS_SwCluC_00082] [SWS_SwCluC_00083] [SWS_SwCluC_00085] [SWS_SwCluC_00090] [SWS_SwCluC_CONSTR_00073]
[SRS_SwCluC_00002]	Identification of intended connections by unique IDs	[SWS_SwCluC_00015] [SWS_SwCluC_00021] [SWS_SwCluC_00034] [SWS_SwCluC_00035] [SWS_SwCluC_00043]
[SRS_SwCluC_00003]	Bidirectional connections	[SWS_SwCluC_00024] [SWS_SwCluC_00026]
[SRS_SwCluC_00004]	Connection multiplicity	[SWS_SwCluC_00027] [SWS_SwCluC_00045] [SWS_SwCluC_00046] [SWS_SwCluC_00047] [SWS_SwCluC_00089]
[SRS_SwCluC_00005]	Substitute Resource Providers	[SWS_SwCluC_00008] [SWS_SwCluC_00054] [SWS_SwCluC_00055] [SWS_SwCluC_CONSTR_00087]





Requirement	Description	Satisfied by
[SRS_SwCluC_00006]	C API	[SWS_SwCluC_00002] [SWS_SwCluC_00062] [SWS_SwCluC_00063] [SWS_SwCluC_00064] [SWS_SwCluC_00065] [SWS_SwCluC_00066] [SWS_SwCluC_00067] [SWS_SwCluC_00068] [SWS_SwCluC_00069] [SWS_SwCluC_01000] [SWS_SwCluC_01002] [SWS_SwCluC_10000] [SWS_SwCluC_10001] [SWS_SwCluC_10002] [SWS_SwCluC_10003] [SWS_SwCluC_10004] [SWS_SwCluC_10005] [SWS_SwCluC_10006] [SWS_SwCluC_10007] [SWS_SwCluC_10008] [SWS_SwCluC_10009] [SWS_SwCluC_10010] [SWS_SwCluC_10011] [SWS_SwCluC_10020] [SWS_SwCluC_10021] [SWS_SwCluC_10022] [SWS_SwCluC_10023] [SWS_SwCluC_10032] [SWS_SwCluC_10033] [SWS_SwCluC_10034]
[SRS_SwCluC_00009]	Support missing interface partners	[SWS_SwCluC_00015] [SWS_SwCluC_00019] [SWS_SwCluC_00030] [SWS_SwCluC_00034] [SWS_SwCluC_00035] [SWS_SwCluC_00044]
[SRS_SwCluC_00010]	Static safeguard of Software Cluster connections	[SWS_SwCluC_00015] [SWS_SwCluC_00022] [SWS_SwCluC_03005] [SWS_SwCluC_03033] [SWS_SwCluC_03034] [SWS_SwCluC_03035] [SWS_SwCluC_03036] [SWS_SwCluC_03037] [SWS_SwCluC_03038] [SWS_SwCluC_03039] [SWS_SwCluC_03040] [SWS_SwCluC_03041] [SWS_SwCluC_03042] [SWS_SwCluC_03043] [SWS_SwCluC_03044] [SWS_SwCluC_03045] [SWS_SwCluC_03046] [SWS_SwCluC_03047]
[SRS_SwCluC_00011]	Separation of immutable memory and memory modifiable at the connection phase	[SWS_SwCluC_00001] [SWS_SwCluC_00003] [SWS_SwCluC_00004] [SWS_SwCluC_00005] [SWS_SwCluC_00006] [SWS_SwCluC_00007] [SWS_SwCluC_00008] [SWS_SwCluC_00009] [SWS_SwCluC_00010] [SWS_SwCluC_00011] [SWS_SwCluC_00012] [SWS_SwCluC_00015] [SWS_SwCluC_00028] [SWS_SwCluC_00029] [SWS_SwCluC_00030] [SWS_SwCluC_00031] [SWS_SwCluC_00032] [SWS_SwCluC_00033] [SWS_SwCluC_00034] [SWS_SwCluC_00035] [SWS_SwCluC_00042] [SWS_SwCluC_00070] [SWS_SwCluC_00072] [SWS_SwCluC_00079] [SWS_SwCluC_00080] [SWS_SwCluC_00081] [SWS_SwCluC_00082] [SWS_SwCluC_00083] [SWS_SwCluC_00085]
[SRS_SwCluC_00012]	direct linkage	[SWS_SwCluC_00057] [SWS_SwCluC_00058] [SWS_SwCluC_00059] [SWS_SwCluC_00060] [SWS_SwCluC_00061]
[SRS_SwCluC_00013]	Initialization with C-compiler and linker means	[SWS_SwCluC_00001] [SWS_SwCluC_00003] [SWS_SwCluC_00004] [SWS_SwCluC_00005] [SWS_SwCluC_00006] [SWS_SwCluC_00007] [SWS_SwCluC_00008] [SWS_SwCluC_00009] [SWS_SwCluC_00010] [SWS_SwCluC_00011] [SWS_SwCluC_00012] [SWS_SwCluC_00029] [SWS_SwCluC_00030] [SWS_SwCluC_00031] [SWS_SwCluC_00032] [SWS_SwCluC_00033] [SWS_SwCluC_00034] [SWS_SwCluC_00035] [SWS_SwCluC_00042] [SWS_SwCluC_00070] [SWS_SwCluC_00072]
[SRS_SwCluC_00014]	Standardized persistence in memory	[SWS_SwCluC_00001] [SWS_SwCluC_00003] [SWS_SwCluC_00004] [SWS_SwCluC_00005] [SWS_SwCluC_00006] [SWS_SwCluC_00007] [SWS_SwCluC_00008] [SWS_SwCluC_00009] [SWS_SwCluC_00010] [SWS_SwCluC_00011] [SWS_SwCluC_00012] [SWS_SwCluC_00015] [SWS_SwCluC_00016] [SWS_SwCluC_00017] [SWS_SwCluC_00018] [SWS_SwCluC_00019] [SWS_SwCluC_00020] [SWS_SwCluC_00021] [SWS_SwCluC_00022] [SWS_SwCluC_00023] [SWS_SwCluC_00024] [SWS_SwCluC_00025] [SWS_SwCluC_00026] [SWS_SwCluC_00027] [SWS_SwCluC_00028] [SWS_SwCluC_00029] [SWS_SwCluC_00030] [SWS_SwCluC_00031] [SWS_SwCluC_00032] [SWS_SwCluC_00033] [SWS_SwCluC_00034] [SWS_SwCluC_00035] [SWS_SwCluC_00036] [SWS_SwCluC_00037] [SWS_SwCluC_00040] [SWS_SwCluC_00041] [SWS_SwCluC_00042] [SWS_SwCluC_00070] [SWS_SwCluC_00072] [SWS_SwCluC_00079] [SWS_SwCluC_00080] [SWS_SwCluC_00081] [SWS_SwCluC_00082] [SWS_SwCluC_00083] [SWS_SwCluC_00085] [SWS_SwCluC_00090] [SWS_SwCluC_CONSTR_00073] [SWS_SwCluC_CONSTR_00091] [SWS_SwCluC_CONSTR_00092] [SWS_SwCluC_CONSTR_00093] [SWS_SwCluC_CONSTR_00094]
[SRS_SwCluC_00100]	Cross Software Cluster Communication Plug-Ins	[SWS_SwCluC_03000] [SWS_SwCluC_03001] [SWS_SwCluC_03002] [SWS_SwCluC_03003] [SWS_SwCluC_03004] [SWS_SwCluC_03006] [SWS_SwCluC_03007] [SWS_SwCluC_03008] [SWS_SwCluC_03009] [SWS_SwCluC_03010] [SWS_SwCluC_03011] [SWS_SwCluC_03064] [SWS_SwCluC_03065] [SWS_SwCluC_03068] [SWS_SwCluC_03080]





Requirement	Description	Satisfied by
		<p style="text-align: center;">△</p> <p>[SWS_SwCluC_03081] [SWS_SwCluC_03082] [SWS_SwCluC_03101] [SWS_SwCluC_03102] [SWS_SwCluC_03103] [SWS_SwCluC_03106] [SWS_SwCluC_03107] [SWS_SwCluC_03108] [SWS_SwCluC_03109] [SWS_SwCluC_03110] [SWS_SwCluC_03111] [SWS_SwCluC_03112] [SWS_SwCluC_03113] [SWS_SwCluC_03120] [SWS_SwCluC_03121] [SWS_SwCluC_03122] [SWS_SwCluC_03123] [SWS_SwCluC_03124] [SWS_SwCluC_03126] [SWS_SwCluC_03130] [SWS_SwCluC_03131] [SWS_SwCluC_03132] [SWS_SwCluC_03133] [SWS_SwCluC_03134] [SWS_SwCluC_03135] [SWS_SwCluC_03136] [SWS_SwCluC_03137] [SWS_SwCluC_03138] [SWS_SwCluC_03139] [SWS_SwCluC_03140] [SWS_SwCluC_03142] [SWS_SwCluC_03143] [SWS_SwCluC_03144] [SWS_SwCluC_03145] [SWS_SwCluC_03146] [SWS_SwCluC_03147] [SWS_SwCluC_03150] [SWS_SwCluC_03151] [SWS_SwCluC_03152] [SWS_SwCluC_03153] [SWS_SwCluC_03154] [SWS_SwCluC_03155] [SWS_SwCluC_03156] [SWS_SwCluC_03157] [SWS_SwCluC_03158] [SWS_SwCluC_03159] [SWS_SwCluC_03160] [SWS_SwCluC_03161] [SWS_SwCluC_03162] [SWS_SwCluC_03163] [SWS_SwCluC_03164] [SWS_SwCluC_03165] [SWS_SwCluC_03166] [SWS_SwCluC_03167] [SWS_SwCluC_03168] [SWS_SwCluC_03169] [SWS_SwCluC_03170] [SWS_SwCluC_03171] [SWS_SwCluC_03173] [SWS_SwCluC_03174] [SWS_SwCluC_03175] [SWS_SwCluC_03176] [SWS_SwCluC_03177] [SWS_SwCluC_11000] [SWS_SwCluC_11001] [SWS_SwCluC_CONSTR_03066] [SWS_SwCluC_CONSTR_03067] [SWS_SwCluC_CONSTR_03069]</p>
[SRS_SwCluC_00101]	'1:n' Sender-receiver communication	<p>[SWS_SwCluC_03034] [SWS_SwCluC_03035] [SWS_SwCluC_03036] [SWS_SwCluC_03037] [SWS_SwCluC_03038] [SWS_SwCluC_03039] [SWS_SwCluC_03040] [SWS_SwCluC_03041] [SWS_SwCluC_03042] [SWS_SwCluC_03043] [SWS_SwCluC_03044] [SWS_SwCluC_03064] [SWS_SwCluC_03065] [SWS_SwCluC_03068] [SWS_SwCluC_03080] [SWS_SwCluC_03081] [SWS_SwCluC_03082] [SWS_SwCluC_03101] [SWS_SwCluC_03102] [SWS_SwCluC_03103] [SWS_SwCluC_03106] [SWS_SwCluC_03107] [SWS_SwCluC_03108] [SWS_SwCluC_03109] [SWS_SwCluC_03110] [SWS_SwCluC_03111] [SWS_SwCluC_03112] [SWS_SwCluC_03113] [SWS_SwCluC_03120] [SWS_SwCluC_03121] [SWS_SwCluC_03122] [SWS_SwCluC_03123] [SWS_SwCluC_03124] [SWS_SwCluC_03126] [SWS_SwCluC_03130] [SWS_SwCluC_03131] [SWS_SwCluC_03132] [SWS_SwCluC_03133] [SWS_SwCluC_03135] [SWS_SwCluC_03136] [SWS_SwCluC_03137] [SWS_SwCluC_03138] [SWS_SwCluC_03139] [SWS_SwCluC_03140] [SWS_SwCluC_03142] [SWS_SwCluC_03143] [SWS_SwCluC_03144] [SWS_SwCluC_03156] [SWS_SwCluC_03157] [SWS_SwCluC_03158] [SWS_SwCluC_03159]</p>
[SRS_SwCluC_00102]	'n:1' Sender-receiver communication	<p>[SWS_SwCluC_03034] [SWS_SwCluC_03035] [SWS_SwCluC_03036] [SWS_SwCluC_03037] [SWS_SwCluC_03038] [SWS_SwCluC_03039] [SWS_SwCluC_03040] [SWS_SwCluC_03041] [SWS_SwCluC_03042] [SWS_SwCluC_03043] [SWS_SwCluC_03044]</p>
[SRS_SwCluC_00103]	'n:1' Client-server communication	<p>[SWS_SwCluC_03035] [SWS_SwCluC_03036] [SWS_SwCluC_03037] [SWS_SwCluC_03038] [SWS_SwCluC_03039] [SWS_SwCluC_03040] [SWS_SwCluC_03041] [SWS_SwCluC_03042] [SWS_SwCluC_03043] [SWS_SwCluC_03044] [SWS_SwCluC_03047] [SWS_SwCluC_03134] [SWS_SwCluC_03145] [SWS_SwCluC_03146] [SWS_SwCluC_03147] [SWS_SwCluC_03150] [SWS_SwCluC_03151] [SWS_SwCluC_03152] [SWS_SwCluC_03153] [SWS_SwCluC_03154] [SWS_SwCluC_03155] [SWS_SwCluC_03160] [SWS_SwCluC_03161] [SWS_SwCluC_03162] [SWS_SwCluC_03163] [SWS_SwCluC_03164] [SWS_SwCluC_03165] [SWS_SwCluC_03166] [SWS_SwCluC_03167] [SWS_SwCluC_03168] [SWS_SwCluC_03169] [SWS_SwCluC_03170] [SWS_SwCluC_03171] [SWS_SwCluC_03173] [SWS_SwCluC_03174] [SWS_SwCluC_03175] [SWS_SwCluC_03176] [SWS_SwCluC_03177]</p>





Requirement	Description	Satisfied by
[SRS_SwCluC_00104]	'1:n' Mode Switch Communication	[SWS_SwCluC_03015] [SWS_SwCluC_03016] [SWS_SwCluC_03017] [SWS_SwCluC_03018] [SWS_SwCluC_03019] [SWS_SwCluC_03022] [SWS_SwCluC_03023] [SWS_SwCluC_03024] [SWS_SwCluC_03025] [SWS_SwCluC_03026] [SWS_SwCluC_03027] [SWS_SwCluC_03028] [SWS_SwCluC_03029] [SWS_SwCluC_03030] [SWS_SwCluC_03031] [SWS_SwCluC_03045] [SWS_SwCluC_03057] [SWS_SwCluC_03061] [SWS_SwCluC_03062] [SWS_SwCluC_03063] [SWS_SwCluC_CONSTR_03020] [SWS_SwCluC_CONSTR_03021] [SWS_SwCluC_CONSTR_03032]
[SRS_SwCluC_00105]	'1:n' External Trigger communication	[SWS_SwCluC_03046] [SWS_SwCluC_03048] [SWS_SwCluC_03049] [SWS_SwCluC_03050] [SWS_SwCluC_03051] [SWS_SwCluC_03052] [SWS_SwCluC_03055] [SWS_SwCluC_03056] [SWS_SwCluC_03058] [SWS_SwCluC_03060] [SWS_SwCluC_CONSTR_03053] [SWS_SwCluC_CONSTR_03054] [SWS_SwCluC_CONSTR_03059]
[SRS_SwCluC_00106]	'1:n' Parameter Communication	[SWS_SwCluC_03006] [SWS_SwCluC_03007] [SWS_SwCluC_03008] [SWS_SwCluC_03009] [SWS_SwCluC_03010] [SWS_SwCluC_03011] [SWS_SwCluC_03034] [SWS_SwCluC_03035] [SWS_SwCluC_03036] [SWS_SwCluC_03037] [SWS_SwCluC_03038] [SWS_SwCluC_03039] [SWS_SwCluC_03040] [SWS_SwCluC_03041] [SWS_SwCluC_03042] [SWS_SwCluC_03043] [SWS_SwCluC_03044]
[SRS_SwCluC_00108]	Prevent from writing directly to memory of other Software Clusters	[SWS_SwCluC_11000] [SWS_SwCluC_11001]
[SRS_SwCluC_00206]	NV blocks in Applicative Software Cluster	[SWS_SwCluC_02101] [SWS_SwCluC_02102] [SWS_SwCluC_02103] [SWS_SwCluC_02104] [SWS_SwCluC_02105] [SWS_SwCluC_02106] [SWS_SwCluC_02107] [SWS_SwCluC_02108] [SWS_SwCluC_02109] [SWS_SwCluC_02110] [SWS_SwCluC_02111] [SWS_SwCluC_02112] [SWS_SwCluC_02113] [SWS_SwCluC_02114] [SWS_SwCluC_02115] [SWS_SwCluC_02116] [SWS_SwCluC_02117] [SWS_SwCluC_02118] [SWS_SwCluC_02119] [SWS_SwCluC_02120] [SWS_SwCluC_02121] [SWS_SwCluC_02122] [SWS_SwCluC_02123] [SWS_SwCluC_02124] [SWS_SwCluC_02125] [SWS_SwCluC_02126] [SWS_SwCluC_02127] [SWS_SwCluC_02128] [SWS_SwCluC_02129] [SWS_SwCluC_02132] [SWS_SwCluC_02133] [SWS_SwCluC_02136] [SWS_SwCluC_02137] [SWS_SwCluC_02138] [SWS_SwCluC_02139] [SWS_SwCluC_02142] [SWS_SwCluC_02144] [SWS_SwCluC_CONSTR_02134] [SWS_SwCluC_CONSTR_02135] [SWS_SwCluC_CONSTR_02141]
[SRS_SwCluC_00207]	LdCom Invocation from Applicative Software Cluster	[SWS_SwCluC_02500] [SWS_SwCluC_02501] [SWS_SwCluC_02502] [SWS_SwCluC_02503] [SWS_SwCluC_02504] [SWS_SwCluC_02505] [SWS_SwCluC_02506] [SWS_SwCluC_02507] [SWS_SwCluC_02508] [SWS_SwCluC_02509] [SWS_SwCluC_02510] [SWS_SwCluC_02511] [SWS_SwCluC_02512] [SWS_SwCluC_02513] [SWS_SwCluC_02514] [SWS_SwCluC_02515] [SWS_SwCluC_02516] [SWS_SwCluC_02517] [SWS_SwCluC_02518] [SWS_SwCluC_02519] [SWS_SwCluC_02520] [SWS_SwCluC_02521] [SWS_SwCluC_02522] [SWS_SwCluC_02523] [SWS_SwCluC_02524] [SWS_SwCluC_02525] [SWS_SwCluC_02526] [SWS_SwCluC_02527] [SWS_SwCluC_02528] [SWS_SwCluC_02529] [SWS_SwCluC_02530] [SWS_SwCluC_02531] [SWS_SwCluC_02532] [SWS_SwCluC_02533] [SWS_SwCluC_02534] [SWS_SwCluC_02535] [SWS_SwCluC_02536] [SWS_SwCluC_02537] [SWS_SwCluC_02538] [SWS_SwCluC_02539] [SWS_SwCluC_02540] [SWS_SwCluC_02541] [SWS_SwCluC_02542] [SWS_SwCluC_02543] [SWS_SwCluC_02544] [SWS_SwCluC_02550] [SWS_SwCluC_02551] [SWS_SwCluC_02552] [SWS_SwCluC_CONSTR_02546] [SWS_SwCluC_CONSTR_02547] [SWS_SwCluC_CONSTR_02555] [SWS_SwCluC_CONSTR_02556] [SWS_SwCluC_CONSTR_02557] [SWS_SwCluC_CONSTR_02558] [SWS_SwCluC_CONSTR_02559] [SWS_SwCluC_CONSTR_02560] [SWS_SwCluC_CONSTR_02561] [SWS_SwCluC_CONSTR_02562]





Requirement	Description	Satisfied by
[SRS_SwCluC_00208]	Dcm contribution by Applicative Software Cluster	[SWS_SwCluC_02300] [SWS_SwCluC_02301] [SWS_SwCluC_02302] [SWS_SwCluC_02303] [SWS_SwCluC_02304] [SWS_SwCluC_02305] [SWS_SwCluC_02306] [SWS_SwCluC_02307] [SWS_SwCluC_02308] [SWS_SwCluC_02309] [SWS_SwCluC_02310] [SWS_SwCluC_02311] [SWS_SwCluC_02312] [SWS_SwCluC_02313] [SWS_SwCluC_02314] [SWS_SwCluC_02315] [SWS_SwCluC_02316] [SWS_SwCluC_02317] [SWS_SwCluC_02318] [SWS_SwCluC_02319] [SWS_SwCluC_02320] [SWS_SwCluC_02321] [SWS_SwCluC_02322] [SWS_SwCluC_02323] [SWS_SwCluC_02324] [SWS_SwCluC_02325] [SWS_SwCluC_02330] [SWS_SwCluC_02331] [SWS_SwCluC_02332] [SWS_SwCluC_02333] [SWS_SwCluC_02334] [SWS_SwCluC_02335] [SWS_SwCluC_02336] [SWS_SwCluC_02337] [SWS_SwCluC_02338] [SWS_SwCluC_02339] [SWS_SwCluC_02341] [SWS_SwCluC_02342] [SWS_SwCluC_02350] [SWS_SwCluC_02351] [SWS_SwCluC_02352] [SWS_SwCluC_02353] [SWS_SwCluC_02354] [SWS_SwCluC_02355] [SWS_SwCluC_02356] [SWS_SwCluC_02357] [SWS_SwCluC_02358] [SWS_SwCluC_02359] [SWS_SwCluC_02360] [SWS_SwCluC_02361] [SWS_SwCluC_02362] [SWS_SwCluC_02363] [SWS_SwCluC_02364] [SWS_SwCluC_02365] [SWS_SwCluC_02366] [SWS_SwCluC_02367] [SWS_SwCluC_02368] [SWS_SwCluC_02369] [SWS_SwCluC_02370] [SWS_SwCluC_02371] [SWS_SwCluC_02372] [SWS_SwCluC_02373] [SWS_SwCluC_02374] [SWS_SwCluC_02375] [SWS_SwCluC_02376] [SWS_SwCluC_02380] [SWS_SwCluC_02381] [SWS_SwCluC_02382] [SWS_SwCluC_02383] [SWS_SwCluC_02384] [SWS_SwCluC_02385] [SWS_SwCluC_02386] [SWS_SwCluC_02387] [SWS_SwCluC_02388] [SWS_SwCluC_02400] [SWS_SwCluC_02401] [SWS_SwCluC_02402] [SWS_SwCluC_02403] [SWS_SwCluC_02404] [SWS_SwCluC_02405] [SWS_SwCluC_02406] [SWS_SwCluC_02407] [SWS_SwCluC_02408] [SWS_SwCluC_02409] [SWS_SwCluC_02410] [SWS_SwCluC_02411] [SWS_SwCluC_02412] [SWS_SwCluC_02413] [SWS_SwCluC_02414] [SWS_SwCluC_02415] [SWS_SwCluC_02416] [SWS_SwCluC_02417] [SWS_SwCluC_02418] [SWS_SwCluC_02419] [SWS_SwCluC_02420] [SWS_SwCluC_02421] [SWS_SwCluC_02422] [SWS_SwCluC_02423] [SWS_SwCluC_02424] [SWS_SwCluC_02425]
[SRS_SwCluC_00209]	Diagnostic Monitors in Applicative Software Cluster	[SWS_SwCluC_02700] [SWS_SwCluC_02701] [SWS_SwCluC_02702] [SWS_SwCluC_02703] [SWS_SwCluC_02704] [SWS_SwCluC_02705] [SWS_SwCluC_02706] [SWS_SwCluC_02707] [SWS_SwCluC_02708] [SWS_SwCluC_02709] [SWS_SwCluC_02710] [SWS_SwCluC_02711] [SWS_SwCluC_02712] [SWS_SwCluC_02713] [SWS_SwCluC_02714] [SWS_SwCluC_02715] [SWS_SwCluC_02716] [SWS_SwCluC_02717] [SWS_SwCluC_02718] [SWS_SwCluC_02719] [SWS_SwCluC_02723] [SWS_SwCluC_02724] [SWS_SwCluC_02725] [SWS_SwCluC_02726] [SWS_SwCluC_02727] [SWS_SwCluC_02728] [SWS_SwCluC_02729] [SWS_SwCluC_02730] [SWS_SwCluC_02731] [SWS_SwCluC_02732] [SWS_SwCluC_02733] [SWS_SwCluC_02734] [SWS_SwCluC_02735] [SWS_SwCluC_02736] [SWS_SwCluC_02737] [SWS_SwCluC_02738] [SWS_SwCluC_02739] [SWS_SwCluC_02740] [SWS_SwCluC_02741] [SWS_SwCluC_02742] [SWS_SwCluC_02743] [SWS_SwCluC_02744] [SWS_SwCluC_02745] [SWS_SwCluC_02746] [SWS_SwCluC_02747] [SWS_SwCluC_02748] [SWS_SwCluC_02749] [SWS_SwCluC_02751]
[SRS_SwCluC_00210]	Usage of Function Inhibitions in Applicative Software Cluster	[SWS_SwCluC_03200] [SWS_SwCluC_03201] [SWS_SwCluC_03202] [SWS_SwCluC_03203] [SWS_SwCluC_03204] [SWS_SwCluC_03205] [SWS_SwCluC_03206] [SWS_SwCluC_03207] [SWS_SwCluC_03208] [SWS_SwCluC_03209] [SWS_SwCluC_03210] [SWS_SwCluC_03211] [SWS_SwCluC_03212] [SWS_SwCluC_03213] [SWS_SwCluC_03214] [SWS_SwCluC_03215] [SWS_SwCluC_03217] [SWS_SwCluC_03218]
[SRS_SwCluC_00211]	Com Invocation from Applicative Software Cluster	[SWS_SwCluC_02548] [SWS_SwCluC_02549] [SWS_SwCluC_02600] [SWS_SwCluC_02601] [SWS_SwCluC_02602] [SWS_SwCluC_02603] [SWS_SwCluC_02604] [SWS_SwCluC_02605] [SWS_SwCluC_02606] [SWS_SwCluC_02607] [SWS_SwCluC_02608] [SWS_SwCluC_02609] [SWS_SwCluC_02610] [SWS_SwCluC_02611] [SWS_SwCluC_02612] [SWS_SwCluC_02613] [SWS_SwCluC_02614] [SWS_SwCluC_02615]



△

Requirement	Description	Satisfied by
		<p style="text-align: center;">△</p> <p>[SWS_SwCluC_02616] [SWS_SwCluC_02617] [SWS_SwCluC_02618] [SWS_SwCluC_02619] [SWS_SwCluC_02620] [SWS_SwCluC_02621] [SWS_SwCluC_02622] [SWS_SwCluC_02623] [SWS_SwCluC_02624] [SWS_SwCluC_02625] [SWS_SwCluC_02626] [SWS_SwCluC_02627] [SWS_SwCluC_02628] [SWS_SwCluC_02629] [SWS_SwCluC_02630] [SWS_SwCluC_02631] [SWS_SwCluC_02632] [SWS_SwCluC_02633] [SWS_SwCluC_02634] [SWS_SwCluC_02636] [SWS_SwCluC_02637] [SWS_SwCluC_02638] [SWS_SwCluC_02639] [SWS_SwCluC_02640] [SWS_SwCluC_02642] [SWS_SwCluC_02643] [SWS_SwCluC_02644] [SWS_SwCluC_02645] [SWS_SwCluC_02646] [SWS_SwCluC_02647] [SWS_SwCluC_02648] [SWS_SwCluC_02649] [SWS_SwCluC_02651] [SWS_SwCluC_02652] [SWS_SwCluC_02661] [SWS_SwCluC_02662] [SWS_SwCluC_02663] [SWS_SwCluC_CONSTR_02554] [SWS_SwCluC_CONSTR_02653] [SWS_SwCluC_CONSTR_02654] [SWS_SwCluC_CONSTR_02655] [SWS_SwCluC_CONSTR_02656] [SWS_SwCluC_CONSTR_02657] [SWS_SwCluC_CONSTR_02658] [SWS_SwCluC_CONSTR_02659] [SWS_SwCluC_CONSTR_02660]</p>
[SRS_SwCluC_00300]	A2L Generation Support	<p>[SWS_SwCluC_03071] [SWS_SwCluC_03072] [SWS_SwCluC_03073] [SWS_SwCluC_03074] [SWS_SwCluC_03075] [SWS_SwCluC_03076] [SWS_SwCluC_03077]</p>

Table 6.1: RequirementsTracing

7 Functional specification

7.1 Binary Manifest

7.1.1 Overview

With the concept of [Software Clusters](#), the overall software of an AUTOSAR Classic Platform Architecture can be split into smaller units. In such a clustered AUTOSAR Classic Platform Architecture a resource is used to describes any capability which

- is needed to operate the software in [Software Clusters](#)

AND

- which is provided by one [Software Cluster](#) for another [Software Clusters](#)

Please note as well document [6] with its description about [CpSoftwareClusterResources](#).

Each [Software Cluster](#) is an independent build unit, and the result of the cluster-specific build processes are the [Binary Objects](#). The [Binary Manifests](#) provide the means to connect the [Binary Objects](#) that are deployed on the same [machine](#). Hence, the [Binary Manifest](#) is the well-defined interface of the [Software Cluster's Binary Object](#).

The [Binary Manifest](#) provides any information, which is required to access a resource inside a [Software Cluster](#), and to connect provided and required resources of [Software Clusters](#). A resource, in this context, can be anything that is required to operate the software. For example, a sender receiver interface of the [Software Cluster](#) or a NV block. An obvious property of such a resource is, whether it is provided or required by a [Software Cluster](#).

The [Binary Manifest](#) has the following core characteristics:

- The [Binary Manifest](#) gets created during the build of the [Software Cluster](#), since it has to store information which might be build dependent (e.g. data and function addresses, ID values to use BSW APIs, attribute values, hashes)
- The [Binary Manifest](#) provides a C interface towards [Software Cluster's](#) implementation. This supports an abstraction in the other functional blocks of the [Software Cluster Connection](#) from the [Binary Manifest's](#) table implementation. The C interface is also accessible by implementations of CDDs requiring the usage of the [Binary Manifest](#).
- The [Binary Manifest](#) defines an ECU-C interface to [Software Cluster's](#) build tools. This supports an abstract usage of the [Binary Manifest](#) by other functional blocks of the [Software Cluster Connection](#), as well as the usage of the [Binary Manifest](#) by CDDs.

- The **Binary Manifest** format can be easily interpreted by the target **machine**. Note: In contrast to concepts for other domains (e.g. Java, Android), in this specification it is not the goal to provide a textual manifest (XML, JSON).
- An unique identifier per resource is used for the connection process. These are explicitly assigned since hash numbers are not suitable.
- The **Binary Manifest** shall have guarding information, to ensure that only compatible interfaces are getting connected. These guarding values consist of a hash over certain interface properties (see 7.3.2.2 and 7.5).

7.1.2 Logical structure of a Binary Manifest

The Figure 7.1 provides an overview of the conceptual elements of the **Binary Manifest** necessary for a fictional example resource.

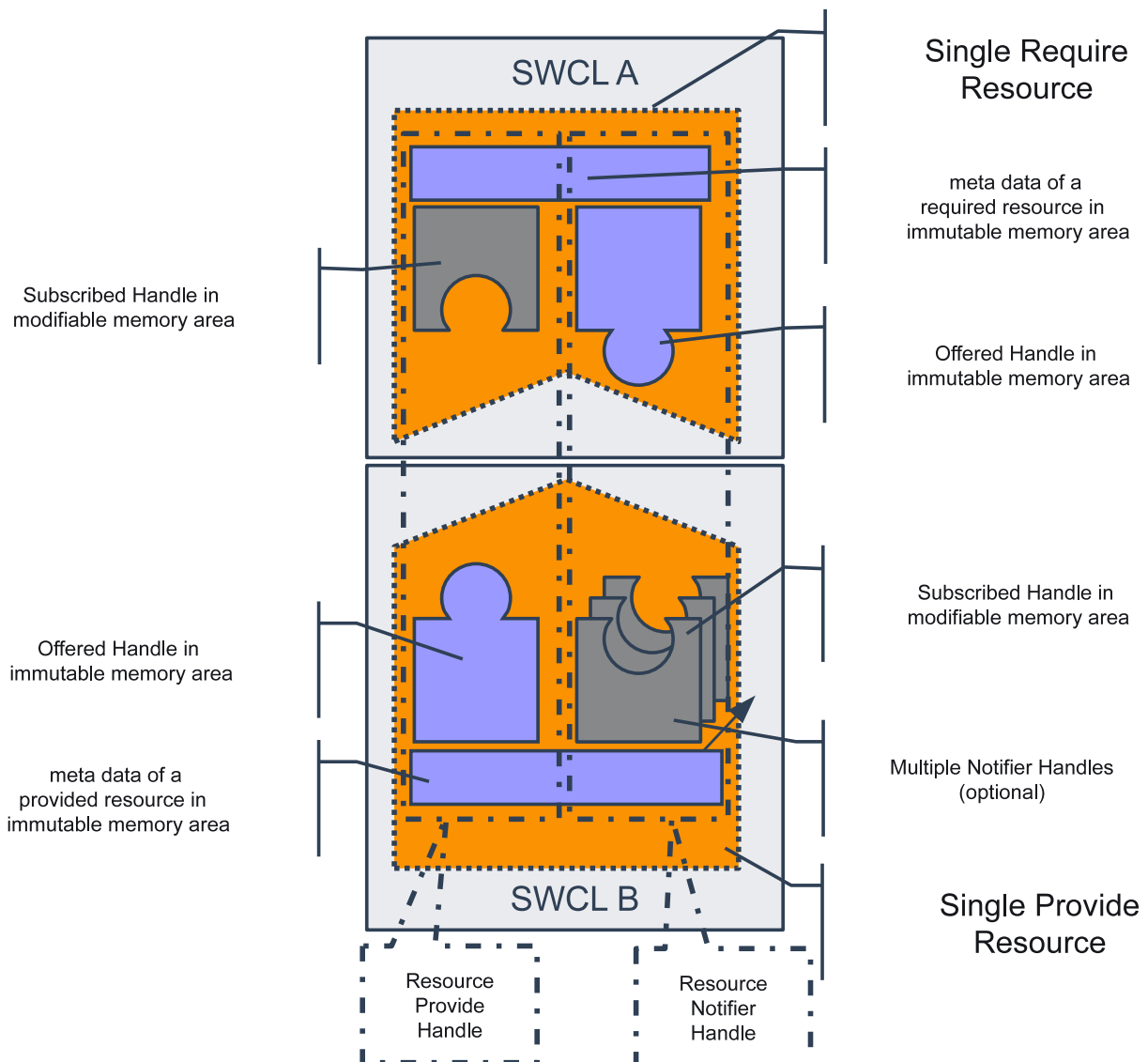


Figure 7.1: Logical structure of a Binary Manifest

The [Binary Manifest](#) provides the ability to connect required and provided resources of Software Clusters. Each resource is qualified by a set of meta data, e.g.

- the unique identifier of the resource
- the type of the resource (e.g. a sender receiver communication or connection of NV blocks)
- the nature, whether a resource is required or provided by this [Software Cluster](#)
- in case of required resources, whether it is mandatory for the operation of the [Software Cluster](#)
- guarding information (e.g. hash values) to ensure that only compatible resources are getting connected
- the number of stored handles, and where those handles can be found in the the according tables

Each type of resource is qualified by the number of handles, their purpose for the connection, and the order in which they are placed in the [Binary Manifest](#). The handles hold the information, which the [Software Cluster](#) requesting a resource needs to access it in the providing [Software Cluster](#).

A single handle can be

- a data pointer
- a function pointer
- a value

In some cases, a connection of a resource may require the exchange of handles in both directions. Consistently the [Binary Manifest](#) distinguishes between two types of handles on the logical level:

- [Provide Handle](#) to publish a handle from resource provider to resource requester
- [Notifier Handle](#) to publish a handle from resource requester to resource provider

An example for a [Provide Handle](#) would be a propagated API function to call an AUTOSAR Service from a [Applicative Software Cluster](#).

An example for a [Notifier Handle](#) would be a callback notification from a connected AUTOSAR Service.

In addition, it is supported that a resource provider gets connected to multiple resource requester even if it utilizes [Notifier Handles](#). But in this case for each potential resource requester a own set of [Notifier Handles](#) needs to be reserved for each resource requester. Hence a [Notifier Handle Set](#) holds all the [Notifier Handles](#) for one resource requester.

The [Binary Manifest](#) is composed of the following parts:

- the [Binary Manifest Header](#). It is the central entry point, provides administrative data and references to the other parts of the manifest.
- the [Interface Descriptor Table](#). It contains one row of meta information per [Resource Entry](#) (required or provided interface) - unique ID, properties to operate the resource at run-time, hash values about interface characteristics and semantics.
- the [Offered Interface](#). It contains all handles that are offered to other clusters - as explained above, either because a resource is provided, or because a required resource also needs a handle in the other direction.
- the [Subscribed Interface](#). Vice versa, it contains all handles that this [Software Cluster](#) holds that were offered by other [Software Clusters](#)
- checksums and markers ([Immutable Tables Checksum](#), [Subscribed Interface Validity Marker](#))

Some parts of the [Binary Manifest](#) are fixed when building a [Software Cluster](#), called immutable, while others have to be changed in the connection process, called modifiable:

- immutable
 - the [Binary Manifest Header](#)
 - the [Interface Descriptor Table](#)
 - offered handles
 - the default value for subscribed handles, to support operation in the unconnected state
- modifiable To establish a connection, this memory area has to store the handles from other SWCLs. Additionally, information might be stored to identify whether the resource is connected at all (only defaults are visible) and to which SWCL.

7.1.3 Mapping between Logical structure and Configuration structure

The logical structure shown in [Figure 7.1](#) is also reflected in the configuration structure of the [Binary Manifest](#).

The set of handles, relevant to connect one type of resource, is defined in the [SwCluCBManifResourceType](#). A [Resource Type](#) is characterized by a defined set of [Provide Handles](#) and [Notifier Handles](#), with specific types in a well defined order.

Handles can optionally exist. This is useful for resources which do not always need the full set of handles in all feature configurations. For instance, sender receiver com-

munication works with or without `sendIndication`. Spending always a new `ResourceTypeId` is not appropriate due to the limited numberspace! But the `Resource Provider` and `Resource Requester` have to have the same mutual understanding of the optionality conditions and both have to see a matching configuration to determine the identical handle configuration for a particular `CpSoftwareClusterResource`.

This is the prerequisite to enable the connection between Software Clusters.

Optional handles are always added after the non-optional ones!

In addition it is highly recommended that the configuration values resulting in different handle configurations are also considered for the guard value calculation.

Each required or provided resource of this `Software Cluster` corresponds to one `Resource Entry` in the `Binary Manifest`.

One `Resource Entry` corresponds to exactly one row in the `Interface Descriptor Table`, plus one or several handles in the `Offered Interface` and / or `Subscribed Interface`. Depending whether the resource is required or provided, `Provide Handles` and `Notifier Handles` are implemented in the `Software Cluster` or expected to be set by the other `Software Cluster`. Please note as well Table 7.1.

Therefore, the configuration provides two distinct definitions for `Provide Resource Entries` as `SwCluCBManifProvideResourceEntry` and `Require Resource Entries` as `SwCluCBManifRequireResourceEntry`.

From a `Binary Manifest` user perspective, it is an important use case to iterate over uniform `Resource Entries` by an index. For this purpose, the configuration provides the ability to group `Resource Entries` of the same `Resource Type` in `Resource Entry Groups`. A `Provide Resource Entry Group`, being a group of `Provide Resource Entries`, is represented as `SwCluCBManifProvideResourceEntryGroup`. Correspondingly, a `Require Resource Entry Group`, being a group of `Require Resource Entries`, is represented as `SwCluCBManifRequireResourceEntryGroup`.

In this and the previous section, three pairs of terms have been introduced, for three different logical layers within the `Binary Manifest`:

1. provide and require resources
2. `Provide Handles` and `Notifier Handles`
3. offered and subscribed handles

A single connection (between provide and require resource, 1), consists of one or more logical channels (provide and notifier handles, 2), which are implemented in the binary manifest (offered and subscribed handles, 3).

An analogy might help to clarify these terms.

Consider a satellite receiver connected to a TV via a cable, which internally contains many wires. The satellite receiver provides 'video', the TV requests it (provide and require resource). To make a single connection between provide and require side, multiple separate data flows (provide and notifier handles) are required. Some transmit the video and audio signal from the provider to the requestor, while others transmit remote control signals in the opposite direction. Both the TV and the satellite receiver have a connector with many pins (provide and notifier handles).

For the 'left audio channel' ([Provide Handle](#)), a certain pin is used. On that pin, the satellite receiver sends a signal (an offered handle implements this [Provide Handle](#)), which the TV reads from the same pin (a subscribed handle implements this [Provide Handle](#)). For the 'remote control signal' ([Notifier Handle](#)) pin, the TV sends the signal (offered handle implements the [Notifier Handles](#)), while the satellite receiver reads the signal (subscribed handle implements the [Notifier Handles](#)).

The possible combinations of the three layers are shown in table [7.1](#).

7.1.4 Implementation structure of a Binary Manifest

Usually, in AUTOSAR the implementation of BSW modules is not standardized. But the various data tables of the [Binary Manifest](#) implement an interface to [Software Cluster](#) connector algorithms. Therefore, the implemented layout and semantic of the tables is standardized in this document. Furthermore, an abstract set of requirements for such connector algorithms is defined in section [7.1.5](#).

The Figure [7.2](#) provides an overview about the individual tables of the [Binary Manifest](#), and their main relationships.

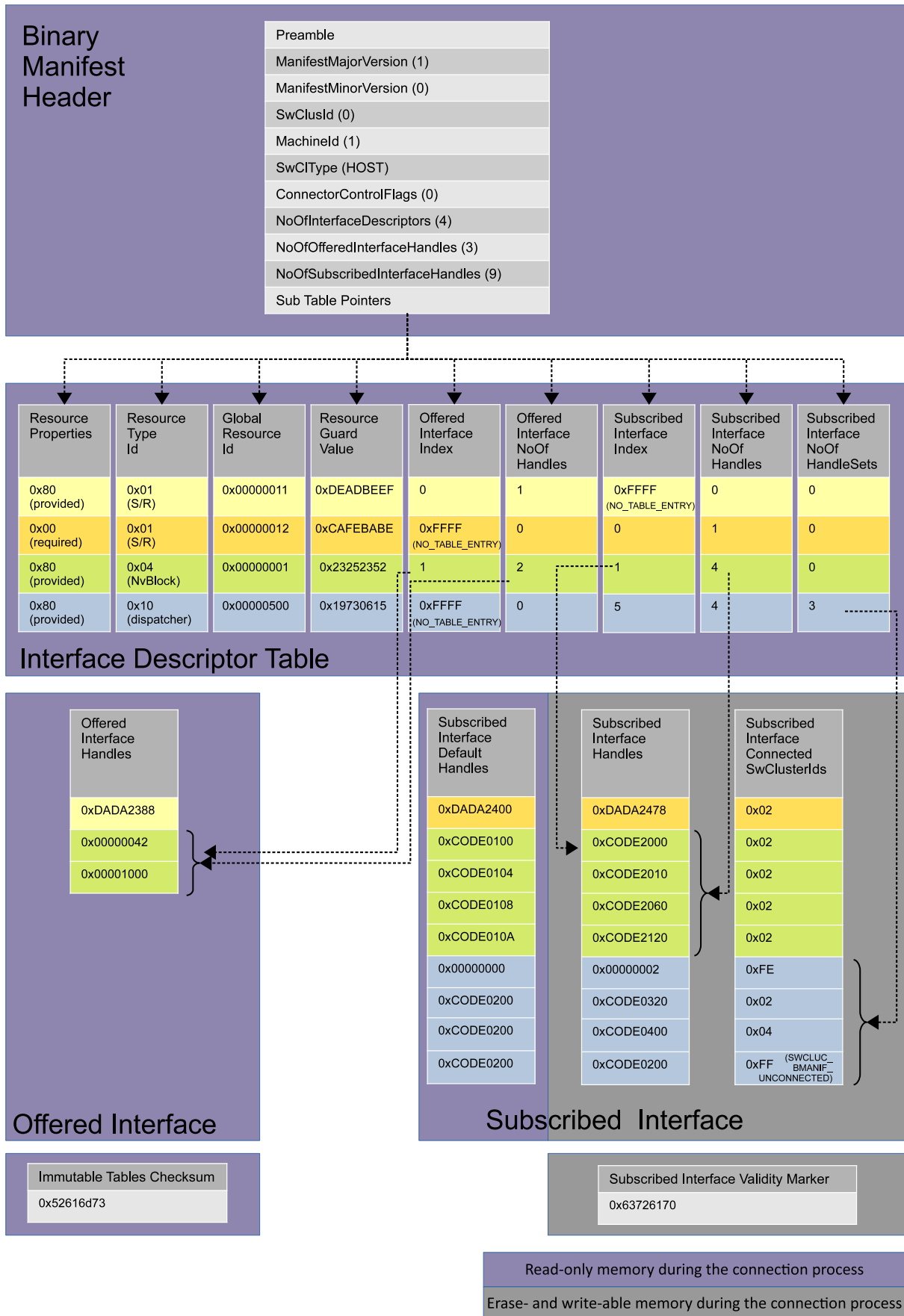


Figure 7.2: Implementation structure of a Binary Manifest

7.1.4.1 Implementation Overview

Each `Software Cluster`'s `Binary Manifest` has a central entry point, called the `Binary Manifest Header`. It contains the AUTOSAR standardized meta data of a `Software Cluster`, and pointers to the sub tables (and their columns, as explained below) of the `Binary Manifest`. It is specified in 7.1.4.3.

Each row in the `Interface Descriptor Table` stores the properties of a single resource. For instance, the global resource Id. To avoid alignment gaps and padding bytes, which could occur if multiple elements are put in a structure, the table is not stored as an array of such a structure. Instead, the individual properties are organized in separate arrays. The individual arrays of the `Interface Descriptor Table` are required to have exactly the same number of elements and the identical order. In other words, these arrays can be seen as the columns of the `Interface Descriptor Table` and the identical index denotes the table row. Each row describes one resource. It is specified in 7.1.4.4.

Each of these resources can now use a dedicated number of handles in the offered and / or subscribed handle tables. The indication of the associated handles is done with the start index for the interface table(s) and the number of handles in the interface table(s). Since the number and placement of offered and subscribed handles can differ, a separate 'index' and a separate 'number of handles' for offered and subscribed handle tables exist. A special `NO_TABLE_ENTRY` value in the index indicates that no handle exists in that table. In this case, the number of handles is set to zero.

For example, in figure 7.2, the resource with id `0x11` (marked yellow) has 1 offered handle at index 0 and no subscribed handle (indicated by `NO_TABLE_ENTRY`). The resource with id `0x1` (marked green), has two handles in the offered table (starting at index 1) and four handles in the subscribed table (also starting at index 1).

The Offered Interface consists of one array - the Offered Interface Handle Column - to store the fixed handles, which are offered for the connection process (for example, in the case of a send port, the addresses of the data buffer related to the port that this software cluster provides). It is specified in 7.1.4.5

The Subscribed Interface consists out of three arrays:

- `Subscribed Interface Default Handle Column` offers default values for the subscribed handles, which are used in case the connection process does not have a connection partner in another `Software Cluster`.
- `Subscribed Interface Handle Column` holds the handles modified by the connection process. Those values are taken during the connection process from the `Offered Interface Handle Column` of another `Software Cluster`.
- The `Subscribed Interface Connected SwClusterId Column` holds the `Software Cluster Id`, from which the handle values are taken.

It is specified in 7.1.4.6.

Please note that the [Subscribed Interface Handle Column](#) and [Subscribed Interface Connected SwClusterId Column](#) need to be located in a memory area that can be erased and re-written during the connection process. In contrast, the [Subscribed Interface Default Handle Column](#) (like all the other [Binary Manifest](#) tables) needs to be located in a memory area which is read-only during the connection process. This split ensures that the connection process can be re-started at any point of time - even if the [Subscribed Interface Handle Column](#) and [Subscribed Interface Connected SwClusterId Column](#) is already erased and not yet re-written.

7.1.4.2 Multiple Notifier Sets Introduction

To support 1:n connections that require individual notifications of the requesters, the [Binary Manifest](#) supports the storage of multiple [Notifier Handle Sets](#). In this case, the `SubscribedInterfaceNoOfHandleSets` field describes how many handles sets of other Software Clusters can be connected. Please note that the value 0 means that only a single connection is supported. 1..n means multiple connections are possible. If this is selected, the first handle entry is used to store the actual number of connected [Software Clusters](#) - similar to a dynamic length array. With this handling, it possible to preserve the `MULTIPLE_NOTIFIER_SETS` semantic for a [Binary Manifest](#) user, even if only at most one connection is supported.

It is also possible to connect interfaces that require more than one handle per connection. The number of handles per connection is not stored in the [Binary Manifest](#), but if it is required, it can be calculated as $(\text{SubscribedInterfaceNoOfHandles} - 1) / \text{SubscribedInterfaceNoOfHandleSets}$.

As an example, in figure 7.2, the resource with id 0x500 (marked blue) has 3 handle sets (the entry in the column `SubscribedInterfaceNoOfHandleSets` is set to 3). So at most three connections can be made to this resource. In this example, each of the three handle set consists of one handle, and since one more row is used for meta information, `SubscribedInterfaceNoOfHandles` is set to $3 * 1 + 1 = 4$.

As explained above, the number of established connections is stored in the first handle entry. In this case, the entry is set to 0x2, so two connections are used and one is unused. The value 0xFE in `SubscribedInterfaceConnectedClusterIds` additionally indicates that this row does not contain a handle. The next two rows each show the target address of the handle and the cluster id. Since the last connection is currently unused, in the last row the handle value is set to the default value, and `SubscribedInterfaceConnectedClusterIds` is set to `SWCLUC_BMANIF_UNCONNECTED (0xFF)`.

More in depth information about multiple [Notifier Handle Sets](#) can be found in 7.1.5.4. As part of that chapter, table 7.2 also shows an example with four sets, where each set consists of two handles.

7.1.4.3 Binary Manifest Header

[SWS_SwCluC_00001]{DRAFT} [The [Binary Manifest](#) of the [Software Cluster Connection](#) shall provide exactly one instance of the [Binary Manifest Header](#)

```
1 const SwCluC_BManif_HeaderType SwCluC_BManif_Header =
2     { <initialization> };
```

]([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00002]{DRAFT} [The header file `SwCluC_BManif.h` shall export the declaration of the [Binary Manifest Header](#)

```
1 extern const SwCluC_BManif_HeaderType SwCluC_BManif_Header;
```

]([SRS_SwCluC_00006](#))

[SWS_SwCluC_00003]{DRAFT} [The element `Preamble` of the [Binary Manifest Header](#) shall be set to the value `0x41524350464C4558`.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#), [SRS_SwCluC_00014](#))

Note: The purpose of the `Preamble` is to serve as an obvious marker of the [Binary Manifest](#)'s begin in memory. In addition, the value is chosen so that byte and word order can be detected, when the [Binary Manifest](#) is read in a byte stream or from a [Binary Object](#) file.

[SWS_SwCluC_00004]{DRAFT} [The element `ManifestMajorVersion` of the [Binary Manifest Header](#) shall be set to the value `0x01`.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00005]{DRAFT} [The element `ManifestMinorVersion` of the [Binary Manifest Header](#) shall be set to the value `0x01`.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00006]{DRAFT} [The element `SwClusterId` of the [Binary Manifest Header](#) shall be set to the value of the configuration parameter `SwCluC-SoftwareClusterId` of the selected [SwCluCDefinition](#).]([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00007]{DRAFT} [The element `MachineId` of the [Binary Manifest Header](#) shall be set to the value of the configuration parameter `SwCluC-MachineId` of the selected [SwCluCDefinition](#).]([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00008]{DRAFT} [The element `SwClusterType` of the [Binary Manifest Header](#) shall be set according to the value of the configuration parameter `SwCluCSoftwareClusterType` of the selected [SwCluCDefinition](#) with the following encoding:

- `HOST_SWCL` : `0x00`

- APPLICATIVE_SWCL : 0x01
- SUBSTITUTION_SWCL : 0xFF

|(SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00013, SRS_SwCluC_00014, SRS_SwCluC_00005)

[SWS_SwCluC_00009]{DRAFT} [The bit SWCLUC_BMANIF_DISABLE_ON_ECU_CONNECTION in the element ConnectorControlFlags of the Binary Manifest Header shall be set if the SwCluCBManifOnBoardConnectorControl is set to DISABLE_ON_ECU_CONNECTOR. Otherwise, the bit is not set (value 0).](SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00013, SRS_SwCluC_00014)

Note: It is not specified, if it is possible to reflash a cluster with changed addresses, how a system would detect this, and how it would behave in this case. This might change in future releases of this specification.

[SWS_SwCluC_00010]{DRAFT} [The element NoOfInterfaceDescriptors of the Binary Manifest Header shall be set to the number of SwCluCBManifProvideResourceEntry + number of SwCluCBManifRequireResourceEntry containers in the configuration.](SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00013, SRS_SwCluC_00014)

[SWS_SwCluC_00011]{DRAFT} [The element NoOfOfferedInterfaceHandles of the Binary Manifest Header shall be set to the number of offered handles.](SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00013, SRS_SwCluC_00014)

[SWS_SwCluC_00012]{DRAFT} [The element NoOfSubscribedInterfaceHandles of the Binary Manifest Header shall be set to the number of handles in the Subscribed Interface.](SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00013, SRS_SwCluC_00014)

[SWS_SwCluC_00070]{DRAFT} [The element ImmutableTablesChecksumPtr of the Binary Manifest Header shall reference the Immutable Tables Checksum.](SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00013, SRS_SwCluC_00014)

[SWS_SwCluC_00072]{DRAFT} [The element SubscribedInterfaceValidityMarkerPtr of the Binary Manifest Header shall reference the Subscribed Interface Validity Marker.](SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00013, SRS_SwCluC_00014)

[SWS_SwCluC_00042]{DRAFT} [The elements

- ResourcePropertiesDescriptorColumnPtr
- ResourceTypeDescriptorColumnPtr
- GlobalResourceIdDescriptorColumnPtr
- ResourceGuardValueDescriptorColumnPtr

- OfferedInterfaceIndexDescriptorColumnPtr
- OfferedInterfaceNoOfHandlesDescriptorColumnPtr
- SubscribedInterfaceIndexDescriptorColumnPtr
- SubscribedInterfaceNoOfHandlesDescriptorColumnPtr
- SubscribedInterfaceNoOfHandleSetsDescriptorColumnPtr
- OfferedInterfaceHandleColumnPtr
- SubscribedInterfaceHandleDefaultColumnPtr
- SubscribedInterfaceHandleColumnPtr
- SubscribedInterfaceConnectedSwClusterIdColumnPtr

of the [Binary Manifest Header](#) shall reference the according array of the [Interface Descriptor Table](#) if the [SwCluCManifestDescriptorTreatment](#) is set to `EMBED_DESCRIPTOR`. Otherwise the elements are initialized to `NULL_PTR`.] ([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#), [SRS_SwCluC_00014](#))

The order of the elements given by the data type [SwCluC_BManifestHeaderType](#).

7.1.4.4 Interface Descriptor Table

[[SWS_SwCluC_00015](#)]{DRAFT} [If the [SwCluCManifestDescriptorTreatment](#) is set to `EMBED_DESCRIPTOR` the [Binary Manifest of the Software Cluster Connection](#) shall provide exactly one instance of the [Interface Descriptor Table](#) composed out of the following column arrays:

```

1  /* descriptor table column for resource properties */
2  const SwCluC_BManifest_ResourcePropertiesType
      SwCluC_BManifest_ResourcePropertiesDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };
3
4  /* descriptor table column for resource type Ids */
5  const SwCluC_BManifest_ResourceTypeIdType
      SwCluC_BManifest_ResourceTypeIdDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };
6
7  /* descriptor table column for global resource Ids */
8  const SwCluC_BManifest_GlobalResourceIdType
      SwCluC_BManifest_GlobalResourceIdDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };
9
10 /* descriptor table column for guard values */
11 const SwCluC_BManifest_ResourceGuardValueType
      SwCluC_BManifest_ResourceGuardValueDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };
12
13 /* descriptor table column for offered interface table index */

```

```

14  const SwCluC_BManif_TableIndexType
      SwCluC_BManif_OfferedInterfaceIndexDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };
15
16  /* descriptor table column for number of handles in offered interface
      table*/
17  const SwCluC_BManif_HandleIndexType
      SwCluC_BManif_OfferedInterfaceNoOfHandlesDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };
18
19  /* descriptor table column for subscribed interface table index */
20  const SwCluC_BManif_TableIndexType
      SwCluC_BManif_SubscribedInterfaceIndexDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };
21
22  /* descriptor table column for number of handles in subscribed
      interface table*/
23  const SwCluC_BManif_HandleIndexType
      SwCluC_BManif_SubscribedInterfaceNoOfHandlesDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };
24
25  /* descriptor table column for number of handle sets in subscribed
      interface table*/
26  const SwCluC_BManif_HandleIndexType
      SwCluC_BManif_SubscribedInterfaceNoOfHandleSetsDescriptorColumn[
      SWCLUC_BMANIF_NO_OF_DESCRIPTOR] = { <initialization> };

```

](SRS_SwCluC_00001, SRS_SwCluC_00014, SRS_SwCluC_00002, SRS_SwCluC_00009, SRS_SwCluC_00010, SRS_SwCluC_00011)

[SWS_SwCluC_00016]{DRAFT} [For each `SwCluC_BManif_ProvideResourceEntry` and `SwCluC_BManif_RequireResourceEntry` in a Software Cluster's configuration, the Binary Manifest of the Software Cluster Connection shall provide one row in the Interface Descriptor Table.](SRS_SwCluC_00001, SRS_SwCluC_00014)

Note: This means that each array in the Interface Descriptor Table gets one element per row. In the below requirements, the term 'element of the X column' refers to one cell of the Interface Descriptor Table at a certain row and column. In this way, the content of each cell of the Interface Descriptor Table is specified.

[SWS_SwCluC_00017]{DRAFT} [The rows in the Interface Descriptor Table shall be sorted in ascending order of resource type Ids and rows with equal resource type Ids shall be sorted in turn in ascending order of global resource Ids.](SRS_SwCluC_00001, SRS_SwCluC_00014)

[SWS_SwCluC_00018]{DRAFT} [The bit `SWCLUC_BMANIF_PROVIDED_RESOURCE` in the element of the 'resource properties' column shall be set, if the row belongs to a `SwCluC_BManif_ProvideResourceEntry`. Otherwise the bit `SWCLUC_BMANIF_PROVIDED_RESOURCE` shall not be set.](SRS_SwCluC_00001, SRS_SwCluC_00014)

[SWS_SwCluC_00019]{DRAFT} [The bit `SWCLUC_BMANIF_MANDATORY_RESOURCE` in the element of the 'resource properties' column shall be set, if the row belongs to a `SwCluCManifestRequireResourceEntry` where the `SwCluCManifestIsMandatory` is `true`. Otherwise the bit `SWCLUC_BMANIF_MANDATORY_RESOURCE` shall not be set.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00009](#))

Please note that for `SwCluCManifestProvideResourceEntry`s the `SWCLUC_BMANIF_MANDATORY_RESOURCE` bit is always 0.

[SWS_SwCluC_00090] [The reserved bits 5 .. 0 in the element of the 'resource properties' column shall be set to 0]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00020]{DRAFT} [The element in the 'resource type Id' column shall be set to the `SwCluCManifestResourceId` of the applicable `SwCluCManifestResourceType`.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#))

Note: The applicable `SwCluCManifestResourceType` is referenced by the owning `SwCluCManifestProvideResourceEntryGroup` / `SwCluCManifestRequireResourceEntryGroup`

[SWS_SwCluC_00021]{DRAFT} [The element in the 'global resource Id' column shall be set to the attribute value `globalResourceId` of the referenced `CpSoftwareClusterResource` as given via `SwCluCManifestResourceRef`. If the reference `SwCluCManifestResourceRef` is not set, the element in the global resource Id column shall be set to 0.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00002](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00022]{DRAFT} [The element in the 'guard value' column shall be set to the `SwCluCManifestResourceGuardValue` of the `SwCluCManifestProvideResourceEntry` / `SwCluCManifestRequireResourceEntry` container in the configuration.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00010](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00023]{DRAFT} [The element in the 'offered interface table index' column shall be set to the index of the `Offered Interface` where the first offered handle for this resource is allocated. In case the resource has no offered handle the value is set to `NO_TABLE_ENTRY`.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00024]{DRAFT} [The element in the 'number of handles in offered interface table' column shall be set to number of offered handles for this resource. In case the resource has no offered handle, the value is set to 0.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00003](#))

[SWS_SwCluC_00025]{DRAFT} [The element in the 'subscribed interface table index' column shall be set to the index of the `Subscribed Interface` where the first subscribed handle for this resource is allocated. In case the resource has no subscribed handle the value is set to `NO_TABLE_ENTRY`.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00026]{DRAFT} [The element in the 'number of handles in subscribed interface table' column shall be set to number of subscribed handles for this

resource. In case the resource has no subscribed handle, the value is set to 0.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00003](#))

[SWS_SwCluC_00027]{DRAFT} [The element in the 'number of handle sets in subscribed interface table' column shall be set to the [SwCluCBManifMaxNumberOfNotifierSets](#) if the applicable [SwCluCBManifResourceType](#) has set [SwCluCBManifMultipleNotifierSupport](#) to [MULTIPLE_NOTIFIER_SETS](#). Otherwise the value is set to 0.]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00004](#))

Depending whether a resource is provided or required by a [Software Cluster](#), the [Provide Handles](#) and [Notifier Handles](#) need to be put either in the [Offered Interface](#) or in the [Subscribed Interface](#). The table [7.1](#) defines, how many of the handles for a [SwCluCBManifProvideResourceEntry](#) / [SwCluCBManifRequireResourceEntry](#) are created in the [Offered Interface](#) or [Subscribed Interface](#).

[SWS_SwCluC_00088]{DRAFT} Number of offered and subscribed handles, depending on the resource direction [

	Provide Handle	Notifier Handle
Provided Resource	<p>One handle in the <i>Offered Interface</i> per defined <i>SwCluCManifProvideHandle</i></p> <p>The handles are initialized with the <i>SwCluCManifProvideSymbols</i> in the given order.</p>	<p>SINGLE_NOTIFIER_SET</p> <p>One handle in the <i>Subscribed Interface</i> per defined <i>SwCluCManifNotifierHandle</i></p> <p>The handles are initialized with the <i>SwCluCManifDefaultNotifierSymbols</i> in the given order.</p> <p>The Software Cluster Ids are set to <i>SWCLUC_BMANIF_UNCONNECTED</i></p>
		<p>MULTIPLE_NOTIFIER_SETS</p> <p>One handle in the <i>Subscribed Interface</i> per defined <i>SwCluCManifNotifierHandle</i> multiplied by <i>SwCluCManifMaxNumberOfNotifierSets</i> plus one</p> <p>The first handle is initialized to 0.</p> <p>The remaining handles are initialized with consecutive sets of <i>SwCluCManifDefaultNotifierSymbols</i> in the given order.</p> <p>The Software Cluster Ids are set to <i>SWCLUC_BMANIF_UNCONNECTED</i>.</p>
Required Resource	<p>One handle in the <i>Subscribed Interface</i> per defined <i>SwCluCManifProvideHandle</i></p> <p>The handles are initialized with the <i>SwCluCManifDefaultProvideSymbols</i> in the given order.</p>	<p>One handle in the <i>Offered Interface</i> per defined <i>SwCluCManifNotifierHandle</i></p> <p>The handles are initialized with the <i>SwCluCManifNotifierSymbols</i> in the given order.</p>

Table 7.1: Number of offered and subscribed handles, depending on the resource direction

]0

7.1.4.5 Offered Interface Table

[SWS_SwCluC_00028]{DRAFT} [The [Binary Manifest of the Software Cluster Connection](#) shall provide exactly one instance of the [Offered Interface Handle Column](#)

```
1 const SwCluC_BManif_HandleType
   SwCluC_BManif_OfferedInterfaceHandleColumn[
   SWCLUC_BMANIF_NO_OF_OFFERED_HANDLES] = { <initialization> };
```

]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00011](#))

[SWS_SwCluC_00029]{DRAFT} [The [Binary Manifest of the Software Cluster Connection](#) shall allocate and initialize the number of handles in the [Offered Interface](#) according to [\[SWS_SwCluC_00088\]](#).]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#))

7.1.4.6 Subscribed Interface Table

[SWS_SwCluC_00030]{DRAFT} [The [Binary Manifest of the Software Cluster Connection](#) shall provide exactly one instance of the [Subscribed Interface Default Handle Column](#)

```
1 const SwCluC_BManif_HandleType
   SwCluC_BManif_SubscribedInterfaceDefaultHandleColumn[
   SWCLUC_BMANIF_NO_OF_SUBSCRIBED_HANDLES] = { <initialization> };
```

]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#), [SRS_SwCluC_00009](#))

[SWS_SwCluC_00031]{DRAFT} [The [Binary Manifest of the Software Cluster Connection](#) shall allocate and initialize the number of handles in the [Subscribed Interface Default Handle Column](#) according to [\[SWS_SwCluC_00088\]](#).]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#))

[SWS_SwCluC_00032]{DRAFT} [The [Binary Manifest of the Software Cluster Connection](#) shall provide exactly one instance of the [Subscribed Interface Handle Column](#)

```
1 const SwCluC_BManif_HandleType
   SwCluC_BManif_SubscribedInterfaceHandleColumn[
   SWCLUC_BMANIF_NO_OF_SUBSCRIBED_HANDLES] = { <initialization> };
```

]([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00013](#))

[SWS_SwCluC_00033]{DRAFT} [The [Binary Manifest of the Software Cluster Connection](#) shall allocate and initialize the number of handles in the [Subscribed Interface Handle Column](#) according to [\[SWS_SwCluC_00088\]](#).]

(*SRS_SwCluC_00001*, *SRS_SwCluC_00014*, *SRS_SwCluC_00011*, *SRS_SwCluC_00013*)

[SWS_SwCluC_00034]{DRAFT} [The *Binary Manifest of the Software Cluster Connection* shall provide exactly one instance of the *Subscribed Interface Connected SwClusterId Column*

```
1 const SwCluC_BManif_SwClusterIdType
   SwCluC_BManif_SubscribedInterfaceConnectedSwClusterIdColumn[
   SWCLUC_BMANIF_NO_OF_SUBSCRIBED_HANDLES] = { <initialization> };
```

](*SRS_SwCluC_00001*, *SRS_SwCluC_00014*, *SRS_SwCluC_00011*, *SRS_SwCluC_00013*, *SRS_SwCluC_00009*, *SRS_SwCluC_00002*)

[SWS_SwCluC_00035]{DRAFT} [The *Binary Manifest of the Software Cluster Connection* shall allocate and initialize the number of handles in the *Subscribed Interface Connected SwClusterId Column* according to **[SWS_SwCluC_00088]**.](*SRS_SwCluC_00001*, *SRS_SwCluC_00014*, *SRS_SwCluC_00011*, *SRS_SwCluC_00013*, *SRS_SwCluC_00009*, *SRS_SwCluC_00002*)

7.1.4.7 Administrative Data

7.1.4.7.1 Immutable Tables Checksum

The *Immutable Tables Checksum* is built over all those constants of the *Binary Manifest* that are not changed by the Software Cluster connection step.

This includes

- the *Binary Manifest Header*
- all arrays of the *Interface Descriptor Table*
- all arrays of the *Offered Interface*
- the *Subscribed Interface Default Handle Column*

The checksum is created as part of the software build.

[SWS_SwCluC_00036]{DRAFT} [The *Binary Manifest of the Software Cluster Connection* shall provide exactly one instance of the *Immutable Tables Checksum*

```
1 const uint32 SwCluC_BManif_ImmutableTablesChecksum = <initialization>;
```

](*SRS_SwCluC_00001*, *SRS_SwCluC_00014*)

[SWS_SwCluC_00037]{DRAFT} [The *Immutable Tables Checksum* shall be set to the value of the configuration parameter *SwCluC_BManif_ImmutableTablesChecksum*.](*SRS_SwCluC_00001*, *SRS_SwCluC_00014*)

[SWS_SwCluC_CONSTR_00073]{DRAFT} [The [Immutable Tables Checksum](#) shall be calculated on the binary representation in memory of the immutable memory area (inclusive reserved memory space) in ascending order of memory address as CRC32 according to 32-bit Ethernet CRC Calculation as described in document [7].] ([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#))

7.1.4.7.2 Subscribed Interface Validity Marker

The [Subscribed Interface Validity Marker](#) indicates that all subscribed tables are written after the Software Cluster connection step. It needs to be set to the valid value after the Software Cluster connection step. The invalid /valid values are not standardized, since those values need to be chosen according to the flash technology storing the [Binary Manifest](#).

[SWS_SwCluC_00040]{DRAFT} [The [Binary Manifest](#) of the [Software Cluster Connection](#) shall provide exactly one instance of the [Subscribed Interface Validity Marker](#)

```
1  const uint32 SwCluC_BManif_SubscribedInterfaceValidityMarker = <
    initialization>;
```

] ([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00041]{DRAFT} [Initially, and after a completed connection process, the [Subscribed Interface Validity Marker](#) shall be set to the value of the configuration parameter `SwCluC_BManif_SubscribedInterfaceValidityMarker`.] ([SRS_SwCluC_00001](#), [SRS_SwCluC_00014](#))

7.1.4.8 Memory Mapping

[SWS_SwCluC_00079]{DRAFT} [The [Binary Manifest](#) of the [Software Cluster Connection](#) shall use the `<feature> = BMANIF` according to [\[SWS_MemMap_00040\]](#) of document [8].] ([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00014](#))

[SWS_SwCluC_00080]{DRAFT} [The [Binary Manifest](#) of the [Software Cluster Connection](#) shall map the [Binary Manifest Header](#) to a constant, 32 bit aligned memory section named `CONST_IMMUTABLE_HEADER`.] ([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00014](#))

Example 7.1

```
1  #define SWCLUC_BMANIF_START_SEC_CONST_IMMUTABLE_HEADER_32
2  #include "SwCluC_MemMap.h"
3
4  const SwCluC_BManif_HeaderType SwCluC_BManif_Header = ...;
5
6  #define SWCLUC_BMANIF_STOP_SEC_CONST_IMMUTABLE_HEADER_32
```

```
7 #include "SwCluC_MemMap.h"
```

[SWS_SwCluC_00081]{DRAFT} [The Binary Manifest of the Software Cluster Connection shall map the immutable columns of all tables to a constant, 32 bit aligned memory section named `CONST_IMMUTABLE_COLUMNS`. The immutable columns are:

- all columns of the Interface Descriptor Table
- Offered Interface Handle Column
- Subscribed Interface Default Handle Column

](*SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00014*)

Example 7.2

```
1 #define SWCLUC_BMANIF_START_SEC_CONST_IMMUTABLE_COLUMNS_32
2 #include "SwCluC_MemMap.h"
3
4 const SwCluC_BManif_ResourcePropertiesType
   SwCluC_BManif_ResourcePropertiesDescriptorColumn[
   SWCLUC_BMANIF_NO_OF_DESCRIPTORS] = ...;
5
6 ...
7
8 const SwCluC_BManif_HandleType
   SwCluC_BManif_SubscribedInterfaceDefaultHandleColumn[
   SWCLUC_BMANIF_NO_OF_SUBSCRIBED_HANDLES] = ...;
9
10 #define SWCLUC_BMANIF_STOP_SEC_CONST_IMMUTABLE_COLUMNS_32
11 #include "SwCluC_MemMap.h"
```

[SWS_SwCluC_00082]{DRAFT} [The Binary Manifest of the Software Cluster Connection shall map the modifiable columns to a constant, 32 bit aligned memory section named `CONST_MODIFIABLE_COLUMNS`. The modifiable columns are

- Subscribed Interface Handle Column
- Subscribed Interface Connected SwClusterId Column

)](*SRS_SwCluC_00001, SRS_SwCluC_00011, SRS_SwCluC_00014*)

Example 7.3

```
1 #define SWCLUC_BMANIF_START_SEC_CONST_MODIFIABLE_COLUMNS_32
2 #include "SwCluC_MemMap.h"
3
4 const SwCluC_BManif_HandleType
   SwCluC_BManif_SubscribedInterfaceHandleColumn[
   SWCLUC_BMANIF_NO_OF_SUBSCRIBED_HANDLES] = ...;
5
```

```

6  const SwCluC_BManif_SwClusterIdType
    SwCluC_BManif_SubscribedInterfaceConnectedSwClusterIdColumn[
        SWCLUC_BMANIF_NO_OF_SUBSCRIBED_HANDLES] = ...;
7
8  #define SWCLUC_BMANIF_STOP_SEC_CONST_MODIFIABLE_COLUMNS_32
9  #include "SwCluC_MemMap.h"

```

[SWS_SwCluC_00083]{DRAFT} [The Binary Manifest of the Software Cluster Connection shall map the Immutable Tables Checksum to a constant, 32 bit aligned memory section named `CONST_IMMUTABLE_TABLES_CHECKSUM`.] ([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00014](#))

Rationale: Checksums handling in memory might need a fixed location.

Example 7.4

```

1  #define SWCLUC_BMANIF_START_SEC_CONST_IMMUTABLE_TABLES_CHECKSUM_32
2  #include "SwCluC_MemMap.h"
3
4  const uint32 SwCluC_BManif_ImmutableTablesChecksum = ...;
5
6  #define SWCLUC_BMANIF_STOP_SEC_CONST_IMMUTABLE_TABLES_CHECKSUM_32
7  #include "SwCluC_MemMap.h"

```

[SWS_SwCluC_00085]{DRAFT} [The Binary Manifest of the Software Cluster Connection shall map the Subscribed Interface Validity Marker to a constant, 32 bit aligned memory section named `CONST_SUBSCRIBED_TABLES_MARKER`.] ([SRS_SwCluC_00001](#), [SRS_SwCluC_00011](#), [SRS_SwCluC_00014](#))

Rationale: Validity handling in memory might need a fixed location.

Example 7.5

```

1  #define SWCLUC_BMANIF_START_SEC_CONST_SUBSCRIBED_TABLES_MARKER_32
2  #include "SwCluC_MemMap.h"
3
4  const uint32 SwCluC_BManif_SubscribedInterfaceValidityMarker = ...;
5
6  #define SWCLUC_BMANIF_STOP_SEC_CONST_SUBSCRIBED_TABLES_MARKER_32
7  #include "SwCluC_MemMap.h"

```

7.1.5 Connecting Software Clusters

7.1.5.1 Overview

The AUTOSAR Methodology supports two different approaches to connect the [Binary Objects](#) of the different [Software Clusters](#).

- The **Software Clusters** are connected after the individual and unconnected **Binary Objects** are programmed on the target ECU, here called **On-board**.
- The **Software Clusters** are connected before the **Binary Objects** are programmed on the target ECU outside the target **machine**, here called **Off-board**.

In the **On-board** case, the **On-board Software Cluster Connector** is a software, which is executed on the target ECU during the reprogramming phase. After reprogramming one or more unconnected **Binary Objects**, the **Software Cluster Connector** stores the connection result into the modifiable area of the **Subscribed Interface** of the **Binary Manifest**.

Please note: The design of the **Binary Manifest** assumes that a connection phase might get aborted at any point in time (e.g. power loss), and a restart or recovery of the connection phase should be supported.

In the **Off-board** case, the **Off-board Software Cluster Connector** is a tool, which is used to read the files of the **Binary Objects**, and to store the result of the connection in files. When reprogramming the ECU, already connected **Binary Objects** are used. Such an **Off-board** process supports additionally the variant to store meta data contained in the **Interface Descriptor Table** outside the **Binary Objects**, with the benefit that no ROM for this meta data is occupied.

Most requirements given in the section are kept universal, and can be applied for the **On-board** and **Off-board** case. If requirements are specific to one case, this is indicated in the requirement. Since AUTOSAR does not standardize the architecture of a flash boot loader, no further interfaces for the **On-board Software Cluster Connector** are standardized.

Some general remarks on **machines**:

Direct connections between **Software Clusters** based on the **Binary Manifest** are only made locally (on the same machine). Connecting **Software Clusters** across 'virtual' machines on the same micro controller is not possible, as it would break the separation between the virtual machines.

7.1.5.2 Connecting Resources

[SWS_SwCluC_00043]{DRAFT} [The **Software Cluster Connector** shall connect a required resource with a provided resource, if all of the following conditions are fulfilled:

- the machine Id of the **Software Cluster** providing the resource and the machine Id of the **Software Cluster(s)** requesting the resource are identical
- the global resource Id of provided resource and required resource(s) are equal
- the global resource Id is not 0

- the resource type Id of provided resource and required resource is equal
- the guard value of provided resource and required resource is equal
- the number of `Provide Handles` of provided resource and required resource is equal
- the number of `Notifier Handles` of provided resource and required resource is equal
- in case `Notifier Handles` exist, the provided resource has a `Notifier Handle Sets` available for the required resource

]([SRS_SwCluC_00002](#))

Note: The global resource Id = 0 is reserved to indicate the absence of a valid Id. The `Software Cluster Connector` shall only connect if the global resource Id != 0.

7.1.5.3 Handling of the `Substitution Software Cluster`

The purpose of the `Substitution Software Cluster` is to temporarily overwrite the output of a `Software Cluster` for development purposes. For instance, if a sender receiver signal 'Y' of Software Cluster A does not have the right functional behavior, it is possible to add a `Substitution Software Cluster` which provides a fixed version of sender receiver signal 'Y'. The fixed version of sender receiver signal 'Y' is provided with the identical global resource Id. In such a case, all `Software Clusters` requiring the sender receiver signal 'Y' are getting connected to the `Substitution Software Cluster` instead of the original providing `Software Cluster`. If the `Substitution Software Cluster` additionally requires the sender receiver signal 'Y', it gets connected to the original providing `Software Cluster`. With this principle, it is possible to take the original version of the signal as a basis for the overwriting one.

The utilization of a `Substitution Software Cluster` is only intended for development purposes, and not supported in productive systems. It is intended as a quick means for problem solving during development (for example, if a signal that is provided by a third party needs to be modified). At a `machine`, at most one `Substitution Software Cluster` is supported to avoid ambiguity about the effective resource provider.

[SWS_SwCluC_CONSTR_00087]{DRAFT} [At a `machine`, at most one `Substitution Software Cluster` shall exist.]([SRS_SwCluC_00005](#))

[SWS_SwCluC_00054]{DRAFT} [In case a resource is provided by the `Substitution Software Cluster`, the `Software Cluster Connector` shall connect all `Software Clusters` requiring the resource, except the `Substitution Software Cluster` itself, to the resource of the `Substitution Software Cluster`. This happens, regardless if another `Software Cluster` provides the resource as well.

Nevertheless, the matching criteria defined in [SWS_SwCluC_00043] apply.](SRS_SwCluC_00005)

[SWS_SwCluC_00055]{DRAFT} [In case a resource is required by a *Substitution Software Cluster*, the *Software Cluster Connector* shall connect this required resource only to provided resources of *Software Clusters* not being a *Substitution Software Cluster*. Nevertheless, the matching criteria defined in [SWS_SwCluC_00043] apply.](SRS_SwCluC_00005)

7.1.5.4 Multiple Notifier Sets

Note: A provided resource can optionally have the ability to handle multiple notifier sets. In this case, the number of handle sets in the *Interface Descriptor Table* indicates how many handle sets the *Software Cluster Connector* can register at most. The first handle in the *Subscribed Interface Handle Column* provides the number of actual registered *Notifier Handle Sets*. The subsequent handles in the *Subscribed Interface Handle Column* are store N sets of handles in a consecutive order. The order of the different handle sets written by the *Software Cluster Connector* are determined by the *Software Cluster Id*.

[SWS_SwCluC_00045]{DRAFT} [In case the provided resource supports multiple *Notifier Handle Sets*, the *Software Cluster Connector* shall sort the *Notifier Handle Sets* in ascending order of *Software Cluster Id* of the according required resource(s), and place them in the *Subscribed Interface Handle Column* of the providing *Software Cluster*, starting at the second handle.](SRS_SwCluC_00004)

[SWS_SwCluC_00046]{DRAFT} [In case the provided resource supports multiple *Notifier Handle Sets*, the *Software Cluster Connector* shall write as first entry the actual number of used *Notifier Handle Sets* in the *Subscribed Interface Handle Column*.](SRS_SwCluC_00004)

Note: the actual number of used *Notifier Handle Sets* is equal to the actual number of connected required resources, since each resource occupies one *Notifier Handle Set*.

[SWS_SwCluC_00089]{DRAFT} [In case the provided resource supports multiple *Notifier Handle Sets*, the *Software Cluster Connector* shall write as first entry in the *Subscribed Interface Connected SwClusterId Column* the value 0xFE.](SRS_SwCluC_00004)

Note: the value 0xFE indicates that the corresponding cell in the *Subscribed Interface Handle Column* holds the actual number of used notifier sets.

[SWS_SwCluC_00047]{DRAFT} [In case the provide resource supports multiple *Notifier Handle Sets*, and less required resources are connected, the *Software Cluster Connector* shall fill the remaining handles in the *Subscribed Interface Handle Column* with the default values from the *Subscribed In-*

terface Default Handle Column of the Software Cluster providing the resource.](SRS_SwCluC_00004)

The example in table 7.2 shows a connection of a resource provider offering at most 4 Notifier Handle sets. The resource type uses two Notifier Handles. The provided resource is connected to two different Software Clusters, and the remaining two Notifier Handle sets are filled with defaults.

Subscribed Interface Handle Column	Subscribed Interface Connected SwClusterId Column	remark
2	0xFE	actual number of used notifier sets, entry in Subscribed Interface Connected SwClusterId Column is set to 0
0xABCDABCD	2	first handle of required resource in Software Cluster 2
0xABC04711	2	second handle of required resource in Software Cluster 2
0xDADA0010	5	first handle of required resource in Software Cluster 5
0xDADA0020	5	second handle of required resource in Software Cluster 5
0xC0CAC01A	0xFF	first default handle of resource provider, entry in Subscribed Interface Connected SwClusterId Column is set to SWCLUC_BMANIF_UNCONNECTED
0xADD511FE	0xFF	second default handle of resource provider, entry in Subscribed Interface Connected SwClusterId Column is set to SWCLUC_BMANIF_UNCONNECTED
0xC0CAC01A	0xFF	first default handle of resource provider, entry in Subscribed Interface Connected SwClusterId Column is set to SWCLUC_BMANIF_UNCONNECTED
0xADD511FE	0xFF	second default handle of resource provider, entry in Subscribed Interface Connected SwClusterId Column is set to SWCLUC_BMANIF_UNCONNECTED

Table 7.2: Example of multiple notifier set connection

7.1.5.5 Unconnected Resources

[SWS_SwCluC_00044]{DRAFT} [The Software Cluster Connector shall copy the Subscribed Interface Default Handle Column entries to the Subscribed Interface Handle Column, and set the related entries in the Subscribed Interface Connected SwClusterId Column to SWCLUC_BMANIF_UNCONNECTED for

- all Provide Handles of any required resources, for which no connect was applied
- AND
- all Notifier Handles of any provided resources, for which no connect was applied

](SRS_SwCluC_00009)

7.1.5.6 Disabling of the On-board Software Cluster Connection

In general, it is possible that an ECU is equipped with an [On-board Software Cluster Connector](#) but this connector should not be used or shall only be used on specific [machines](#).

One reason to do so is a configuration where the software on one [machine](#) requires a certification applied on the whole [machine](#)'s software whereas the software on the other [machine](#) can be updated cluster-wise.

For example, such a configuration uses two [machines](#) on the same microcontroller, here called `Machine A` and `Machine B`.

`Machine A` needs a certification, which is only granted for the whole software on the [machine](#). Nevertheless, the software is developed by multiple parties, with the means of [Software Clusters](#). The [Software Clusters](#) still needs to be connected, which is done with the [Off-board Software Cluster Connector](#) before the certification is done.

For `Machine B`, this restriction does not apply. Here, the [On-board Software Cluster Connector](#) shall be used to do some in-field partial updates which requires a working [On-board Software Cluster Connector](#). Since the flash-boot-loader for the whole microcontroller still has to work, the [On-board Software Cluster Connector](#) is disabled for `Machine A`, in order to protect it from unintended changes, caused by an [On-board Software Cluster Connector](#) run intended for `Machine B`.

[SWS_SwCluC_00056]{DRAFT} [If the bit `SWCLUC_BMANIF_DISABLE_ON_ECU_CONNECTION` in the `ConnectorControlFlags` of the [Binary Manifest Header](#) of the [Host Software Cluster](#) is set, the [Software Cluster Connection On-board](#) for this [machine](#) is not executed.]()

7.1.5.7 Errors during software cluster connection

In some cases, the information inside the [Binary Manifests](#) might be incompatible with the cluster connector, be incomplete or inconsistent. For example, reading a [Binary Manifest](#), which is structurally incompatible to the [Software Cluster Connector](#)'s implementation, may have completely undefined results. This section lists a set of conditions that lead to an abort of the connection process.

The behavior of the system after an abort is currently not specified. The implementation of recovery mechanisms is advisable.

[SWS_SwCluC_00048]{DRAFT} [The [Software Cluster Connector](#) shall detect, if the `ManifestMajorVersion` and `ManifestMinorVersion` numbers given in the [Binary Manifest Header](#) are compatible to the [Software Cluster Connector](#)'s implementation. In case of any incompatible version number, the connection process for this specific [machine](#) shall be aborted.]()

If the Software Cluster Ids are not unique on the same `machine`, the `Subscribed Interface Connected SwClusterId Column` cannot be set unambiguous:

[SWS_SwCluC_00049]{DRAFT} [The Software Cluster Connector shall detect, if the Software Cluster Ids of all `Software Clusters` belonging to same `machine` are not unique. In this case, the connection process for this specific `machine` shall be aborted.]()

If a `Software Cluster` requires a mandatory resource that no other `Software Cluster` provides, the operation of the `Software Cluster` is not possible.

[SWS_SwCluC_00050]{DRAFT} [The Software Cluster Connector shall detect, if for a `Software Cluster` any mandatory required resource is not provided by any other `Software Cluster`. In this case, the connection process for this specific `Software Cluster` shall be aborted.]()

In general, a resource shall only be provided once. The only exception is a second provision by the `Substitution Software Cluster`, which overrides the provision by another `Software Cluster`.

[SWS_SwCluC_00051]{DRAFT} [The Software Cluster Connector shall detect, if for a `machine` a resource with the identical resource type id and identical global resource Id is provided twice by `Software Clusters`, none of which is a `Substitution Software Cluster`. In this case, the connection process for this specific `machine` shall be aborted.]()

[SWS_SwCluC_00052]{DRAFT} [In case the resource does not support multiple notifier sets, but uses `Notifier Handles`, the Software Cluster Connector shall detect if for a `machine` a resource with the identical resource type id and identical global resource Id is required more than one time. In this case, the connection process for this specific `machine` shall be aborted.]()

[SWS_SwCluC_00053]{DRAFT} [In case the resource does support multiple notifier sets, the Software Cluster Connector shall detect if for a `machine` a resource with the identical resource type id and identical global resource Id is required more often as notifier sets are available. In this case, the connection process for this specific `machine` shall be aborted.]()

7.1.6 Software Cluster Binary Manifest Descriptor

The Software Cluster Binary Manifest Descriptor describes the `Binary Manifest` in the `Binary Object`. This is mandatory for `Off-board` connection, but also supporting the `On-board` connection, since it enables to apply checks outside the target ECU.

[SWS_SwCluC_00086]{DRAFT} [The `Binary Manifest` of the `Software Cluster Connection` shall provide one instance of the `CpSoftwareClusterBinaryManifestDescriptor`, which holds the description of the generated `Binary Manifest`. This includes

- the `cpSoftwareCluster` reference to the given `CpSoftwareCluster`
- the `softwareClusterId` of the given `CpSoftwareCluster`
- the `metaDataFields` for `IMMUTABLE_TABLES_CHECKSUM` and `SUBSCRIBED_INTERFACE_VALIDITY_MARKER`
- a `provideResources` for each `Provide Resource Entry`
- a `requireResources` for each `Require Resource Entry`
- a `resourceDefinitions` for each `Resource Type`

]()

7.1.7 Error Classification

On source code level, the implementation of the `Binary Manifest` is a set of constant tables accessible with a set of C-Macros. Those in turn need to be capable to be static C-initializers. Since the constant tables cannot change at runtime, any implementation of run-time detecting error code is not useful. Therefore, the following sections do not define error codes, and are marked as not applicable.

7.1.7.1 Development Errors

Development errors are not applicable for the `Binary Manifest` of the `Software Cluster Connection`.

7.1.7.2 Runtime Errors

Runtime errors are not applicable for the `Binary Manifest` of the `Software Cluster Connection`.

7.1.7.3 Transient Faults

Transient Faults are not applicable for the `Binary Manifest` of the `Software Cluster Connection`.

7.1.7.4 Production Errors

Production Errors are not applicable for the `Binary Manifest` of the `Software Cluster Connection`.

7.1.7.5 Extended Production Errors

Extended Production Errors are not applicable for the [Binary Manifest](#) of the [Software Cluster Connection](#).

7.2 Software Cluster Base Configuration Check

The [Software Cluster Base Configuration Check](#) provides a basic mechanism, to avoid that [Applicative Software Clusters](#) with severe configuration incompatibilities are connected to the [Host Software Cluster](#). This mechanism utilizes the guard value and mandatory feature of the [Binary Manifest](#). The configuration information collected in the two parameters [SwCluCAutoBaseConfigDescriptor](#) and [SwCluCUserBaseConfigDescriptor](#) is used to calculate the guard value for a mandatory connection.

In case of deviating configurations in the [Host Software Cluster](#) and [Applicative Software Cluster](#), the connection fails. In this case, the whole [Applicative Software Cluster](#) stays unconnected, and is skipped in the connection process (treated, as if it was not present). Please note as well [[SWS_SwCluC_00050](#)].

It is implementation dependent, which configuration settings are automatically collected into the [SwCluCAutoBaseConfigDescriptor](#). But it is strongly recommended, to consider only those ECU / [machine](#) wide settings that are functional critical, in order to avoid frequent incompatibilities. For instance, compiler settings impacting the EAIB compatibility, like fill byte usage, stack usage, machine register usage.

[SWS_SwCluC_00075]{DRAFT} [In the [Host Software Cluster](#), the [Software Cluster Base Configuration Check](#) of the [Software Cluster Connection](#) shall create one [Provide Resource Entry](#) for each configured [SwCluCBaseConfigurationCheck](#), and add it to the [Binary Manifest](#).]()

[SWS_SwCluC_00076]{DRAFT} [In the [Applicative Software Cluster](#), the [Software Cluster Base Configuration Check](#) of the [Software Cluster Connection](#) shall create one **mandatory** [Require Resource Entry](#) for each configured [SwCluCBaseConfigurationCheck](#), and add them to the [Binary Manifest](#).]()

[SWS_SwCluC_00077]{DRAFT} [The [SwCluCBManifResourceGuardValue](#) of the [Resource Entries](#) of [[SWS_SwCluC_00075](#)] and [[SWS_SwCluC_00076](#)] shall be calculated out of the given multi line strings in [SwCluCAutoBaseConfigDescriptor](#) and [SwCluCUserBaseConfigDescriptor](#).]()

7.3 Cross Cluster Communication

7.3.1 Overview

The Cross Cluster Communication implements the VFB communication between [Software Clusters](#). For this purpose, the Cross Cluster Communication implements a [Cross Software Cluster Communication Plug-In](#), as specified in document [9].

Due to the intended separation of [Software Clusters](#) (e.g. well separated build units, independent timing), some VFB communication features would either become very inefficient or would break the intended separation. Hence, this specification selected a subset of the most common VFB features expected for such systems, as explained in document [1].

Those are in brief:

- 1:n last is the best sender-receiver communication
- n:1 queued sender-receiver communication
- n:1 client server communication - asynchronous server calls only
- 1:n mode switch communication
- 1:n parameter communication
- limited applicability of [PortInterfaceMappings](#), see [7.3.2.2](#)
- [DataReceivedEvents](#) are polled
- [AsynchronousServerCallReturnsEvent](#) are polled
- [OperationInvokedEvents](#) for Cross Cluster Communication calls are polled

When designing an implementation of a Cross Cluster Communication, this intended separation has to be considered. Since each [Software Clusters](#) is built by its own, only a limited knowledge about the other [Software Clusters](#) is in place. It is also not the target of this Cross Cluster Communication to break the goal of independent development of [Software Clusters](#) by exchange of detailed [Software Clusters](#) internal design and configuration information.

Instead, the code design patterns have to deal with limited knowledge about the communication counter part.

Due to the limited knowledge about the communication partner, the execution of code might fail in partitioned systems.

[SWS_SwCluC_CONSTR_03066]{DRAFT} [The implementation of a Cross Cluster Communication of an [Applicative Software Clusters](#) shall not execute code belonging to any other [Software Cluster](#), except from special entry

points provided by the `Host Software Cluster` (e.g. to implement event passing)] (*SRS_SwCluC_00100*)

For this reason, `SynchronousServerCallPoints` are not supported for `Cross Cluster Communication`. Please note as well [SWS_Rte_72023] in document [9].

In addition, inside a partitioned system it is not guaranteed that the RAM is writable for each `Software Cluster` or any given partition of a `Software Cluster`.

[SWS_SwCluC_CONSTR_03067]{DRAFT} [The implementation of a `Cross Cluster Communication` of an `Applicative Software Cluster` shall only write into the memory of other `Software Clusters`, if that write does not cross a `Partition` border. To determine this, the `Applicative Software Cluster` needs to determine if the `CpSoftwareClusterResource` is assigned to the same partition.] (*SRS_SwCluC_00100*)

Please note: It might be required to restrict some communication patterns to intra-partition communication, for example to reach a certain performance. But this decreases the overall flexibility in the design and integration of `Software Clusters`, and therefore should only be done in rare, exceptional cases.

7.3.2 General Requirements

7.3.2.1 Cross Software Cluster Communication Plug-In

[SWS_SwCluC_03000]{DRAFT} [The `Cross Cluster Communication` of the `Software Cluster Connection` shall implement a `Cross Software Cluster Communication Plug-In` according to document [9].] (*SRS_SwCluC_00100*)

[SWS_SwCluC_03000] means among other things:

[SWS_SwCluC_03001]{DRAFT} [The `Vendor Specific Module Definition` (see document [10]) of the `Software Cluster Connection` shall provide a definition of the container `RteRipsPluginProps` as a `subContainer` of the container `SwCluCXcc` with the `shortName` `RteRipsPluginProps`.] (*SRS_SwCluC_00100*)

Please note that the `shortName` of the `EcucContainerValue` related to the `RteRipsPluginProps` container can be freely chosen. This name will be used to determine the name part `<PlugIn>` in the `RTE Implementation Plug-In Services` according to [SWS_Rte_70034].

[SWS_SwCluC_CONSTR_03069]{DRAFT} [The `EcucModuleConfigurationValues` of the `SwCluC` module shall set the `RteRipsPluginCommunicationScope` to `RTE_RIPS_CROSS_SW_CLUSTER_COM`.] (*SRS_SwCluC_00100*)

Note: It is possible to use multiple `Cross Software Cluster Communication Plug-Ins` in one `Software Cluster`, but this might be rarely used in practice.

Additionally, the Cross Cluster Communication of the Software Cluster Connection has to provide all the RTE Implementation Plug-In Services according to the given input configuration.

[SWS_SwCluC_03002]{DRAFT} [The Cross Cluster Communication of the Software Cluster Connection shall implement the RTE Implementation Plug-In functionality for a Communication Graph, if

- a FlatInstanceDescriptor references the Communication Graph

AND

- and the rtePluginProps.associatedCrossSwClusterComRtePlugin references the RteRipsPluginProps container of this Cross Cluster Communication implementation.

](SRS_SwCluC_00100)

7.3.2.2 Cross Cluster Communication and Binary Manifest

[SWS_SwCluC_03003]{DRAFT} [The Cross Cluster Communication of the Software Cluster Connection shall put one Resource Entry for each associated Communication Graph into the Binary Manifest.](SRS_SwCluC_00100)

[SWS_SwCluC_03004]{DRAFT} [The SwCluCBManifResourceRef of the Resource Entry shall reference the CpSoftwareClusterCommunicationResource which is mapped by the PortElementToCommunicationResourceMapping to the Communication Graph.](SRS_SwCluC_00100)

[SWS_SwCluC_03005]{DRAFT} [The SwCluCBManifResourceGuardValue of the Resource Entry shall be calculated out of the following properties:

- data range
- resolution
- physical meaning
- encoding
- structure in memory

](SRS_SwCluC_00010)

[SWS_SwCluC_03033]{DRAFT} [The Cross Cluster Communication of the Software Cluster Connection shall normalize the numerical values of attributes, before using them for the SwCluCBManifResourceGuardValue calculation, in such a way that compatible numerical values according to [TPS_GST_02501] result in the identical guard value.](SRS_SwCluC_00010)

Note: [SWS_SwCluC_03033] shall ensure that the calculated guard value result is independent from the number format (e.g. decimal, binary, octal and hexadecimal)

of the given attributes. For instance the values, 1, 1.0, 0x000001 are semantically equivalent, and have to result in the identical guard value contribution.

Additionally, the below requirements on the guard value calculation in some cases define default values for optional existing attributes. This ensures an identical guard value on provider an requester side, even if one side expresses the default value explicitly and the other side omits it.

Only in special cases, the `shortName` of AUTOSAR model elements has an explicit semantic. Hence, `shortNames` are not relevant for the guard value, except if it is explicitly specified.

The list of attributes relevant for the guard value, specified in the following requirements, guarantee that different implementations of `Software Cluster Connections` treat the same set of required and provided ports as compatible, if these conditions apply:

- the elements in the `PortInterfaces` are compatible according to document [11] without `PortInterfaceMappings`
- or if a `PortInterfaceMappings` would be applied, it does not result in
 - Data Scaling
NOR
 - utilization of a `SubElementMapping`
NOR
 - utilization of a `ClientServerOperationMapping`
NOR
 - utilization of a `ClientServerApplicationErrorMapping`
NOR
 - utilization of a `ModeDeclarationMappingSet`

Please note:

In case the following requirements about guard value calculation list references as attributes, the calculation has to consider the attributes of the referenced model element. The fully qualified `shortName` (building the reference) is not part of the guard value. For example, the attribute `requiredInterface` is a reference to a `PortInterface`. In this case, the mentioned attributes of the referenced `PortInterface` are relevant instead of the fully qualified `shortName` (e.g. `/Example/PortInterfaces/MyS-RInterface`)

7.3.2.2.1 Guard value calculation for Data Communication Graphs

[SWS_SwCluC_03034]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the `SwCluCManifResourceGuardValue` calculation for a Data Communication Graph:

- `PPortPrototype.providedInterface`
- `RPortPrototype.requiredInterface`
- `CpSoftwareClusterCommunicationResource.comProps.sendIndication`
 - in case of `SenderReceiverInterface`
 - * `VariableDataPrototype.type`
 - * `VariableDataPrototype.swDataDefProps`
 - * `SenderReceiverInterface.invalidationPolicy.handleInvalid` (if existing for the `VariableDataPrototype`, else `dontInvalidate` applies)
 - in case of `NvDataInterface`
 - * `VariableDataPrototype.type`
 - * `VariableDataPrototype.swDataDefProps`
 - in case of `ParameterInterface`
 - * `ParameterDataPrototype.type`
 - * `ParameterDataPrototype.swDataDefProps`

Additionally, attributes of `VariableDataPrototype` / `ParameterDataPrototype.type` according to [SWS_SwCluC_03035] and [SWS_SwCluC_03036] apply.

|(SRS_SwCluC_00010, SRS_SwCluC_00101, SRS_SwCluC_00102, SRS_SwCluC_00106)

7.3.2.2.2 Guard value calculation for Client Server Communication Graphs

[SWS_SwCluC_03047]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the `SwCluCManifResourceGuardValue` calculation for a Client Server Communication Graph:

- `PPortPrototype.providedInterface`
- `RPortPrototype.requiredInterface`
 - `ClientServerOperation.arguments` {ordered}

- * `ArgumentDataPrototype.type`
- * `ArgumentDataPrototype.direction`
- `ClientServerOperation.possibleError.errorCode` (if existing, else one `errorCode = 0` is applied)

Additionally, attributes of `ArgumentDataPrototype.type` according to [\[SWS_SwCluC_03035\]](#) and [\[SWS_SwCluC_03036\]](#) apply.

]([SRS_SwCluC_00010](#), [SRS_SwCluC_00103](#))

7.3.2.2.3 Guard value calculation for Mode Communication Graphs

[SWS_SwCluC_03045]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the `SwCluCManifestResourceGuardValue` calculation for a Mode Communication Graph:

- `PPortPrototype.providedInterface`
- `RPortPrototype.requiredInterface`
- `ModeDeclarationGroupPrototype.type`
- `ModeDeclarationGroup.category`
- `ModeDeclarationGroup.initialMode`
- `ModeDeclarationGroup.modeDeclarations`
- `ModeDeclarationGroup.onTransitionValue` (in case `ModeDeclarationGroup.category == EXPLICIT_ORDER`)
- `ModeDeclarationGroup.modeUserErrorBehavior` (if existing)
- `ModeDeclarationGroup.modeManagerErrorBehavior` (if existing)
- `ModeDeclarationGroup.modeTransitions` (if existing) for each `ModeDeclaration` the following attributes apply:

- `ModeDeclaration.shortName`
- `ModeDeclaration.value` (in case `ModeDeclarationGroup.category == EXPLICIT_ORDER`)

for each `ModeErrorBehavior` the following attributes apply:

- `ModeErrorBehavior.errorReactionPolicy`
- `ModeErrorBehavior.defaultMode` (if existing)

for each `ModeTransition` the following attributes apply:

- `ModeTransition.exitedMode`

- `ModeTransition.enteredMode`

](*SRS_SwCluC_00010*, *SRS_SwCluC_00104*)

7.3.2.2.4 Guard value calculation for Trigger Communication Graphs

[SWS_SwCluC_03046]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the `SwCluCBManifResourceGuardValue` calculation for a Trigger Communication Graph:

- `PPortPrototype.providedInterface`
- `RPortPrototype.requiredInterface`
- `Trigger.swImplPolicy` (if existing, else `standard` applies)

](*SRS_SwCluC_00010*, *SRS_SwCluC_00105*)

7.3.2.2.5 Guard value calculation for DataPrototypes

[SWS_SwCluC_03035]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the referenced `ApplicationDataType` for the `SwCluCBManifResourceGuardValue` calculation:

- in case the `AutosarDataPrototype.type` refers to a `ApplicationPrimitiveDataType`
 - `ApplicationPrimitiveDataType.category`
 - `ApplicationPrimitiveDataType.swDataDefProps`
- in case the `AutosarDataPrototype.type` refers to a `ApplicationArrayDataType`
 - `ApplicationArrayDataType.dynamicArraySizeProfile` (if existing)
 - `ApplicationArrayDataType.element`
 - `ApplicationArrayElement.type`
 - `ApplicationArrayElement.arraySizeSemantics` (if existing, else `fixedSize` applies)
 - `ApplicationArrayElement.arraySizeHandling` (if existing)
 - `ApplicationArrayElement.maxNumberOfElements`
- in case the `AutosarDataPrototype.type` refers to a `ApplicationRecordDataType`

- `ApplicationRecordDataType.elements` {ordered}
- `ApplicationRecordElement.type`
- `ApplicationRecordElement.isOptional` (if existing, else false applies)

Additionally, the attributes of the mapped `ImplementationDataType` according to [SWS_SwCluC_03036] and the attributes of `SwDataDefProps` according to [SWS_SwCluC_03037] apply. (SRS_SwCluC_00010, SRS_SwCluC_00101, SRS_SwCluC_00102, SRS_SwCluC_00103, SRS_SwCluC_00106)

[SWS_SwCluC_03036]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the referenced or mapped `ImplementationDataType` for the `SwCluCManifResourceGuardValue` calculation:

- in case the `AutosarDataPrototype.type` refers to an `ImplementationDataType` of category `TYPE_REFERENCE` or the `ApplicationDataType` according to [SWS_SwCluC_03036] is mapped to an `ImplementationDataType` of category `TYPE_REFERENCE`
 - `ImplementationDataType.swDataDefProps`
- in case the `AutosarDataPrototype.type` refers to an `ImplementationDataType` of category `VALUE` or the `ApplicationDataType` according to [SWS_SwCluC_03036] is mapped to an `ImplementationDataType` of category `VALUE`
 - `ImplementationDataType.swDataDefProps`
- in case the `AutosarDataPrototype.type` refers to an `ImplementationDataType` of category `ARRAY` or the `ApplicationDataType` according to [SWS_SwCluC_03036] is mapped to an `ImplementationDataType` of category `ARRAY`
 - `ImplementationDataType.dynamicArraySizeProfile` (if existing)
 - `ImplementationDataType.swDataDefProps`
 - `ImplementationDataType.subElement`
 - `ImplementationDataTypeElement.arraySizeSemantics` (if existing, else `fixedSize` applies)
 - `ImplementationDataTypeElement.arraySizeHandling` (if existing)
 - `ImplementationDataTypeElement.swDataDefProps`
- in case `AutosarDataPrototype.type` refers to an `ImplementationDataType` of category `STRUCTURE` or the `ApplicationDataType` according to [SWS_SwCluC_03036] is mapped to an `ImplementationDataType` of category `STRUCTURE`
 - `ImplementationDataType.swDataDefProps`

- `ImplementationDataType.isStructWithOptionalElement` (if existing, else `false` applies)
- `ImplementationDataType.subElement` {ordered}
- `ImplementationDataTypeElement.arraySizeSemantics` (if existing, else `fixedSize` applies)
- `ImplementationDataTypeElement.arraySizeHandling` (if existing)
- `ImplementationDataTypeElement.swDataDefProps`

Additionally, the attributes of `SwDataDefProps` according to [SWS_SwCluC_03037] apply.

](*SRS_SwCluC_00010*, *SRS_SwCluC_00101*, *SRS_SwCluC_00102*, *SRS_SwCluC_00103*, *SRS_SwCluC_00106*)

[SWS_SwCluC_03037]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the given `SwDataDefProps` for the `SwCluCBManifResourceGuardValue` calculation:

- `SwDataDefProps.additionalNativeTypeQualifier` (if existing)
- `SwDataDefProps.swImplPolicy` (if existing, else `standard` applies)
- `SwDataDefProps.invalidValue` (if existing)
- `SwDataDefProps.unit` (if existing, else `CompuMethod.unit` applies)
- `SwDataDefProps.dataConstr` (if existing)
- `SwDataDefProps.compuMethod` (if existing)
- `SwDataDefProps.swTextProps` (if existing)
- `SwDataDefProps.baseType` (if existing)
- `SwDataDefProps.swValueBlockSize` (if existing)
- `SwDataDefProps.swValueBlockSizeMult` (if existing)
- `SwDataDefProps.valueAxisDataType` (if existing)
- `SwDataDefProps.implementationDataType` (if existing)

Here, the prioritization of `SwDataDefProps` attributes according to [constr_1015] applies. Only the most prior `SwDataDefProps` attributes are considered.

Additionally, the attributes of `Unit`, `CompuMethod`, `DataConstr`, and `BaseType` according to [SWS_SwCluC_03038], [SWS_SwCluC_03039], [SWS_SwCluC_03042], [SWS_SwCluC_03043], and [SWS_SwCluC_03044] apply.

](*SRS_SwCluC_00010*, *SRS_SwCluC_00101*, *SRS_SwCluC_00102*, *SRS_SwCluC_00103*, *SRS_SwCluC_00106*)

[SWS_SwCluC_03038]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the given `Unit` for the `SwCluCManifResourceGuardValue` calculation:

- `Unit.factorSiToUnit` (if existing, else 1 applies)
- `Unit.offsetSiToUnit` (if existing, else 0 applies)
- `Unit.physicalDimension` (if existing, else [\[SWS_SwCluC_03041\]](#) applies)

Additionally, the attributes of `PhysicalDimension` according to [\[SWS_SwCluC_03040\]](#) and [\[SWS_SwCluC_03041\]](#) apply. ([SRS_SwCluC_00010](#), [SRS_SwCluC_00101](#), [SRS_SwCluC_00102](#), [SRS_SwCluC_00103](#), [SRS_SwCluC_00106](#))

[SWS_SwCluC_03039]{DRAFT} [In case **no** `Unit` is given for the `SwCluCManifResourceGuardValue` calculation, the Cross Cluster Communication shall use the following attribute values:

- `Unit.factorSiToUnit` = 1
- `Unit.offsetSiToUnit` = 0
- `Unit.physicalDimension` according to [\[SWS_SwCluC_03041\]](#)

Additionally, the attributes of `PhysicalDimension` according to [\[SWS_SwCluC_03041\]](#) apply. ([SRS_SwCluC_00010](#), [SRS_SwCluC_00101](#), [SRS_SwCluC_00102](#), [SRS_SwCluC_00103](#), [SRS_SwCluC_00106](#))

[SWS_SwCluC_03040]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the given `PhysicalDimension` for the `SwCluCManifResourceGuardValue` calculation:

- `PhysicalDimension.shortName`
- `PhysicalDimension.currentExp` (if existing, else 0 applies)
- `PhysicalDimension.lengthExp` (if existing, else 0 applies)
- `PhysicalDimension.luminousIntensityExp` (if existing, else 0 applies)
- `PhysicalDimension.massExp` (if existing, else 0 applies)
- `PhysicalDimension.molarAmountExp` (if existing, else 0 applies)
- `PhysicalDimension.temperatureExp` (if existing, else 0 applies)
- `PhysicalDimension.timeExp` (if existing, else 0 applies)

]([SRS_SwCluC_00010](#), [SRS_SwCluC_00101](#), [SRS_SwCluC_00102](#), [SRS_SwCluC_00103](#), [SRS_SwCluC_00106](#))

[SWS_SwCluC_03041]{DRAFT} [In case **no** `PhysicalDimension` is given, the Cross Cluster Communication shall use the following attribute values for the `SwCluCManifResourceGuardValue` calculation:

- `PhysicalDimension.shortName` = NoDimension
- `PhysicalDimension.currentExp` = 0
- `PhysicalDimension.lengthExp` = 0
- `PhysicalDimension.luminousIntensityExp` = 0
- `PhysicalDimension.massExp` = 0
- `PhysicalDimension.molarAmountExp` = 0
- `PhysicalDimension.temperatureExp` = 0
- `PhysicalDimension.timeExp` = 0

|(SRS_SwCluC_00010, SRS_SwCluC_00101, SRS_SwCluC_00102, SRS_SwCluC_00103, SRS_SwCluC_00106)

The guard value calculation has to consider the intended compatibility of `CompuMethod` with different `category`, as defined in document [11] with [constr_1176] and [constr_1192]. Please note as well [constr_1375] w.r.t the existence of attributes of `CompuMethods`. Nevertheless, not all attributes are considered for the guard value, in order to increase compatibility between provider and requesters.

[SWS_SwCluC_03042]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the given `CompuMethod` for the `SwCluCManifResourceGuardValue` calculation:

- `CompuMethod.unit`
- in case the `CompuMethod.category` is IDENTICAL
 - it is treated like a `CompuMethod` where the `category` is LINEAR where the `CompuMethod.compuPhysToInternal.compuScale` yield the conversion

$$int = \frac{N_0 + N_1 * phys}{D_0} \text{ if } N_0 \sim 0 \ \&\& \ N_1 \sim 1 \ \&\& \ D_0 \sim 0$$
- in case the `CompuMethod.category` is LINEAR
 - `CompuMethod.compuPhysToInternal.compuScale`
 - OR
 - `CompuMethod.compuInternalToPhys.compuScale`

where the `compuInternalToPhys` coefficients are converted to equivalent `compuPhysToInternal` coefficients.

 - * `CompuScale.compuNumerator.vs {ordered}`
 - * `CompuScale.compuDenominator.v`
- in case the `CompuMethod.category` is SCALE_LINEAR

- `CompuMethod.compuPhysToInternal.compuScales`
- `CompuMethod.compuPhysToInternal.compuDefaultValue` (if existing)
- OR
- `CompuMethod.compuInternalToPhys.compuScales`
- `CompuMethod.compuInternalToPhys.compuDefaultValue` (if existing)
 - * `CompuScale.upperLimit`
 - * `CompuScale.lowerLimit`
 - * `CompuScale.compuNumerator.vs {ordered}`
 - * `CompuScale.compuDenominator.v`
- in case the `CompuMethod.category` is `RAT_FUNC` with 2 `compuNumerator` coefficients and 1 `compuDenominator` coefficient
 - this case is treated like a `CompuMethod` where the `category` is `LINEAR`
- in case the `CompuMethod.category` is `RAT_FUNC` not matching the `LINEAR` case
 - `CompuMethod.compuPhysToInternal.compuScale`
 - `CompuMethod.compuPhysToInternal.compuDefaultValue` (if existing)
 - AND
 - `CompuMethod.compuInternalToPhys.compuScale`
 - `CompuMethod.compuInternalToPhys.compuDefaultValue` (if existing)
 - * `CompuScale.upperLimit`
 - * `CompuScale.lowerLimit`
 - * `CompuScale.compuInverseValue.vf` (if existing)
 - * `CompuScale.compuNumerator.vs {ordered}`
 - * `CompuScale.compuDenominator.v {ordered}`
- in case the `CompuMethod.category` is `SCALE_RAT_FUNC`
 - `CompuMethod.compuPhysToInternal.compuScales`
 - `CompuMethod.compuPhysToInternal.compuDefaultValue` (if existing)

AND

- `CompuMethod.compuInternalToPhys.compuScaleS`
- `CompuMethod.compuInternalToPhys.compuDefaultValue` (if existing)
 - * `CompuScale.upperLimit`
 - * `CompuScale.lowerLimit`
 - * `CompuScale.compuInverseValue.vf` (if existing)
 - * `CompuScale.compuNumerator.vs {ordered}`
 - * `CompuScale.compuDenominator.v {ordered}`
- in case the `CompuMethod.category` is `TEXTTABLE`
 - `CompuMethod.compuInternalToPhys.compuScaleS`
 - `CompuMethod.compuInternalToPhys.compuDefaultValue` (if existing)
 - * `CompuScale.upperLimit`
 - * `CompuScale.lowerLimit`
 - * `CompuScale.compuInverseValue.vf` (if existing)
 - * `CompuScale.compuConst.vt`
- in case the `CompuMethod.category` is `BITFIELD_TEXTTABLE`
 - `CompuMethod.compuInternalToPhys.compuScaleS`
 - `CompuMethod.compuInternalToPhys.compuDefaultValue` (if existing)
 - * `CompuScale.upperLimit`
 - * `CompuScale.lowerLimit`
 - * `CompuScale.mask`
 - * `CompuScale.compuInverseValue.vf` (if existing)
 - * `CompuScale.compuConst.vt`
- in case the `CompuMethod.category` is `SCALE_LINEAR_AND_TEXTTABLE`
 - `CompuMethod.compuInternalToPhys.compuScaleS`
 - `CompuMethod.compuInternalToPhys.compuDefaultValue` (if existing)
 - * `CompuScale.upperLimit`

- * `CompuScale.lowerLimit`
 - * `CompuScale.mask`
 - * `CompuScale.compuInverseValue.vf` (if existing)
 - * `CompuScale.compuConst.vt` (if existing)
 - * `CompuScale.compuNumerator.vs` {ordered} (if existing)
 - * `CompuScale.compuDenominator.v` (if existing)
- in case the `CompuMethod.category` is `SCALE_RATIONAL_AND_TEXTTABLE`
 - `CompuMethod.compuPhysToInternal.compuScales`
 - `CompuMethod.compuPhysToInternal.compuDefaultValue` (if existing)AND
 - `CompuMethod.compuInternalToPhys.compuScales`
 - `CompuMethod.compuInternalToPhys.compuDefaultValue` (if existing)
 - * `CompuScale.upperLimit`
 - * `CompuScale.lowerLimit`
 - * `CompuScale.mask`
 - * `CompuScale.compuInverseValue.vf` (if existing)
 - * `CompuScale.compuConst.vt` (if existing)
 - * `CompuScale.compuNumerator.vs` {ordered} (if existing)
 - * `CompuScale.compuDenominator.v` {ordered} (if existing)
 - in case the `CompuMethod.category` is `TAB_NOINTP`
 - `CompuMethod.compuInternalToPhys.compuScales`
 - `CompuMethod.compuInternalToPhys.compuDefaultValue` (if existing)
 - * `CompuScale.upperLimit`
 - * `CompuScale.lowerLimit`
 - * `CompuScale.compuInverseValue.vf` (if existing)
 - * `CompuScale.compuConst.vt` (if existing)

Additionally, the attributes of `Unit` according to [[SWS_SwCluC_03038](#)], [[SWS_SwCluC_03039](#)] apply.

](SRS_SwCluC_00010, SRS_SwCluC_00101, SRS_SwCluC_00102, SRS_SwCluC_00103, SRS_SwCluC_00106)

[SWS_SwCluC_03043]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the given `DataConstr` for the `SwCluCManifResourceGuardValue` calculation:

- `DataConstr.dataConstrRule.physConstrs.upperLimit` (if existing)
- `DataConstr.dataConstrRule.physConstrs.lowerLimit` (if existing)
- `DataConstr.dataConstrRule.physConstrs.monotony` (if existing)
- `DataConstr.dataConstrRule.physConstrs.maxGradient` (if existing)
- `DataConstr.dataConstrRule.physConstrs.maxDiff` (if existing)
- `DataConstr.dataConstrRule.internalConstrs.upperLimit` (if existing, else based on `baseTypeEncoding` and `baseTypeSize` the technical upper limit is taken)
- `DataConstr.dataConstrRule.internalConstrs.lowerLimit` (if existing, else based on `baseTypeEncoding` and `baseTypeSize` the technical lower limit is taken)
- `DataConstr.dataConstrRule.internalConstrs.monotony` (if existing, else `noMonotony`)
- `DataConstr.dataConstrRule.internalConstrs.maxGradient` (if existing)
- `DataConstr.dataConstrRule.internalConstrs.maxDiff` (if existing)

](SRS_SwCluC_00010, SRS_SwCluC_00101, SRS_SwCluC_00102, SRS_SwCluC_00103, SRS_SwCluC_00106)

[SWS_SwCluC_03044]{DRAFT} [The Cross Cluster Communication shall use the following attributes for the given `SwBaseType` for the `SwCluCManifResourceGuardValue` calculation:

- `SwBaseType.baseTypeEncoding`
- `SwBaseType.baseTypeSize`
- `SwBaseType.byteOrder`
- `SwBaseType.nativeDeclaration`
- `SwBaseType.memAlignment` (if existing, else 0 applies)

](SRS_SwCluC_00010, SRS_SwCluC_00101, SRS_SwCluC_00102, SRS_SwCluC_00103, SRS_SwCluC_00106)

7.3.2.3 Cross Cluster Communication Base Socket

In general, it is possible that the communication between [Applicative Software Clusters](#) needs some basic infrastructure from the [Host Software Cluster](#). For instance, to pass an activity or event between [Applicative Software Clusters](#). Usually, such event passing requires some knowledge about the overall environment, which is **NOT** available in an [Applicative Software Cluster](#). For instance, in the [Applicative Software Cluster](#), only a subset of the defined partitions (and related [EcucPartitions](#) plus [OsApplications](#)) are known.

[SWS_SwCluC_03012]{DRAFT} [If the implementation of a [Cross Cluster Communication](#) requires some basic infrastructure from the [Host Software Cluster](#), it shall use the means of the [SwCluCXccBaseSocket](#) configuration to link to the resource pool.]()

[SWS_SwCluC_03013]{DRAFT} [The [Cross Cluster Communication](#) of the [Software Cluster Connection](#) shall put one [Resource Entry](#) for each associated [SwCluCXccBaseSocket](#) into the [Binary Manifest](#).]()

7.3.2.4 Cross Cluster Communication and McSupport Data

[SWS_SwCluC_03071]{DRAFT} [The [Cross Cluster Communication](#) of the [Software Cluster Connection](#) shall provide an *MC-Support* (Measurement and Calibration) description as part of its *Basic Software Module Description*.]([SRS_SwCluC_00300](#))

[SWS_SwCluC_03072]{DRAFT} [The [McSupportData](#) element begin part of [\[SWS_SwCluC_03071\]](#) and its sub-structure shall be self-contained, in the sense that there is no need to deliver the whole upstream descriptions of the ECU (including the [ECU Extract](#), [Software Component](#) descriptions, [Basic Software Module](#) descriptions, [ECU Configuration Values](#) descriptions, etc.), in order to later generate the final "A2L"-file. This means that the [Cross Cluster Communication](#) generator has to copy the required information from the upstream descriptions into the [McSupportData](#) element.]([SRS_SwCluC_00300](#))

[SWS_SwCluC_03073]{DRAFT} [The [Cross Cluster Communication](#) generator shall export the effective [SwDataDefProps](#) (including all of the referenced and aggregated sub-elements like e.g. [CompuMethod](#) or [SwRecordLayout](#)) in the role [resultingProperties](#), for each [McDataInstance](#), after resolving the precedence rules defined in the SW-Component Template [\[11\]](#) chapter *Properties of Data Definitions*. Thereby, the [ImplementationDataType](#) properties [compuMethod](#) and [dataConstraint](#) are not taken in consideration for effective [SwDataDefProps](#) of the [McDataInstance](#), due to their refinement nature of **C** and **AI**.]([SRS_SwCluC_00300](#))

[SWS_SwCluC_03074]{DRAFT} [If the parameter [SwCluCXccSwCluCXccDefaultDataHandling](#) is set to `DEFAULTS_AS_CALPRMS`, the [Cross Cluster Communication](#) generator shall export one entry in the [McSupportData](#), describing

the default data instance according to [SWS_SwCluC_03120], with the role `mcParameterInstance`, with the following attributes:

- `swCalibrationAccess` set to `readWrite`
- effective `SwDataDefProps` according to [SWS_SwCluC_03073]
- `McDataInstance.subElements` for array elements or structure elements, if applicable
- `symbol` set to the C-symbol name used for the default data instance

](SRS_SwCluC_00300)

[SWS_SwCluC_03077]{DRAFT} [If the parameter `SwCluCXccSwCluCXccDefaultDataHandling` is set to `DEFAULTS_AS_CONSTANTS`, the Cross Cluster Communication generator shall export one entry in the `McSupportData`, describing the default data instance according to [SWS_SwCluC_03120], with the role `mcParameterInstance`, with the following attributes:

- `swCalibrationAccess` set to `readOnly`
- effective `SwDataDefProps` according to [SWS_SwCluC_03073]
- `McDataInstance.subElements` for array elements or structure elements, if applicable
- `symbol` set to the C-symbol name used for the default data instance

](SRS_SwCluC_00300)

[SWS_SwCluC_03075]{DRAFT} [The Cross Cluster Communication generator shall export one entry in the `McSupportData`, describing the default parameter instance according to [SWS_SwCluC_03006], with the role `mcParameterInstance`, with the following attributes:

- `swCalibrationAccess` set to `readWrite`
- effective `SwDataDefProps` according to [SWS_SwCluC_03073]
- `McDataInstance.subElements` for array elements or structure elements, if applicable
- `symbol` set to the C-symbol name used for the default parameter instance

](SRS_SwCluC_00300)

[SWS_SwCluC_03076]{DRAFT} [If the `swCalibrationAccess` of a `VariableDataPrototype` instance in the delegation `PPortPrototype` at the `CompositionSwComponentType` of the `rootSoftwareComposition` of the Ecu Extract is set to `readOnly` or `readWrite`, the Cross Cluster Communication generator shall export one entry in the `McSupportData`, describing the data instance according to [SWS_SwCluC_03101], with the role `mcVariableInstance`, with the following attributes:

- effective `SwDataDefProps` (inclusive `swCalibrationAccess`) according to [\[SWS_SwCluC_03073\]](#)
- `McDataInstance.subElements` for array elements or structure elements, if applicable
- `symbol` set to the C-symbol name used for the default parameter instance
- if applicable, `flatMapEntry` referencing to the corresponding `FlatInstanceDescriptor` element of the `VariableDataPrototype`

]([SRS_SwCluC_00300](#))

7.3.3 Sender Receiver Communication

7.3.3.1 Restrictions on VFB communication features

[SWS_SwCluC_03064]{DRAFT} [The Cross Cluster Communication of the `Software Cluster Connection` shall reject the configuration, if there is a “N:1” sender-receiver communication with last-is-the-best semantic.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03065]{DRAFT} [The Cross Cluster Communication of the `Software Cluster Connection` shall reject the configuration, if there is a “1:N” sender-receiver communication with event semantic (queued communication).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03068]{DRAFT} [The implementation of a Cross Cluster Communication shall support that `Data Communication Graphs` between `Applicative Software Clusters` can be added, without modification of the `Host Software Cluster`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

7.3.3.2 Transmission

[SWS_SwCluC_03101]{DRAFT} [The Cross Cluster Communication of the `Software Cluster Connection` shall provide one data instance, for each assigned `Data Communication Graph`

- where the referenced `VariableDataPrototype` is owned by a delegation `PPortPrototype` at the `CompositionSwComponentType` at the `CompositionSwComponentType` of the `rootSoftwareComposition` of the `Ecu Extract` AND
- where implicit communication applies to that `VariableDataPrototype` OR
- where explicit unqueued communication applies to that `VariableDataPrototype`

]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03102]{DRAFT} [The data instance created according to [\[SWS_SwCluC_03101\]](#) shall be mapped to a VAR memory section, according to document [\[8\]](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03103]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall make the data instance created according to [\[SWS_SwCluC_03101\]](#), accessible to other [Software Clusters](#) via a [Resource Entry](#) in the [Binary Manifest](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03106]{DRAFT} [When implicit or explicit unqueued communication applies, the `Rte_Rips_Write` shall update the data instance created according to [\[SWS_SwCluC_03101\]](#), and return `RTE_E_OK`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03142]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall provide one [Resource Entry](#), for each assigned [Data Communication Graph](#)

- where the referenced [VariableDataPrototype](#) is owned by a delegation [PPortPrototype](#) at the [CompositionSwComponentType](#) at the [CompositionSwComponentType](#) of the `rootSoftwareComposition` of the `Ecu Extract` AND
- where explicit queued communication applies to that [VariableDataPrototype](#)

This [Resource Entry](#) shall be used to link the access of the related `Rte_Rips_Write` service to the according queue instance on the receiving [Software Cluster](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03107]{DRAFT} [When explicit queued communication applies, and the [Resource Entry](#) created according to [\[SWS_SwCluC_03142\]](#) is not connected, the `Rte_Rips_Write` shall discard the write access, and return `RTE_E_OK`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03108]{DRAFT} [When explicit queued communication applies, and the [Resource Entry](#) created according to [\[SWS_SwCluC_03142\]](#) is connected, the `Rte_Rips_Write` shall check the queue status (See [\[SWS_SwCluC_03135\]](#)), to determine whether or not the data can be enqueued.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03109]{DRAFT} [When explicit queued communication applies, and the [Resource Entry](#) created according to [\[SWS_SwCluC_03142\]](#) is connected, and the queue is not full, the `Rte_Rips_Write` shall enqueue the data and return `RTE_E_OK` (See [\[SWS_SwCluC_03135\]](#)).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03110]{DRAFT} [When explicit queued communication applies, and the `Resource Entry` created according to [\[SWS_SwCluC_03142\]](#) is connected, but the queue is full, the `Rte_Rips_Write` shall discard the write access and return `RTE_E_LIMIT` (See [\[SWS_SwCluC_03135\]](#)).] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03080] [When implicit or explicit unqueued communication applies and the attribute `sendIndication` is set to `anySendOperation` for the `Data Communication Graph`, the call of the `Rte_Rips_Write` shall indicate the occurrence of a send operation for the related data instance.] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

This send operation indication can be used at the receiving `Software Clusters` to determine update flags or data receive event triggers.

Please note that the specification behavior supports a polling of those send indications and therefore an active passing of those events can be avoided.

The value of the `sendIndication` attribute is only relevant for unqueued communication (last-is-the best), since the implementation of a queue on the receiver side provides in any case the possibility to detect enqueue operations.

[SWS_SwCluC_03111]{DRAFT} [The data instance created according to [\[SWS_SwCluC_03101\]](#) shall be initialized according to the `NonqueuedSenderComSpec.initValue` in the `PPortPrototype` at the `CompositionSwComponentType` of the `rootSoftwareComposition` of the `Ecu Extract`.] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03112]{DRAFT} [If acknowledgment is enabled, and the resource created according to [\[SWS_SwCluC_03103\]](#) is not connected, the `Rte_Rips_Feedback` API shall return `RTE_E_UNCONNECTED`.] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03113]{DRAFT} [If acknowledgement is enabled, and the resource created according to [\[SWS_SwCluC_03103\]](#) is connected, the `Rte_Rips_Feedback` API shall return `RTE_E_TRANSMIT_ACK`.] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

7.3.3.3 Reception

[SWS_SwCluC_03120]{DRAFT} [The `Cross Cluster Communication` of the `Software Cluster Connection` shall provide one default data instance, for each assigned `Data Communication Graph`

- where the referenced `VariableDataPrototype` is owned by a delegation `RPortPrototype` at the `CompositionSwComponentType` at the `CompositionSwComponentType` of the `rootSoftwareComposition` of the `Ecu Extract` AND

- where implicit communication applies to that [VariableDataPrototype](#) OR
- where explicit unqueued communication applies to that [VariableDataPrototype](#)

]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03133]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall provide a queue data instance (where the queue size corresponds to the specified queue length), for each assigned [Data Communication Graph](#)

- where the referenced [VariableDataPrototype](#) is owned by a delegation [RPortPrototype](#) at the [CompositionSwComponentType](#) at the [CompositionSwComponentType](#) of the [rootSoftwareComposition](#) of the [Ecu Extract](#) AND
- where explicit queued communication applies to that [VariableDataPrototype](#)

]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03134]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall determine the queue size according to the [queueLength](#) attribute of the [QueuedReceiverComSpec](#)]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03135]{DRAFT} [When [\[SWS_SwCluC_03133\]](#) applies, the Cross Cluster Communication of the [Software Cluster Connection](#) shall provide a data instance, representing the number of possible entries in the queue.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03122]{DRAFT} [If the parameter [SwCluCXccSwCluCXccDefaultDataHandling](#) is set to [DEFAULTS_AS_CONSTANTS](#), the default data instance created according to [\[SWS_SwCluC_03120\]](#) shall be mapped to a [CONST](#) memory section, according to document [\[8\]](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03121]{DRAFT} [If the parameter [SwCluCXccSwCluCXccDefaultDataHandling](#) is set to [DEFAULTS_AS_CALPRMS](#), the default data instance created according to [\[SWS_SwCluC_03120\]](#) shall be mapped to a [CALPRM](#) memory section, according to document [\[8\]](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

Please note as well that section [7.3.2.4](#) is relevant for calibration parameters instantiated by [Software Cluster Connection](#).

[SWS_SwCluC_03123]{DRAFT} [The data instances created according to [\[SWS_SwCluC_03120\]](#), [\[SWS_SwCluC_03133\]](#) and [\[SWS_SwCluC_03135\]](#) shall be mapped to a [VAR](#) memory section, according to document [\[8\]](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03124]{DRAFT} [The default data instance created according to [\[SWS_SwCluC_03120\]](#) shall be initialized according to the `NonqueuedReceiverComSpec.initValue` in the `RPortPrototype` at the `CompositionSwComponentType` of the `rootSoftwareComposition` of the Ecu Extract.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03126]{DRAFT} [The Cross Cluster Communication of the `Software Cluster Connection` shall use the default data instance created according to [\[SWS_SwCluC_03120\]](#) as default handle in the corresponding `Resource Entry` in the `Binary Manifest`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03136]{DRAFT} [The Cross Cluster Communication of the `Software Cluster Connection` shall make the data instance created according to [\[SWS_SwCluC_03133\]](#) accessible from other `Software Clusters`, via a `Resource Entry` in the `Binary Manifest`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03130]{DRAFT} [When explicit unqueued communication via `dataReceivePointByArguments` applies, and the `Resource Entry` created according to [\[SWS_SwCluC_03126\]](#) is not connected, the `Rte_Rips_Read` shall copy the value of default data instance created according to [\[SWS_SwCluC_03120\]](#) to the location of the OUT parameter `<data>`, and shall return `RTE_E_UNCONNECTED`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03131]{DRAFT} [When explicit unqueued communication via `dataReceivePointByValues` applies, and the `Resource Entry` created according to [\[SWS_SwCluC_03126\]](#) is not connected, the `Rte_Rips_DRead` shall return the value of default data instance created according to [\[SWS_SwCluC_03120\]](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03132]{DRAFT} [When explicit unqueued communication via `dataReceivePointByArguments` applies, and the `Resource Entry` created according to [\[SWS_SwCluC_03126\]](#) is connected, the `Rte_Rips_Read` shall copy the value of the corresponding data instance of the provider to the location of the OUT parameter `<data>`, and shall return `RTE_E_OK`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03137]{DRAFT} [When explicit queued communication applies, and the resource created according to [\[SWS_SwCluC_03136\]](#) is not connected, the `Rte_Rips_Read` shall discard the read access, and return `RTE_E_UNCONNECTED`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03138]{DRAFT} [When explicit queued communication applies, and the `Resource Entry` created according to [\[SWS_SwCluC_03136\]](#) is connected, but no entry is available in the queue, the `Rte_Rips_Read` shall discard the read access, and return `RTE_E_NO_DATA`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03139]{DRAFT} [When explicit queued communication applies, and the `Resource Entry` created according to [\[SWS_SwCluC_03136\]](#) is connected, the `Rte_Rips_Read` shall copy the value of the first available entry of the corresponding

data instance to the location of the OUT parameter <data>, and return RTE_E_OK.]
(SRS_SwCluC_00100, SRS_SwCluC_00101)

[SWS_SwCluC_03140]{DRAFT} [When [SWS_SwCluC_03139] applies, the `Rte_Rips_Read` shall dequeue the data from the corresponding queue.](SRS_SwCluC_00100, SRS_SwCluC_00101)

[SWS_SwCluC_03143]{DRAFT} [If explicit unqueued communication applies, and the `Resource Entry` in the `Binary Manifest` created according to [SWS_SwCluC_03126] is not connected, the `Rte_Rips_DataIsUpdated` shall return FALSE.](SRS_SwCluC_00100, SRS_SwCluC_00101)

[SWS_SwCluC_03144]{DRAFT} [If explicit unqueued communication applies, and the `Resource Entry` in the `Binary Manifest` created according to [SWS_SwCluC_03126] is connected, and the sender has updated the data since the previous execution of the corresponding `Rte_Rips_Read` service, the `Rte_Rips_DataIsUpdated` shall return TRUE

](SRS_SwCluC_00100, SRS_SwCluC_00101)

[SWS_SwCluC_03081] [The `Rte_Rips_DataIsUpdated` shall return TRUE if

- explicit unqueued communication applies

AND

- the `Resource Entry` in the `Binary Manifest` created according to [SWS_SwCluC_03126] is connected

AND

- the occurrence of a send operation was indicated after previous calls of `Rte_Read` or `Rte_DRead` related to this reading software component.

](SRS_SwCluC_00100, SRS_SwCluC_00101)

[SWS_SwCluC_03082] [The `Rte_Rips_DataIsUpdated_EventActivation` shall return TRUE if

- the `Resource Entry` in the `Binary Manifest` created according to [SWS_SwCluC_03126] or [SWS_SwCluC_03136] is connected

AND

- the occurrence of a send operation was indicated after previous calls of this instance of the `Rte_Rips_DataIsUpdated_EventActivation`

OR

- a enqueue operation occurred after previous calls of this instance of the `Rte_Rips_DataIsUpdated_EventActivation`

](SRS_SwCluC_00100, SRS_SwCluC_00101)

7.3.4 Client Server Communication

7.3.4.1 General

[SWS_SwCluC_03145]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall ensure that the result of a [ClientServerOperation](#) is dispatched to the correct client, if more than one client invokes the [ClientServerOperation](#)].([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03146]{DRAFT} [The Cross Cluster Communication shall support multiple Clients invoking the same [ClientServerOperation](#) on a server ('N:1' Communication where $N \geq 1$)].([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

7.3.4.2 Timeout

[SWS_SwCluC_03147]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall ensure that timeout monitoring is performed for client-server communication.].([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

7.3.4.3 Buffering

[SWS_SwCluC_03150]{DRAFT} [The Cross Cluster Communication on the client side of the [Software Cluster Connection](#) shall provide a request queue (with queue length 1) for each assigned [Client Server Communication Graph](#), where the referenced [ClientServerOperation](#) is owned by a delegation [RPort-Prototype](#) at the [CompositionSwComponentType](#) of the [rootSoftwareComposition](#) of the [Ecu Extract](#).

Note: The structure of the queue is implementation specific, but the [Cross Cluster Communication](#) has to store at least the IN parameters and IN/OUT parameters, the transaction handle and the status code of the communication.].([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03151]{DRAFT} [The Cross Cluster Communication on the client side of the [Software Cluster Connection](#) shall provide a response queue (with queue length 1), for each assigned [Client Server Communication Graph](#), where the referenced [ClientServerOperation](#) is owned by a delegation [RPort-Prototype](#) at the [CompositionSwComponentType](#) of the [rootSoftwareComposition](#) of the [Ecu Extract](#).

Note: The structure of the queue is implementation specific, but the [Cross Cluster Communication](#) has to store at least the IN/OUT and OUT parameters, the transaction handle and the status code of the communication.].([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03152]{DRAFT} [The Cross Cluster Communication on the server side of the [Software Cluster Connection](#) shall provide a request queue with the specified queue length (See [\[SWS_SwCluC_03154\]](#)), for each assigned [Client Server Communication Graph](#), where the referenced [ClientServerOperation](#) is owned by a delegation [PPortPrototype](#) at the [Composition-SwComponentType](#) of the [rootSoftwareComposition](#) of the Ecu Extract.

Note: The structure of the queue is implementation specific, but the Cross Cluster Communication has to store at least the IN/OUT and OUT parameters, the transaction handle and the status code of the communication.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03153]{DRAFT} [The Cross Cluster Communication on the server side of the [Software Cluster Connection](#) shall provide N response queues (where N corresponds to the specified queue length), for each assigned [Client Server Communication Graph](#), where the referenced [ClientServerOperation](#) is owned by a delegation [PPortPrototype](#) at the [Composition-SwComponentType](#) of the [rootSoftwareComposition](#) of the Ecu Extract.

Note: The structure of the queue is implementation specific, but the Cross Cluster Communication has to store at least the IN/OUT and OUT parameters, the transaction handle and the status code of the communication.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03154]{DRAFT} [The Cross Cluster Communication on the server side of the [Software Cluster Connection](#) shall determine the queue length according to the following priority rules (highest priority first):

1. value of the ECU-C parameter [RteServerQueueLength](#)
2. value of the [queueLength](#) attribute of the [ServerComSpec](#)

]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03155]{DRAFT} [The Cross Cluster Communication on the server side of the [Software Cluster Connection](#) shall handle the requests in a first-in-first-out queue.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03156]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall make the request queue instance created according to [\[SWS_SwCluC_03150\]](#) accessible for the [Software Cluster](#) implementing the server, via a [Resource Entry](#) in the [Binary Manifest](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03157]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall make the response queue instance created according to [\[SWS_SwCluC_03151\]](#) accessible to the [Software Cluster](#), via a [Resource Entry](#) in the [Binary Manifest](#).]([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03158]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall make each request queue instance created according to [\[SWS_SwCluC_03152\]](#) accessible to the [Software Cluster](#), via a [Resource Entry](#) in the [Binary Manifest](#).] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

[SWS_SwCluC_03159]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall make each response queue instance created according to [\[SWS_SwCluC_03153\]](#) accessible to the [Software Clusters](#), via a [Resource Entry](#) in the [Binary Manifest](#).] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00101](#))

7.3.4.4 Response to Request Mapping

The Cross Cluster Communication is responsible to map a response to the corresponding request. The problem of request to response mapping is split into:

- Mapping of a response to the correct client in the [Software Cluster](#).
- Mapping of a response to the correct request within one client in the [Software Cluster](#).

The general approach for request response mapping is to use transaction handles.

[SWS_SwCluC_03160]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall use a transaction handle that contains three parts, of unsigned integer type:

- Software Cluster Identifier
- Client Identifier
- Client Sequence Counter

] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03161]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall use the transaction handle for the identification of client server transactions, communicated between the clusters.] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03162]{DRAFT} [The Cross Cluster Communication on the server side of the [Software Cluster Connection](#) shall return the transaction handle of the request, without modification, together with the response.] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03163]{DRAFT} [The Cross Cluster Communication of the [Software Cluster Connection](#) shall allow only one request per client and server operation at any time.] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

7.3.4.5 Client Side

[SWS_SwCluC_03164]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is not connected, the `Rte_Rips_Invoke` shall return `RTE_E_UNCONNECTED` immediately.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03165]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is connected, the `Rte_Rips_Invoke` shall copy the IN and IN/OUT parameters, inform the server via a status flag, and return `RTE_E_OK`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03166]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is connected, the `Rte_Rips_Invoke` shall return `RTE_E_LIMIT`, until the server's result has been successfully passed to the client, and no timeout occurred. The IN and IN/OUT parameters shall not be modified by `Rte_Rips_Invoke` .]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03167]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is not connected, the `Rte_Rips_ReturnResult` shall return `RTE_E_UNCONNECTED` immediately.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03168]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is connected, the `Rte_Rips_ReturnResult` shall return `RTE_E_NO_DATA`, until the server's result has been successfully passed to the client and no timeout occurred.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03169]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is connected, and the server's result was not available within the specified timeout, the `Rte_Rips_ReturnResult` shall return `RTE_E_TIMEOUT`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03170]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is connected, and an [AsynchronousServerCallReturnsEvent](#) exists, the `Rte_Rips_InvocationHandler` shall poll the corresponding response queue status to determine whether or not the server's result is available.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03171]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is connected, and an [AsynchronousServerCallReturnsEvent](#) exists, the `Rte_Rips_InvocationHandler` shall call the corresponding ACSR `Runnable`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

7.3.4.6 Server Side

[SWS_SwCluC_03173]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the request queue (See [\[SWS_SwCluC_03158\]](#)) is connected, the `Rte_Rips_InvocationHandler` shall poll the corresponding request queue status to determine whether or not there is a pending request.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03174]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the request queue (See [\[SWS_SwCluC_03158\]](#)) is connected, the `Rte_Rips_InvocationHandler` shall invoke the server with IN, IN/OUT and OUT parameters of request resp. response queue.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03175]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the response queue (See [\[SWS_SwCluC_03157\]](#)) is connected, the `Rte_Rips_InvocationHandler` shall call the corresponding server `Runnable`.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03176]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the request queue (See [\[SWS_SwCluC_03158\]](#)) is connected, the `Rte_Rips_InvocationHandler` shall dequeue the server queue after successful invocation of the server queue.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

[SWS_SwCluC_03177]{DRAFT} [If the corresponding resource in the [Binary Manifest](#) representing the request queue (See [\[SWS_SwCluC_03158\]](#)) is connected, the `Rte_Rips_InvocationHandler` shall set the status of the response queue (See [\[SWS_SwCluC_03159\]](#)) accordingly, to inform the client that the request has been processed.]([SRS_SwCluC_00100](#), [SRS_SwCluC_00103](#))

7.3.5 Modes Communication

7.3.5.1 General principles

The mode switch communication in AUTOSAR is a mixture between a data (the mode value) and an event semantic (the transition between modes). It also has some potential impact on the execution of [ExecutableEntities](#), via mode switch events and mode disabling dependencies. Additionally, the transitions between modes are following a strict sequence, as described in document [\[9\]](#).

Introducing this communication pattern to a clustered software architecture requires a technical solution that balances between the local view of software components inside a [Software Cluster](#), and the ECU wide behavior cross several [Software Clusters](#).

For the Software Cluster Communication, the following principles apply:

- The mode providing `Software Cluster` owns the leading mode machine instance, including the mode queue.
- The `Host Software Cluster` provides the mode switch tasks, which are used to execute the mode switch on each partition, where the mode is capable to execute runnables
- The `Host Software Cluster` coordinates the transfer of the mode switch notifications between the `Software Clusters`.
- The `Host Software Cluster` coordinates the conjunction of the `Software Cluster` individual completions of the mode switches, and notifies the mode providing `Software Cluster`
- Nevertheless, the mode switch itself is locally executed in each RTE in a `Software Cluster`. This has the consequence that the order between `on-exit ExecutableEntitys`, `on-transition ExecutableEntitys`, and `on-entry ExecutableEntitys` is only preserved locally inside a `Software Cluster`.
- The interfaces towards the RTE are only called in well-defined OS task contexts.

A sender-receiver communication can be implemented solely between `Applicative Software Clusters`, without impact to the `Host Software Cluster`. In contrast, the adding of a new mode switch communication between `Applicative Software Clusters`, additionally impacts the `Host Software Cluster`.

7.3.5.2 Software Cluster providing a mode

The `Software Cluster` providing the mode owns the leading mode machine instance, including its mode queue. Hence, the information about the next mode, to which the mode machine currently switches, is provided by this `Software Cluster`. In this specification, this mode value is called 'current on-transition value'.

The figure 7.3 illustrates the principle sequence, when a mode switch is initiated by an `Applicative Software Cluster`.

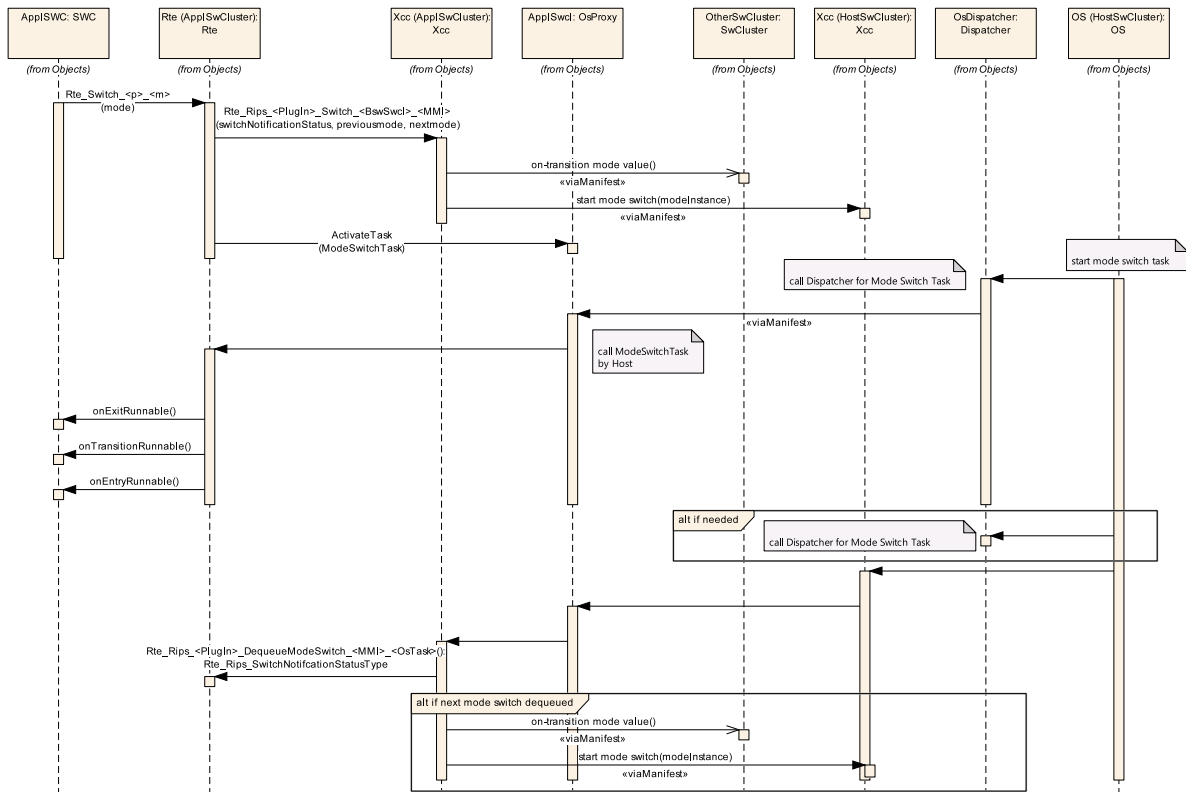


Figure 7.3: Execution of a mode switch provided by an Applicative Software Cluster

- The `mode manager` software component calls the `Rte_Switch` API to do the mode switch notification
- The `Cross Cluster Communication` gets notified via the call of the `Rte_Rips_Switch` Service.
- If the mode queue is empty, this `Rte_Rips_Switch` can be used to determine the first on-transition value.
- The `Cross Cluster Communication` in the `Applicative Software Cluster` informs the `Cross Cluster Communication` in the `Host Software Cluster` that a mode switch starts. This notification is a `Cross Cluster Communication` internal interface, which is not standardized.
- After the start of the mode switch on the host, the according mode switch tasks are scheduled (potentially on multiple partitions). This in turn, via `Dispatchers`, schedules cluster local proxy mode switch tasks.
- Potentially, other proxy mode switch tasks are scheduled before and after. Hence, the RTE cannot dequeue the next mode already in the context of its mode switch tasks.
- After the `Cross Cluster Communication` in the `Host Software Cluster` has determined that a mode switch task has run to end, it uses `Cross`

Cluster Communication to notify the RTE in the **Applicative Software Cluster** to dequeue the next mode.

The figure 7.4 illustrates the principle sequence, when a mode switch is initiated by a **Host Software Cluster**.

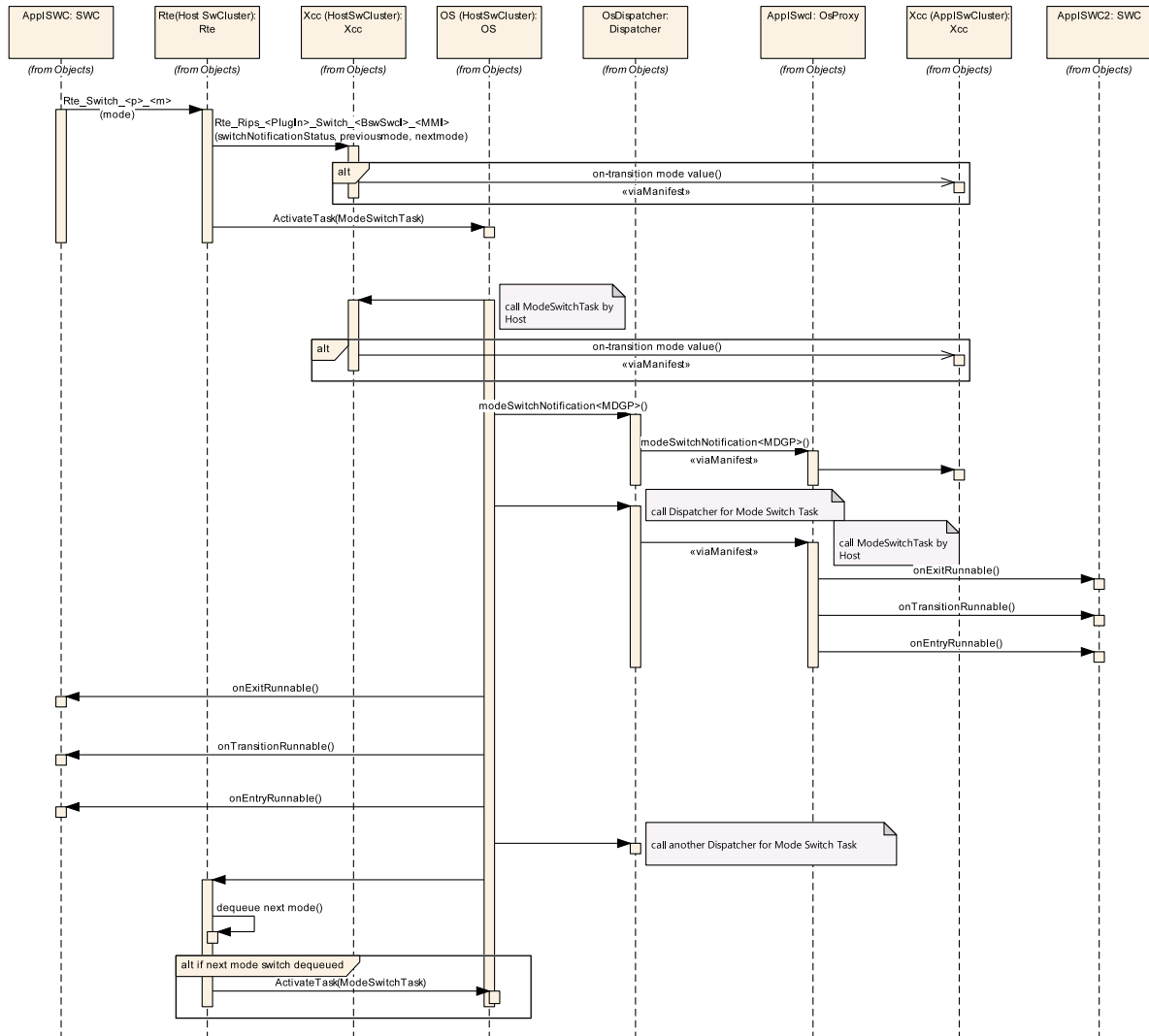


Figure 7.4: Execution of a mode switch provided by an Host Software Cluster

- The **mode manager** software component calls the `Rte_Switch` API, to do the mode switch notification
- The **Cross Cluster Communication** gets notified via the call of the `Rte_Rips_Switch` Service.
- If the mode queue is empty, this `Rte_Rips_Switch` can be used to determine the first on-transition value. Alternatively, this is also possible in the mode switch task(s) before first `Dispatchers` are scheduled.

- After the start of the mode switch on the host, the according mode switch tasks are scheduled (potentially on multiple partitions). This in turn, via `Dispatchers`, schedules cluster local proxy mode switch tasks.
- In contrast to the scenario in figure 7.3, the RTE can dequeue the next mode already in the context of its mode switch tasks.

These two principle sequences lead to the following requirements on the `Cross Cluster Communication`, in case a `Software Cluster` provides a mode.

[SWS_SwCluC_03022]{DRAFT} [The `Cross Cluster Communication` shall signal the current on-transition value to other `Software Clusters`, before the mode switch according to [SWS_Rte_02665] starts.](*SRS_SwCluC_00104*)

Note: This functionality can be implemented inside the `Rte_Rips_Switch` service, in case the mode queue was empty. Otherwise, this can be done shortly before the last mode switch task terminates, or when the first mode switch task starts already for the next transition. The interface to signal the current on-transition is a `Cross Cluster Communication` internal interface, which is not standardized.

[SWS_SwCluC_03057]{DRAFT} [The `Cross Cluster Communication` shall add at the `Software Cluster` providing the mode a `Provided Resource Entry`, and at any `Software Cluster` requiring the trigger an according `Require Resource Entry` in the `Binary Manifest`.](*SRS_SwCluC_00104*)

[SWS_SwCluC_03057] ensures that a missing `mode manager` in a clustered ECU gets detected, even if the implementation may rely additionally on the according `Dispatchers`.

[SWS_SwCluC_03023]{DRAFT} [The `Cross Cluster Communication` of the `Host Software Cluster` shall implement a `mode manager` for each mode which needs to be provided by an `Applicative Software Cluster`.](*SRS_SwCluC_00104*)

With this `mode manager`, it is possible to schedule the according `Dispatchers` for mode switch notification, and `Dispatchers` for `Software Cluster` local proxy mode switch tasks.

To ease the realization of [SWS_SwCluC_03023], all modes provided by the `Applicative Software Clusters`, shall be consumed by the `Host Software Cluster`.

[SWS_SwCluC_CONSTR_03032]{DRAFT} [For each mode provided by an `Applicative Software Cluster`, the `Ecu Extract` of the `Host Software Cluster` shall own a required `mode switch port` at the `CompositionSwComponentType` of the `rootSoftwareComposition`.](*SRS_SwCluC_00104*)

[SWS_SwCluC_03024]{DRAFT} [The `Cross Cluster Communication` of the `Host Software Cluster` shall enqueue the current on-transition values signaled by `Applicative Software Cluster`, via the related `mode manager`. ([SWS_SwCluC_03023])](*SRS_SwCluC_00104*)

Note: In case the [Host Software Cluster](#) provides the mode, the RTE implements already the leading mode machine instance.

[SWS_SwCluC_03025]{DRAFT} [The Cross Cluster Communication of the [Host Software Cluster](#) shall notify the [Applicative Software Cluster](#), when a mode switch tasks starts.]([SRS_SwCluC_00104](#))

Note: This notification is used by the [Applicative Software Cluster](#) to schedule the [mode switch notification](#) runnable, which enqueues the mode switch notification in the cluster local RTE.

[SWS_SwCluC_03026]{DRAFT} [The Cross Cluster Communication of the [Host Software Cluster](#) shall notify the Cross Cluster Communication of the [Applicative Software Cluster](#), when all mode switch tasks run to end.]([SRS_SwCluC_00104](#))

[SWS_SwCluC_03027]{DRAFT} [In case the [Applicative Software Cluster](#) provides the mode, the Cross Cluster Communication of the [Host Software Cluster](#) shall notify the Cross Cluster Communication of the [Applicative Software Cluster](#), when mode switch tasks run to end.]([SRS_SwCluC_00104](#))

[SWS_SwCluC_03028]{DRAFT} [In case the [Applicative Software Cluster](#) provides the mode, the Cross Cluster Communication of the [Applicative Software Cluster](#) notifies the RTE via [Rte_Rips_DequeueModeSwitch](#), when a mode switch task known by this RTE runs to end. The Cross Cluster Communication has to guarantee that the last call of [Rte_Rips_DequeueModeSwitch](#) is not done before the last on-entry [ExecutableEntity](#) in the whole clustered system terminated.]([SRS_SwCluC_00104](#))

Please note [[SWS_Rte_70123](#)], which guarantees a certain execution context for the RTE.

[SWS_SwCluC_03029]{DRAFT} [The Cross Cluster Communication shall provide a Complex Driver Software Component on each [EcucPartition](#), where mode switch task are configured. This component is later called [mode proxy component](#).]([SRS_SwCluC_00104](#))

[SWS_SwCluC_03030]{DRAFT} [The Cross Cluster Communication shall provide [RunnableEntity](#)s to detect the start and end of any mode switch task, which is related to mode communication cross [Software Clusters](#).]([SRS_SwCluC_00104](#))

Please note: the [RunnableEntity](#)s [[SWS_SwCluC_03030](#)] may require additional [ModeAccessPoints](#), as well as suitable [RTEEvents](#). It is possible to use [OsTaskExecutionEvent](#) and a small runtime logic, to determine the ongoing mode transition, or a set of [SwcModeSwitchEvents](#), which activate the [RunnableEntity](#) on any transition of a specific mode machine instance.

7.3.5.3 Host Software Cluster requiring a mode

As explained in section 7.3.5.2, the `Host Software Cluster` is already aware about existing modes communicated cross `Software Clusters`. Therefore, it technically receives already the modes provided by `Applicative Software Clusters`. In case the `Host Software Cluster` owns software components, which require the the mode as well, it shall connect them to its already existing mode manager.

[SWS_SwCluC_03031]{DRAFT} [The `Cross Cluster Communication` shall provide the `AssemblySwConnectors` between the provided `mode switch ports` of the `mode proxy component` and the required `mode switch ports` at the software components requiring the mode, if the mode is required by the `Host Software Cluster`.] (*SRS_SwCluC_00104*)

7.3.5.4 Applicative Software Cluster requiring a mode

For each required mode, the `Applicative Software Cluster` defines `OsTask(s)` plus `dispatcher(s)`, for the `mode switch notification`.

When the mode switch gets notified by the `Host Software Cluster`, the `Cross Cluster Communication` writes the actual on-transition mode value to the RTE of the `Applicative Software Cluster` with the regular `Rte_Switch` API related to a `mode switch port`.

Note: The RTE in the `Applicative Software Cluster` shall still a call `ActivateTask(s)` to trigger the mode switch task(s). This call can still be used by the `Host Software Cluster` to trigger the mode switch task execution in the `Host Software Cluster`, if needed.

For this purpose, the `Cross Cluster Communication` defines a `RunnableEntity` with an `OsTaskExecutionEvent` for each mode on each partition where a `mode switch notification` is configured. When the mode switch is executed, the `Host Software Cluster` schedules the mode switch task(s), which in turn schedule the 'proxy' mode switch task(s) in the `Applicative Software Cluster`

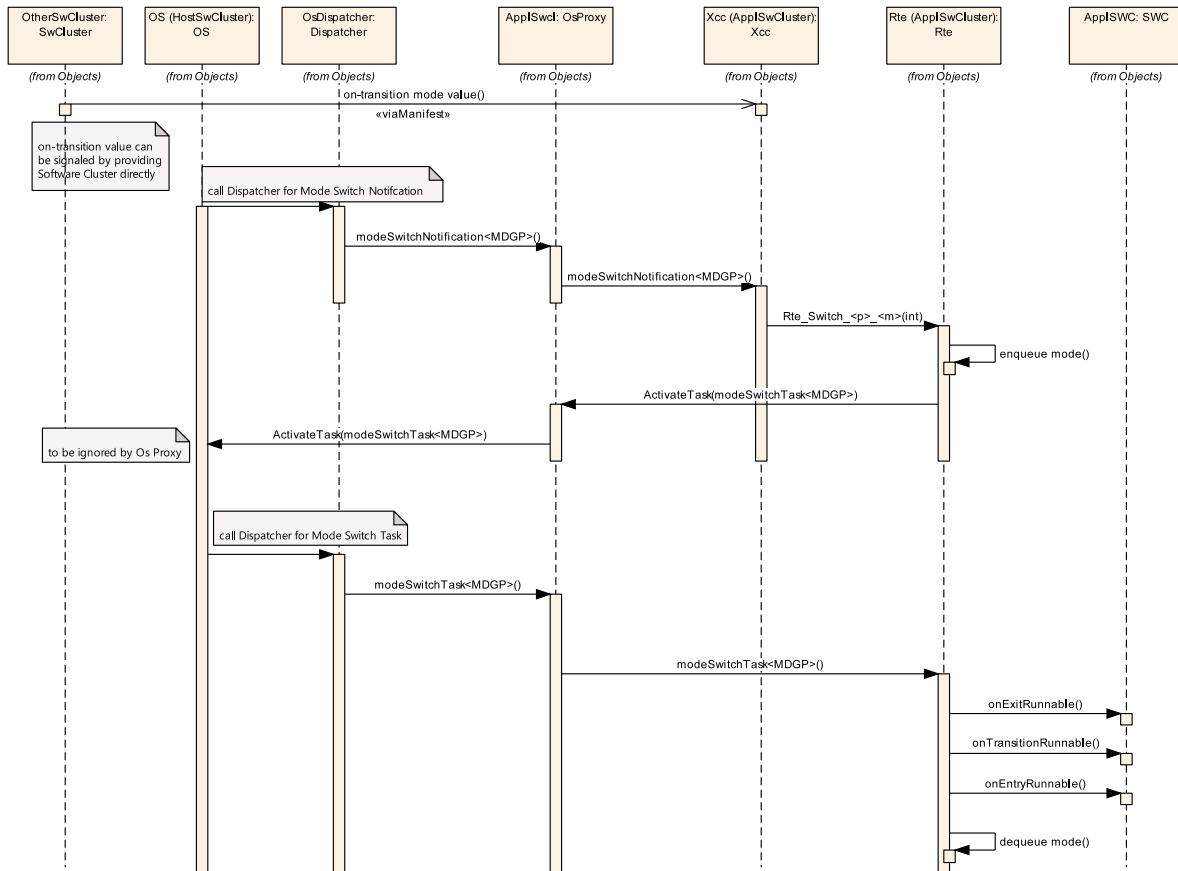


Figure 7.5: Execution of a mode switch in an Applicative Software Cluster

[SWS_SwCluC_03015]{DRAFT} [The Cross Cluster Communication shall provide a Complex Driver Software Component on each *EcucPartition* where a *mode switch port* needs to be provided. This component is later called *mode proxy component*.] (*SRS_SwCluC_00104*)

[SWS_SwCluC_03016]{DRAFT} [The Cross Cluster Communication shall provide a provided mode switch port on each *mode proxy component* where a *mode switch port* needs to be provided.] (*SRS_SwCluC_00104*)

[SWS_SwCluC_03017]{DRAFT} [The Cross Cluster Communication shall provide the *AssemblySwConnectors* between the provided *mode switch ports* and the required *mode switch ports* at the software components requiring the mode.] (*SRS_SwCluC_00104*)

[SWS_SwCluC_03018]{DRAFT} [The Cross Cluster Communication shall provide a *mode switch notification runnable* with an *OsTaskExecution-Event*, for each mode on each partition where a *mode switch notification* is configured. The *RunnableEntity* shall define a *ModeSwitchPoint*.] (*SRS_SwCluC_00104*)

[SWS_SwCluC_03019]{DRAFT} [When the mode switch notification runnable is executed, it shall write the actual on-transition value via `Rte_Switch` to the RTE.]([SRS_SwCluC_00104](#))

[SWS_SwCluC_CONSTR_03020]{DRAFT} [For each `EcucPartition` on which the mode switch notification is configured for a `ModeDeclarationGroupPrototype` required by an `Applicative Software Cluster`, a `OsTask` with a related `SwCluCOsProxyOsTask` and `SwCluCOsProxyOsTaskDispatcher` shall be configured.]([SRS_SwCluC_00104](#))

[SWS_SwCluC_CONSTR_03021]{DRAFT} [For each required `ModeDeclarationGroupPrototype` and `EcucPartition` on which the mode switch notification is configured, a `OsTask` with a related `SwCluCOsProxyOsTask` and `SwCluCOsProxyOsTaskDispatcher` shall be configured.]([SRS_SwCluC_00104](#))

7.3.5.5 Initialization

The approach described in section 7.3.5.2 and 7.3.5.4 leads to local mode machine instances in each `Software Cluster` providing or requiring the mode.

But this also implies that the initialization of the mode machine instances is executed time shifted, without explicit synchronization by the `Cross Cluster Communication`.

When the clustered system starts, the following procedure needs to be preserved!

- Initialization of the `Cross Cluster Communications`
- Execute all `Rte_Init_<InitContainer>` functions in `Applicative Software Clusters` and `Host Software Cluster`. This step already leads to the execution of `on-entry ExecutableEntitys`, triggered by `initialMode` and mapped to `RteInitializationRunnableBatch` container.
- Execute all `Rte_Start` in `Applicative Software Clusters`. This step executes the remaining `on-entry ExecutableEntitys` triggered by `initialMode`. In case the RTE implementation triggers the mode switch task to proceed the transition to the `initialMode`, it will happen here as well.
- Execute `Rte_Start` in the `Host Software Cluster`. This step executes the remaining `on-entry ExecutableEntitys` triggered by `initialMode` in the `Host Software Cluster`. In case the RTE implementation triggers the mode switch task to proceed the transition to the `initialMode`, the mode switch task will now be executed.

7.3.6 Trigger Communication

7.3.6.1 General principles

The trigger communication in AUTOSAR is a pure event semantic, which is used to request the execution of `triggered ExecutableEntitys`.

Introducing this communication pattern to a clustered software architecture, the focus lies on the use case to implement `trigger sources` in `Applicative Software Clusters` and in the `Host Software Cluster`. Nevertheless, the control on the execution of the `triggered ExecutableEntitys` is implemented in the `Host Software Cluster`, to guarantee an ECU wide behavior cross several `Software Clusters`.

For the `Software Cluster Communication`, the following principles apply:

- The trigger providing `Applicative Software Cluster` transfers the occurrence of the trigger to the `Host Software Cluster`.
- The `Host Software Cluster` provides the tasks, which are used to execute the `triggered ExecutableEntitys` on each partition where needed.
- If queuing of triggers is required from a dynamic perspective, this is configured and implemented in the RTE of the `Host Software Cluster`
- Nevertheless, the triggering itself is locally executed in each RTE inside a `Software Cluster`. This has the consequence that the order between `triggered ExecutableEntitys` is only preserved locally inside a single proxy OS Task of an `Applicative Software Cluster`.
- The interfaces towards the RTE are only called in well-defined OS task contexts.

Adding of a new trigger communication between `Applicative Software Clusters`, additionally impacts the `Host Software Cluster`.

7.3.6.2 Software Cluster providing a mode

The figure 7.6 illustrates the principle sequence, when a trigger is raised by an `Applicative Software Cluster`.

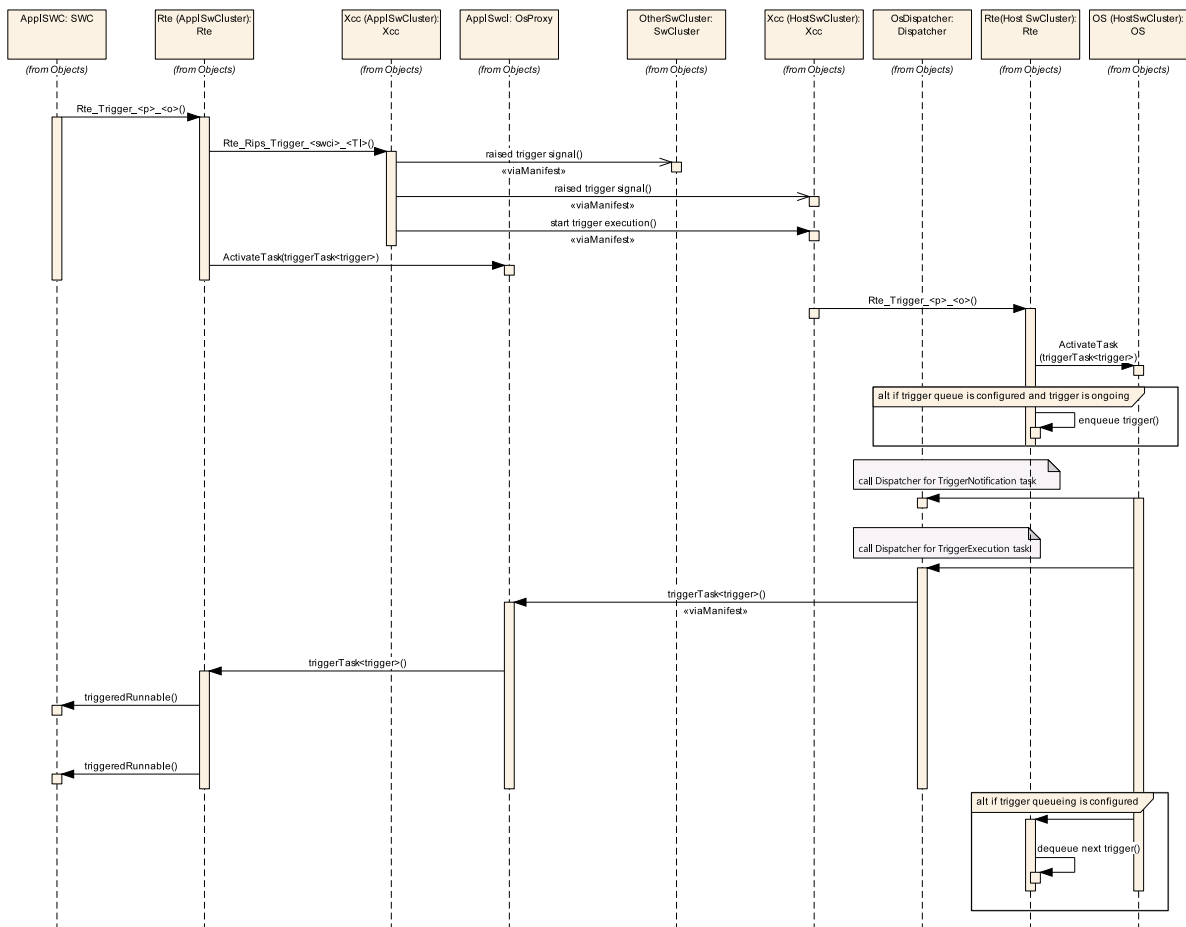


Figure 7.6: Execution of a trigger provided by an Applicative Software Cluster

- The `trigger source` calls the `Rte_Trigger` API, to raise the trigger.
- The Cross Cluster Communication gets notified, via the call of the `Rte_Rips_Trigger` Service.
- The Cross Cluster Communication in the `Applicative Software Cluster` informs the Cross Cluster Communication in the `Host Software Cluster` that a trigger was raised. This notification is a Cross Cluster Communication internal interface, which is not standardized.
- In the `Host Software Cluster`, the Cross Cluster Communication raises the trigger in the RTE, via `Rte_Trigger` API. In case of a configured trigger queue, this would enqueue the trigger.
- After the start of the trigger execution on the `Host Software Cluster`, the according tasks are scheduled (potentially on multiple partitions). This in turn schedules, via Dispatchers, the `Applicative Software Cluster` local proxy tasks.
- In case of a configured trigger queue, the dequeue operation is executed after the last `triggered ExecutableEntity` terminated.

The figure 7.7 illustrates the principle sequence, when a trigger is raised by a **Host Software Cluster**.

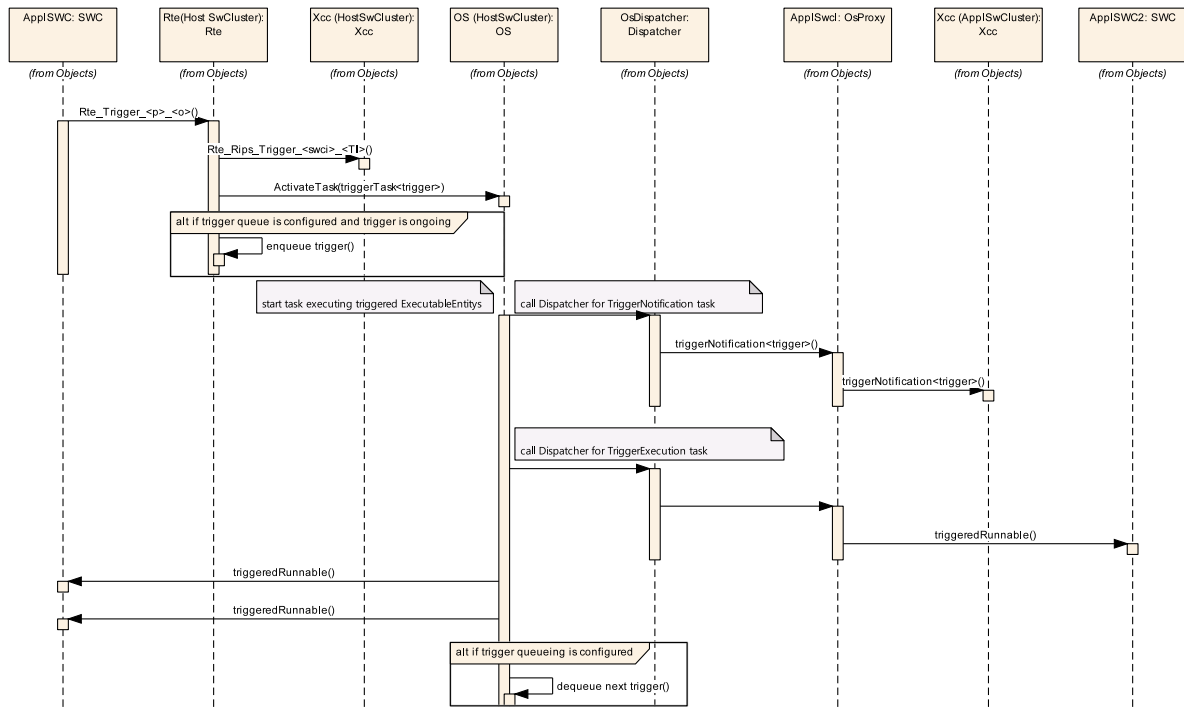


Figure 7.7: Execution of a trigger provided by an Host Software Cluster

- The **trigger source** calls the `Rte_Trigger` API, to raise the trigger.
- The Cross Cluster Communication gets notified, via the call of the `Rte_Rips_Trigger` Service.
- The RTE activates the tasks used for execution of **triggered ExecutableEntities**. If a queue is configured, and a trigger execution is already ongoing, the RTE in the **Host Software Cluster** enqueues the trigger.
- After the start of the trigger execution on the **Host Software Cluster**, the according tasks are scheduled (potentially on multiple partitions). This in turn, schedules, via Dispatchers, the **Applicative Software Cluster** local proxy tasks.
- In case of a configured trigger queue, the dequeue operation is executed after the last **triggered ExecutableEntities** terminated.

These two principle sequences lead to the following requirements on the Cross Cluster Communication, in case a **Software Cluster** provides a mode.

[SWS_SwCluC_03055]{DRAFT} [The Cross Cluster Communication shall signal to other **Software Clusters** that a trigger was raised.] (*SRS_SwCluC_00105*)

Note: This functionality can be implemented inside the `Rte_Rips_Trigger` service. Function wise, it is up to the implementation, if the `[SWS_SwCluC_03055]` is already fulfilled via the connection to the `Host Software Cluster`.

[SWS_SwCluC_03056]{DRAFT} [The Cross Cluster Communication shall, at the `Software Cluster` providing the trigger, add a `Provided Resource Entry` and, at `Software Clusters` requiring the trigger, an according `Require Resource Entry` in the `Binary Manifest`.] (*SRS_SwCluC_00105*)

`[SWS_SwCluC_03056]` ensures that a missing `trigger source` in a clustered ECU gets detected, even if the implementation may rely on the according `Dispatchers`.

[SWS_SwCluC_03058]{DRAFT} [The Cross Cluster Communication of the `Host Software Cluster` shall implement a `trigger source` for each trigger, which needs to be provided by an `Applicative Software Cluster`.] (*SRS_SwCluC_00105*)

With this `trigger source`, it is possible to schedule the according `Dispatchers` for trigger notification, and `Dispatchers` for `Software Cluster` local trigger execution tasks.

To ease the realization of `[SWS_SwCluC_03058]`, all triggers provided by `Applicative Software Clusters`, shall be consumed by the `Host Software Cluster`.

[SWS_SwCluC_CONSTR_03059]{DRAFT} [The `Ecu Extract` of the `Host Software Cluster` shall, for each trigger provided by an `Applicative Software Cluster`, own a required `trigger port` at the `CompositionSwComponentType` of the `rootSoftwareComposition`.] (*SRS_SwCluC_00105*)

[SWS_SwCluC_03060]{DRAFT} [The Cross Cluster Communication of the `Host Software Cluster` shall raise the triggers signaled by `Applicative Software Clusters`, via the related `trigger source`. (`[SWS_SwCluC_03058]`)] (*SRS_SwCluC_00105*)

Note: In case the `Host Software Cluster` provides the trigger, the `trigger source` is already part of the `Software Components` or `BSW Modules` belonging to the `Host Software Cluster`, which are directly interacting with the `RTE` or `SchM`.

[SWS_SwCluC_03061]{DRAFT} [The Cross Cluster Communication of the `Host Software Cluster` shall notify the `Applicative Software Cluster`, when the execution of a trigger tasks starts.] (*SRS_SwCluC_00104*)

Note: This notification is used by the `Applicative Software Cluster` to schedule the `trigger notification runnable`, which raises the trigger at the cluster local `RTE`.

[SWS_SwCluC_03062]{DRAFT} [The Cross Cluster Communication shall provide a `Complex Driver Software Component` on each `EcucPartition`, where a `trigger source` according to `[SWS_SwCluC_03058]` is configured. This component is later called `trigger proxy component`.] (*SRS_SwCluC_00104*)

7.3.6.3 Host Software Cluster requiring a trigger

As explained in section 7.3.6.2, the `Host Software Cluster` is already aware about an existing trigger communicated cross `Software Clusters`. Therefore, it technically already receives the triggers raised by `Applicative Software Clusters`. In case the `Host Software Cluster` owns software components that require the the trigger as well, it shall connect them to its already existing `trigger source`, according to [SWS_SwCluC_03058].

[SWS_SwCluC_03063]{DRAFT} [The Cross Cluster Communication shall provide the `AssemblySwConnectors` between the provided `trigger ports` of the `trigger proxy component`, and the required `trigger ports` at the software components requiring the trigger, if the trigger is required by the `Host Software Cluster`.](SRS_SwCluC_00104)

7.3.6.4 Applicative Software Cluster requiring a trigger

For each required `trigger`, the `Applicative Software Cluster` defines `OsTask(s)` plus `dispatcher(s)`, for the `trigger notification`.

When the trigger gets executed by the `Host Software Cluster`, the `Cross Cluster Communication` raises `trigger` at the RTE of the `Applicative Software Cluster`, with the regular `Rte_Trigger` API related to a `trigger port`.

Note: The RTE in the `Applicative Software Cluster` shall still call `ActivateTask(s)`, to trigger the task(s) used for the execution of `triggered ExecutableEntities`. This call can still be used by the `Host Software Cluster` to trigger the task executions in the `Host Software Cluster`, if needed.

For this purpose, the `Cross Cluster Communication` defines a `RunnableEntity` with an `OsTaskExecutionEvent`, for each trigger and on each partition, a `trigger notification` is configured. When the trigger is executed, the `Host Software Cluster` schedules the task(s), which in turn schedule the 'proxy' task(s) in the `Applicative Software Cluster`.

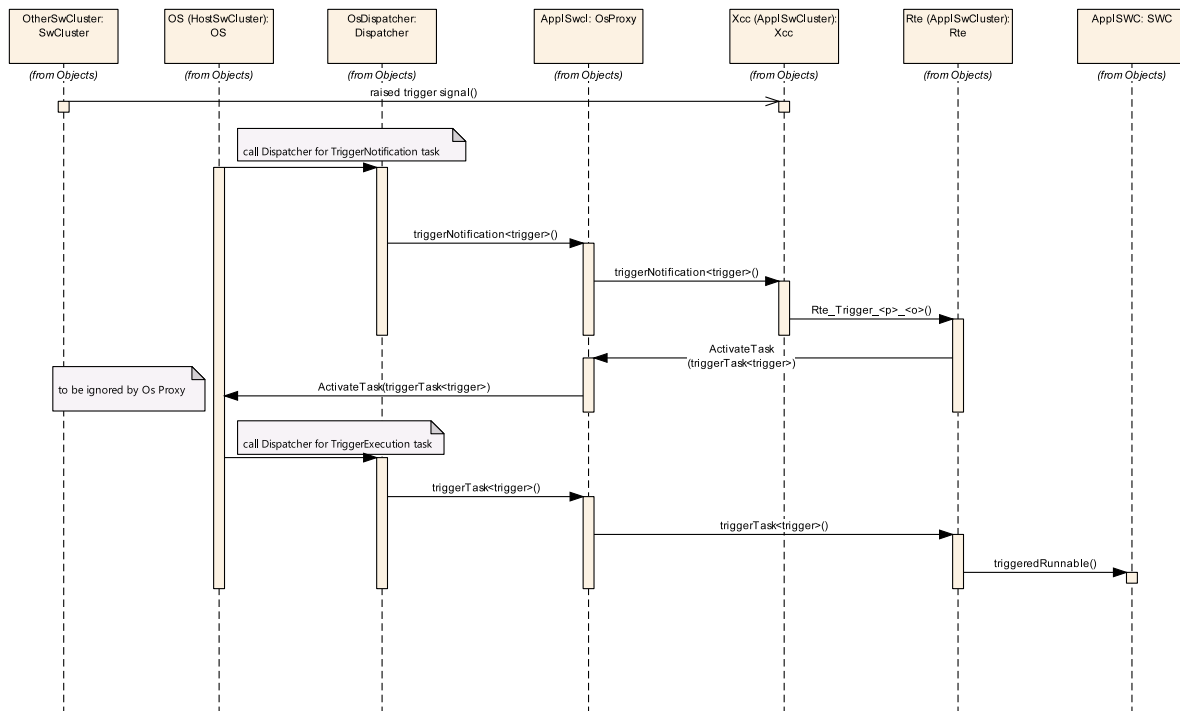


Figure 7.8: Execution of a trigger in an Applicative Software Cluster

[SWS_SwCluC_03048]{DRAFT} [The Cross Cluster Communication shall provide a Complex Driver Software Component, on each `EcucPartition` where a `trigger` port needs to be provided. This component is later called `trigger proxy` component.](SRS_SwCluC_00105)

[SWS_SwCluC_03049]{DRAFT} [The Cross Cluster Communication shall provide a provided `trigger` port, on each `trigger proxy` component.](SRS_SwCluC_00105)

[SWS_SwCluC_03050]{DRAFT} [The Cross Cluster Communication shall provide the `AssemblySwConnectors` between the provided `trigger` ports and the required `trigger` ports, at the software components requiring the trigger.](SRS_SwCluC_00105)

[SWS_SwCluC_03051]{DRAFT} [The Cross Cluster Communication shall provide a trigger notification runnable with an `OsTaskExecutionEvent`, for each trigger on each partition a trigger notification is configured. The `RunnableEntity` shall define a `ExternalTriggeringPoint`.](SRS_SwCluC_00105)

[SWS_SwCluC_03052]{DRAFT} [When the trigger notification runnable is executed, it shall raise the trigger via `Rte_Trigger` to the RTE.](SRS_SwCluC_00105)

[SWS_SwCluC_CONSTR_03053]{DRAFT} [For each `EcucPartition` on which the `trigger` notification is configured for a `trigger` required by an `Applicative Software Cluster`, a `OsTask`, with a related `SwCluCOsProxyOsTask` and `SwCluCOsProxyOsTaskDispatcher`, shall be configured.] (*SRS_SwCluC_00105*)

[SWS_SwCluC_CONSTR_03054]{DRAFT} [For each required `trigger` and `EcucPartition` on which `triggered ExecutableEntitys` needs to be executed, at least one `OsTask`, with a related `SwCluCOsProxyOsTask` and `SwCluCOsProxyOsTaskDispatcher`, shall be configured.] (*SRS_SwCluC_00105*)

7.3.7 Parameter Communication

[SWS_SwCluC_03006]{DRAFT} [The `Cross Cluster Communication` of the `Software Cluster Connection` shall provide one constant default parameter instance for each assigned `Parameter Communication Graph`, which contains `NO PPortPrototype`.] (*SRS_SwCluC_00100, SRS_SwCluC_00106*)

[SWS_SwCluC_03007]{DRAFT} [The default value instance shall be mapped to a `CALPRM` memory section, according to document [8].] (*SRS_SwCluC_00100, SRS_SwCluC_00106*)

Please note as well section 7.3.2.4, which is relevant for calibration parameters instantiated by `Software Cluster Connection`.

[SWS_SwCluC_03008]{DRAFT} [The default value instance shall be initialized according to the `ParameterRequireComSpec.initValue`, in the `RPortPrototype` at the `CompositionSwComponentType` of the `rootSoftwareComposition` of the `Ecu Extract`.] (*SRS_SwCluC_00100, SRS_SwCluC_00106*)

[SWS_SwCluC_03009]{DRAFT} [If the corresponding resource in the `Binary Manifest` is connected, the `Rte_Rips_Prm` shall return the value of the connected parameter.] (*SRS_SwCluC_00100, SRS_SwCluC_00106*)

[SWS_SwCluC_03010]{DRAFT} [If the corresponding resource in the `Binary Manifest` is NOT connected, the `Rte_Rips_Prm` shall return the value of the default parameter instance of [SWS_SwCluC_03006].] (*SRS_SwCluC_00100, SRS_SwCluC_00106*)

Note: The behavior of `Rte_Rips_Prm` function, according to [SWS_SwCluC_03009] and [SWS_SwCluC_03010], can be implemented by referencing the default parameter instance as the default handle in the `Binary Manifest`.

[SWS_SwCluC_03011]{DRAFT} [For each assigned `Parameter Communication Graph`, which contains the `PPortPrototype`, the `Cross Cluster Communication` of the `Software Cluster Connection` shall make the parameter instance, according to [SWS_Rte_80130], accessible for other `Software Clusters`, via a `Resource Entry` in the `Binary Manifest`.] (*SRS_SwCluC_00100, SRS_SwCluC_00106*)

7.3.8 Error Classification

Together with the RTE, the Cross Cluster Connection of the [Software Cluster Connection](#) implements the Run-Time Environment of the [Software Cluster](#). The usual development errors checking API parameters are already covered by the development error detection of RTE. In addition, communication infrastructure errors are not reported as `Production Errors` nor as `Extended Production Errors`. Therefore, the following sections do not define error codes, and are marked as not applicable.

7.3.8.1 Development Errors

Development errors are not applicable for the Cross Cluster Connection of the [Software Cluster Connection](#).

7.3.8.2 Runtime Errors

Runtime errors are not applicable for the Cross Cluster Communication of the [Software Cluster Connection](#).

7.3.8.3 Transient Faults

Transient Faults are not applicable for the Cross Cluster Communication of the [Software Cluster Connection](#).

7.3.8.4 Production Errors

Production Errors are not applicable for the Cross Cluster Communication of the [Software Cluster Connection](#).

7.3.8.5 Extended Production Errors

Extended Production Errors are not applicable for the Cross Cluster Communication of the [Software Cluster Connection](#).

7.4 Proxy Modules

7.4.1 Overview

Since an Applicative Software Cluster does not contain all BSW modules (in extreme case no BSW modules), those missing APIs, as well as the EcuC configuration elements, need to be substituted.

In the AUTOSAR Layered Software Architecture, the dependency between BSW modules can be generalized as follows:

A higher layer BSW module uses the APIs of a lower layer BSW module. In the opposite direction, the lower layer BSW module may call callback functions of the higher layer BSW module. In case of `AUTOSAR Interfaces`, additional Software Component Descriptions are required to enable the RTE generation. If the interface between those BSW modules is configurable, the ECU configuration provides the information about Symbolic Name values and ID values.

Of course, the AUTOSAR Layered Software Architecture also has horizontal dependencies. In this case, rotate the previous paragraph by 90 degrees.

It is important to state that the different BSW Module are directly using the C-interfaces of each other, without abstraction (as the RTE provides between SWCs). This, in turn, causes strong implementation dependencies, like fixed names of include files and C-functions.

Since the actual implementation and integration of the Classic Platform architecture is now split into Software Clusters, missing BSW Modules - those to which interfaces are existing - need to be substituted. The Figure 7.9 illustrates the proxy module approach, for a horizontal interface dependency.

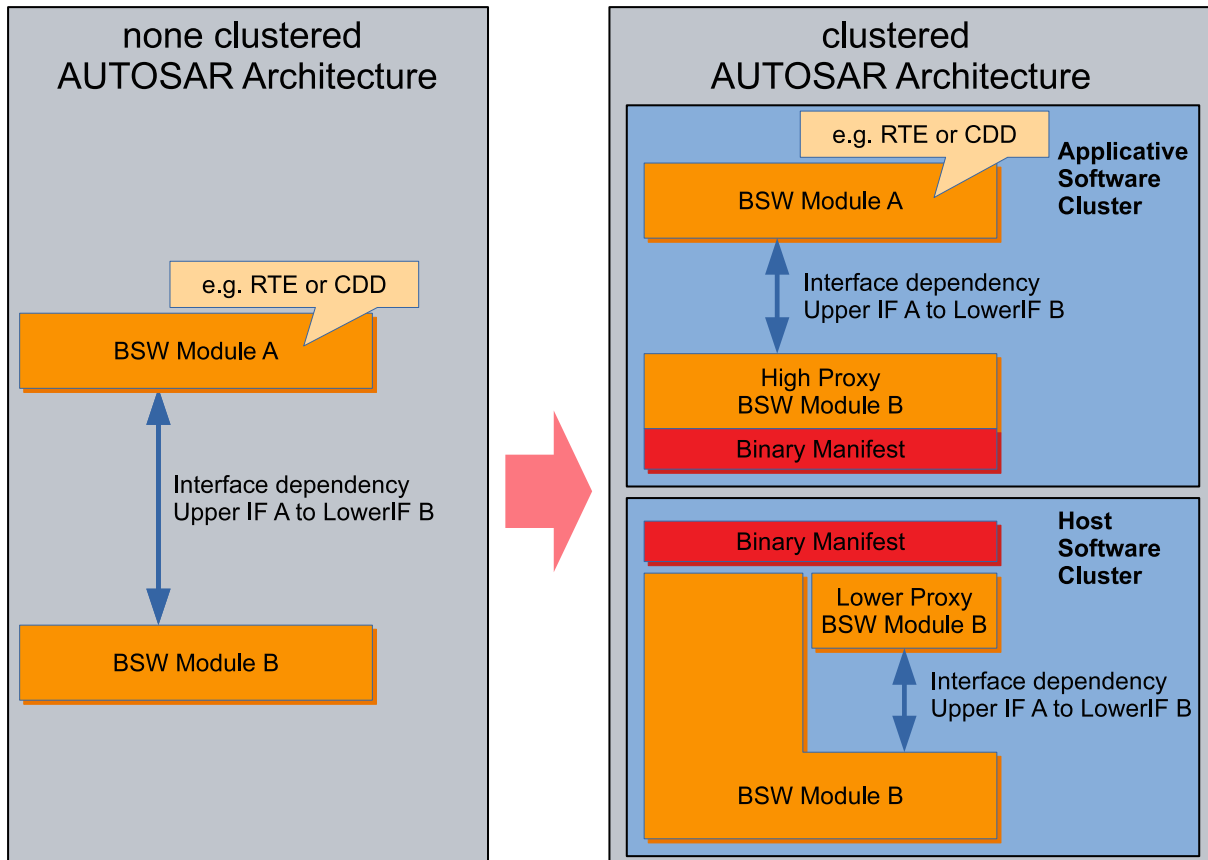


Figure 7.9: Proxy module approach

The C-interfaces of the lower layer BSW module (Module B), towards the higher layer BSW module (Module A), are provided by the High Proxy Module B, in the Software Cluster SWCL 1. The C-interfaces, ECU-C, and Component Descriptions of Module B are provided in the same name space / ARPackage structure as the original Module B. For the users of Module B (in this example Module A), the substitution of Module B by the High Proxy Module B is transparent. Usually, the offered API functions are almost without logic, and are just redirecting the calls towards the substituted module.

In the *Host Software Cluster*, the Low Proxy Module acts as a substitute for all 'user modules' in other Software Clusters. It implements the callback functions, which are normally configured in Module B.

The High Proxy Module and Low Proxy Module are connected via the means of the *Binary Manifest*. Therefore, the functions of the High and Low Proxy modules may contain functionality, which handles this *Binary Manifest* link and the situations if the link is missing. For instance, returning a reasonable error code or even providing some replacement value.

In addition, the High Proxy Module can implement functionality to localize configuration decisions. For instance, in order to support a SWCL local growing of the NV block over integration steps, it could be possible to adjust the locally used size of a NV block. Of course, this is only possible if the configured size in the real NvM is sufficiently large.

By this approach, the SWCs and BSW modules inside the [Applicative Software Clusters](#) call the common BSW inside the [Host Software Cluster](#) via proxies. When doing so, the partitioning of the ECU SW has to be considered, as described in section [7.4.2.2](#) Partitions.

This section specifies the general Proxy Module pattern. Additionally, requirements for specific Proxy Modules are in section [7.4.3](#). Additional Proxy Modules might be standardized by AUTOSAR in future releases.

7.4.2 Abstract Proxy Module Pattern

This section provides some generic design principles and common requirements, relevant for Proxy Module implementations.

Basically, a High Proxy Module provides a kind of facade, which hides the Software Cluster Connection specific mechanisms to access the real BSW Module. In the AUTOSAR Architecture, such a facade has to hide dependencies to

- Standardized Interfaces
- Standardized AUTOSAR Interfaces and the
- ECU Configuration

In addition to the essential need to provide the interfaces to the BSW Module user in the [Applicative Software Cluster](#), the [Proxy Module](#) should implement abstraction functionalities. Those abstraction functionalities decouple the needed configuration in the [High Proxy Module](#), from the configuration done in the BSW Module of the [Host Software Cluster](#). The introduction of abstraction functionalities is a case-by-case decision. On the one hand, they might need a functional support of the according BSW Module, or might lead to functional restrictions. On the other hand, such abstractions are beneficial, since in some cases they can be used to avoid a re-configuration of the [Host Software Cluster](#), after change in the [Applicative Software Cluster](#).

Please note that the following requirements only have limited applicability to an OS Proxy, since the OS APIs are already designed to be called from all the partitions.

7.4.2.1 General Proxy functionality

[SWS_SwCluC_02000]{DRAFT} [The Proxy Module shall support that id values, required to access the BSW service API of the [Host Software Cluster](#), can change, without reconfiguration or rebuild of the [Applicative Software Cluster](#) using the BSW service.]()

7.4.2.2 Partitions

Partitions are shared between the [Host Software Cluster](#) and [Applicative Software Clusters](#). This also applies to the properties of the related [OsApplications](#) and [OsTasks](#). Consequently, [Applicative Software Clusters](#) are not strictly separated from the [Host Software Cluster](#). Due to performance reasons, it is possible to share the identical partition of the [Host Software Cluster](#) with many [Applicative Software Clusters](#). If [Applicative Software Clusters](#) share the same partition, they are also not strictly separated from each other. In contrast, if a [Software Cluster](#) has multiple partitions (e.g. on different safety levels), those partitions are separated from each other.

When a partition crossing function call is required, usually some close interaction with the Operating System is necessary - especially in case of different safety levels. However, the Operating System is not directly available inside an [Applicative Software Cluster](#). Additionally, it needs to be considered that passing synchronous function calls cross partitions, may have an impact on the scheduling behavior of the software. Therefore, it is preferable to do this in the [Host Software Cluster](#), or - even better - to avoid this by implementing the master satellite pattern in the related BSW modules.

As a consequence:

Inside a partition, such direct function calls to the BSW module in the [Host Software Cluster](#) are easily possible, with little overhead. If a BSW service is needed in a specific partition (including safety partition), the [Host Software Cluster](#) should offer the service interface in this partition. If this is not possible, the according transition shall be implemented in the [Host Software Cluster](#). To support this, the [Proxy Modules](#) have to grant a partition write access to their interfaces.

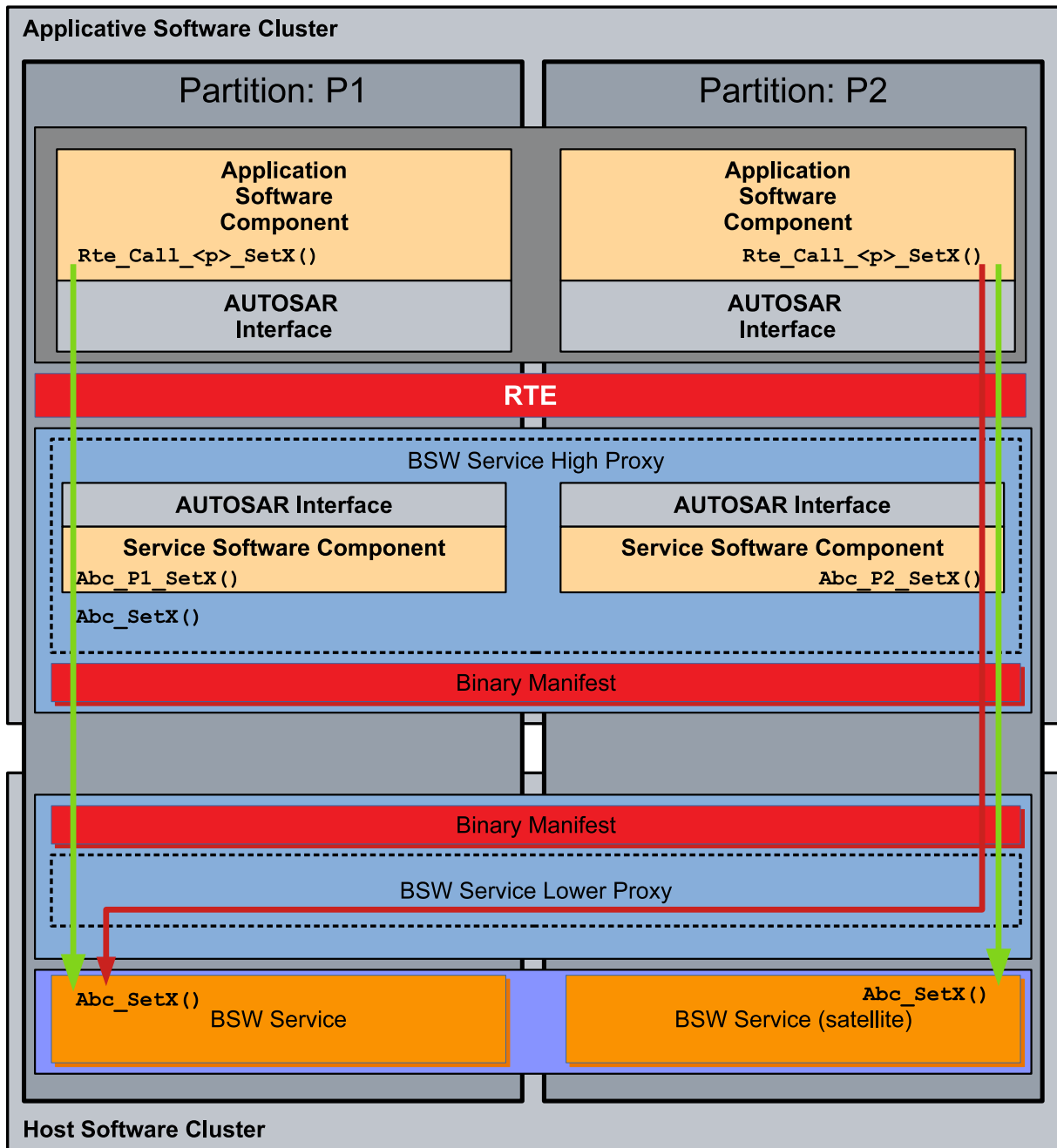


Figure 7.10: Proxy Modules and Partitions

Figure 7.10 illustrates this approach. The AUTOSAR Interfaces are offered in each partition, via a dedicated Service Software Component. The related Standardized AUTOSAR Interface is provided in exactly one Partition, in which the BSW module integration is supported.

A call from the Application Software Component is passed vertically through the partition, down to the BSW Module or its satellite (green arrow).

If such a satellite is not available, the Low Proxy Module can implement (e.g. via SchM) a cross partition call (red arrow).

[SWS_SwCluC_02001]{DRAFT} [The High Proxy shall provide the applicable Standardized AUTOSAR Interfaces once per configured `EcucPartition`. In doing so, the APIs are provided with the original `Mip`, at the `EcucPartition` where the `SwCluCNativeBswApi` is set to `true`. For all the other BSW API instances, the `Mip` is replaced by `<Mip>_<shortName of EcucPartition>`.]()

[SWS_SwCluC_02002]{DRAFT} [The High Proxy shall create one partition specific Service Software Component per configured `EcucPartition`, with the name `<Mip>_<EcucPartition shortName>`.]()

[SWS_SwCluC_02003]{DRAFT} [The High Proxy shall provide the Ports belonging to a specific `CpSoftwareClusterServiceResource`, at the partition specific Service Software Component, to which the using `SwComponentPrototype` is mapped to.]()

7.4.2.3 Unconnected Service Resources

[SWS_SwCluC_02004]{DRAFT} [The High Proxy shall implement a 'reasonable' behavior for a service resource, if the connection to the resource provider does not exist. In any case, the OUT arguments of functions shall return a 'neutral' value, or the value which can be assumed after a reset. The functional behavior should be like the state directly after a reset, before further activity occurred.]()

Note: To decide what is 'reasonable', a functional understanding of the service resource is required. Therefore, [\[SWS_SwCluC_02004\]](#) can only give some rough expectations.

7.4.2.4 Ecu Configuration Principles

[SWS_SwCluC_02005]{DRAFT} [To resolve the ECU Configuration dependency, the Proxy Module implementation shall define a Vendor Specific Module Definition [10], where the ECU Configuration container and parameters are used to configure the interface towards the user of the substituted BSW Module.]()

Typically, those are the containers and parameters configuring the variation (e.g. existence) of Standardized AUTOSAR Interfaces, or parameters relevant for symbolic name values.

In addition, these containers need to have a relationship to the software cluster resource pool, in order to manage the configuration needs exchange between the different `Software Cluster` providers. Furthermore, the `CpSoftwareClusterServiceResource` element defines the `globalResourceId`, which is required for the resource entries in the `Binary Manifest`.

[SWS_SwCluC_02006]{DRAFT} [The Proxy Module implementation shall define an `EcucContainerDef`, which defines a mapping between

- the [EcucContainerDef](#)(s) identifying the module user channel in the `StMD` of substituted BSW Module

AND

- the [CpSoftwareClusterServiceResource](#) representing the user channel in the software cluster resource pool. The [EcucForeignReferenceDef](#) shall be named `SwCluCResourceRef`.

]()

Note that this [EcucContainerDef](#) of [\[SWS_SwCluC_02006\]](#) can also be used to add additional Parameters, to control the connection behavior between `High Proxy` and `Low Proxy`. For an example, see: [SwCluCnvMProxyNvBlock](#).

[SWS_SwCluC_02007]{DRAFT} [The Proxy Module implementation shall define, for each different Proxy Module, an [EcucEnumerationParamDef](#) named `SwCluCProxyGeneration<Mip>`, with the literals

- `PROXY_DISABLED`, to disable the according Proxy Module code and AUTOSAR model generation
- `HIGH_PROXY`, to enable the according High Proxy Module code and AUTOSAR model generation
- `LOW_PROXY`, to enable the according Low Proxy Module code and AUTOSAR model generation

]()

7.4.2.5 Proxy Modules and Binary Manifest

It is the responsibility of the [Proxy Module](#) implementation to put its [Resource Entries](#) into the [Binary Manifest](#) of the respective `Software Cluster`. This includes service user specific channels (e.g. for a specific NV block), or some general links - typically named base socket - to link some generic infrastructure between `Low` and `High Proxy` (e.g. the set of BSW APIs).

[SWS_SwCluC_02008]{DRAFT} [The [Proxy Module](#) implementation of the [Software Cluster Connection](#) shall put one [Resource Entry](#), for each associated [CpSoftwareClusterServiceResource](#), into the [Binary Manifest](#).]()

7.4.3 Specific Proxy Module Requirements

7.4.3.1 OS Proxy

7.4.3.1.1 Enable OS Proxy Generation

[SWS_SwCluC_02200]{DRAFT} [The OS High Proxy code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter `SwCluCProxyGenerationOs` is set to `HIGH_PROXY`.]()

[SWS_SwCluC_02201]{DRAFT} [The OS LOW Proxy code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter `SwCluCProxyGenerationOs` is set to `LOW_PROXY`.]()

7.4.3.1.2 General OS Proxy functionality

7.4.3.1.3 Overview

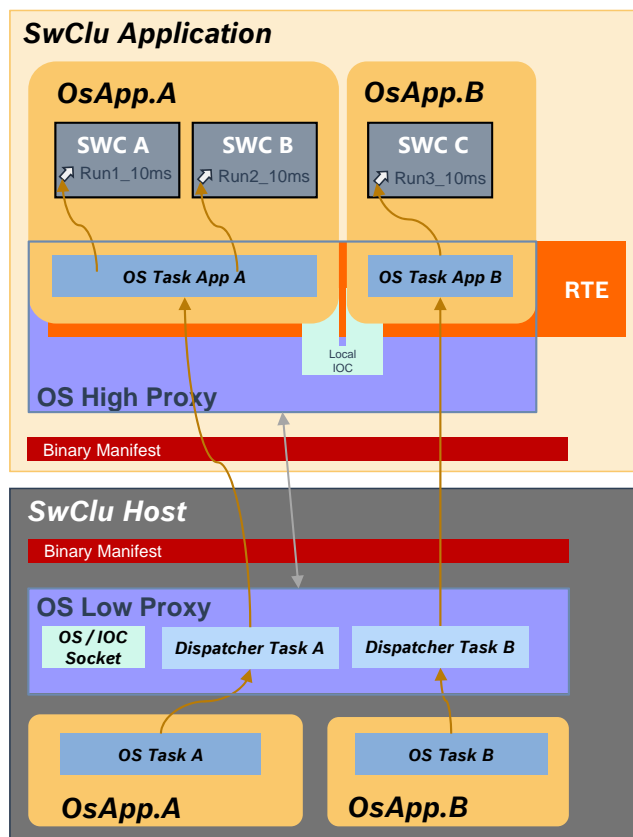


Figure 7.11: Overview of Os Proxy

As explained in chapter 7.1.1 and 7.4.1, the concept of [Software Clusters](#) also has to provide a way to divide some parts of an AUTOSAR [Os](#) into smaller units (see document [12] as reference for [Os](#)). Basically, [runnables](#) of a [Software Cluster](#) shall be executed on a target [machine](#). Since the system is divided into a machine dependent part ([Host Software Cluster](#)), and several [machine](#) independent parts ([Applicative Software Clusters](#)), an interface layer in between is required. This layer will be realized using the proxy concept.

The [Applicative Software Cluster](#) contains proxies for those parts of the OS implementation that are needed to complete the runtime environment (RTE). This is called [Os High Proxy](#).

The [Host Software Cluster](#) will implement the AUTOSAR [Os](#), as well as the overall configuration of the [machine](#). The [Os Low Proxy](#) makes this AUTOSAR [Os](#) accessible by the [Os High Proxy](#), via the [Binary Manifest](#). Both are linked together via the information held in the [Binary Manifest](#), which enables an API exchange between [Software Clusters](#).

The OS features (e.g. tasks, resources, spinlocks) supported by the [Os High Proxy](#) are configured with [EcucModuleConfigurationValues](#) for an [Os EcucModuleDef](#), with according [OsTask](#), [OsResource](#), [OsSpinlock](#), etc. containers. In the [Applicative Software Cluster](#) and in the [Host Software Cluster](#), the configuration of [SwCluCOsProxyOsTask](#), [SwCluCOsProxyOsResource](#), [SwCluCOsProxyOsSpinlock](#), etc. containers map the OS configuration containers to [CpSoftwareClusterServiceResources](#), defining each the [globalResourceId](#) and the [isMandatory](#) attribute, relevant for the corresponding [Resource Entry](#) in the [Binary Manifest](#).

The [Os Low Proxy](#) provides an [Os](#)- and [IOC](#)-socket for each [Applicative Software Cluster](#). It also provides a dispatcher, which is responsible for calling each [Os-Task](#) of an [Applicative Software Cluster](#). The [Os](#)- and [IOC](#)-socket provides basic hooks to the [Applicative Software Cluster](#). These are used, for example, to do basic initialization of local [Os](#) and [IOC](#) channels, to forward general APIs like [spinlock](#), or to do a context switch. The dispatcher controls and calls the [Os-Tasks](#) from the [Applicative Software Cluster](#). This could, for example, be implemented as [runnables](#), which are indirectly calling the [Os High Proxy Task](#), depending on the entry found in the [Binary Manifest](#).

The [Host Software Cluster](#) provides the implementation of [Os Applications](#), and an abstraction of them in the [Os Low Proxy](#). The [Os-Application](#) context inside an [Applicative Software Cluster](#) matches to the corresponding [Os Application](#) in the [Host Software Cluster](#). The [Os High Proxy](#) task is executed in the same context as the low proxy task, and has the same level of priority, trust and access.

As mentioned before, a connection between [Os Low Proxy](#) and [Os High Proxy](#) is required. It will be established with the mechanism described in chapter 7.1.5.2 ([Connecting Resources](#)). In this chapter, the metaclasses for [Os service resources](#) are

defined: `CpSoftwareClusterServiceResources`. Essential to run a `Applicative Software Cluster` are resources for the Os- and IOC-socket and the Os-ProxyTasks. These resources are part of the system's resource pool (`CpSoftwareClusterResourcePool`), which includes the description of their relevant attributes and a unique `ResourceID` (`CpSoftwareClusterResource.globalResourceId`). A valid connection can only be established, if all attributes configured for those resources match during the connection process.

To achieve a robust system, and to avoid erroneous behavior, it is recommended to mark Os service resources as mandatory.

Detailed requirements can be found below.

7.4.3.1.4 General requirements

[SWS_SwCluC_02202]{DRAFT} [The Os Proxy shall support the execution of Os-Tasks from an `Applicative Software Cluster`, in the context of the `Host Software Cluster`.]()

[SWS_SwCluC_CONSTR_02203]{DRAFT} Constraint for Interrupts [Interrupts are not supported for an `Applicative Software Cluster`, since it does not have access to the hardware or peripherals.]()

[SWS_SwCluC_02204]{DRAFT} [The Os Proxy shall support a socket mechanism to hook several `Applicative Software Clusters` onto the `Host Software Cluster`.]()

For example, this can be used to support `Applicative Software Cluster` local IOC communication paths.

[SWS_SwCluC_02205]{DRAFT} [The Os High Proxy interfaces shall be described with the parameters, and according containers, as defined in Os `EcucModuleDef`.]()

[SWS_SwCluC_02206]{DRAFT} [The Os Proxy interfaces shall be configured as a service resource, with the parameters as defined in the `SwCluCOsProxy` container.]()

7.4.3.1.5 OS High Proxy

[SWS_SwCluC_02210]{DRAFT} [The Os High Proxy shall provide the header file `Os.h` and `Os_Cfg.h`.]()

[SWS_SwCluC_02211]{DRAFT} [The Os High Proxy shall provide a proxy Os-Task, to which events from the `Applicative Software Cluster` can be mapped, during the design of an `Applicative Software Cluster`.]()

[SWS_SwCluC_02230]{DRAFT} [The Os High Proxy shall support the implementation of a task body of the proxy `OsTask`, with the `TASK()` macro.]()

[SWS_SwCluC_02212]{DRAFT} [The `Os High Proxy` shall provide a set of `Os-Task` specific APIs that are required by the `Applicative Software Cluster`:

- `ActivateTask`
- `ChainTask`
- `TerminateTask`

]()

[SWS_SwCluC_02213]{DRAFT} [If `SwCluCOsProxyOsTask.SwCluCOsProxy-TaskActivation.PASS_TASK_ACTIVATION` is set for a specific `OsTask`, the API `ActivateTask` called for this OS task shall trigger a task activation in the `Host Software Cluster`.]()

[SWS_SwCluC_02229]{DRAFT} [If `SwCluCOsProxyOsTask.SwCluCOsProxy-TaskActivation.OMIT_TASK_ACTIVATION` is set for a specific `OsTask`, the API `ActivateTask` called for this OS task returns without effect.]()

[SWS_SwCluC_CONSTR_02228]{DRAFT} **Constraint for `ActivateTask`** [`ActivateTask` is only allowed to be called in the task body of the `Applicative Software Cluster`.]()

[SWS_SwCluC_02214]{DRAFT} [The API for `ChainTask` shall be implemented as macro with a return statement in the `Applicative Software Cluster`, to be used in the context of the task body, and will end the execution of the proxy task.]()

Please note: The OS task chaining can only be handled in the OS implementation in the `Host Software Cluster`.

[SWS_SwCluC_CONSTR_02229]{DRAFT} **Constraint for `ChainTask`** [`ChainTask` is only allowed to be called in the task body of the `Applicative Software Cluster`.]()

[SWS_SwCluC_02215]{DRAFT} [The API for `TerminateTask` shall be implemented as macro with a return statement in the `Applicative Software Cluster`, to be used in the context of the task body, to be able to end the proxy tasks execution.]()
()

Please note: The real OS task termination can only be handled in the OS implementation in the `Host Software Cluster`.

[SWS_SwCluC_CONSTR_02230]{DRAFT} **Constraint for `TerminateTask`** [`TerminateTask` is only allowed to be called in the task body of the `Applicative Software Cluster`.]()

[SWS_SwCluC_02216]{DRAFT} [The `Os High Proxy` shall provide default handles for resources, in case a connection to the host system could not be established successfully between `Applicative Software Cluster` and the `Host Software Cluster`. The default implementation shall be a neutral behavior, with the return value `E_OK`, to support "standard status" from OS.]()

See the section [7.4.3.1.6 'Os Low Proxy'](#), for an explanation of how a missing connection is handled in the [Host Software Cluster](#).

[SWS_SwCluC_02217]{DRAFT} [The [Os High Proxy](#) shall have a service resource [SwCluCOsProxyOsTask](#), for each [OsTask](#) required in the [Applicative Software Cluster](#). For this resource, an entry in the [Binary Manifest](#) shall be created, to be able to link the task to its [OsLowProxy](#) implementation in the [Host Software Cluster](#). Since the content of the [OsTask](#) is created by the RTE, this API shall be used and registered as service resource. It is recommended that this resource is marked with the mandatory flag ([SwCluCBManifIsMandatory](#)).]()

[SWS_SwCluC_02218]{DRAFT} [The [Os High Proxy](#) shall contain a required service resource [SwCluCOsProxyOsTaskDispatcher](#). The [Proxy-OsTask](#) shall be called in that context. For this resource, an entry in the [Binary Manifest](#) shall be created, to be able to link it to the [OsLowProxy](#) dispatcher in the [Host Software Cluster](#). It is recommended that this resource is marked with the mandatory flag ([SwCluCBManifIsMandatory](#)).]()

[SWS_SwCluC_02219]{DRAFT} [The [Os High Proxy](#) shall contain a required service resource [SwCluCOsProxyFunctionDispatcher](#). This function dispatcher is used to call functions in context of the [Host Software Cluster](#). For this resource, an entry in the [Binary Manifest](#) shall be created, to be able to link it to the [OsLowProxy](#) dispatcher in the [Host Software Cluster](#).]()

For example, to call an [Applicative Software Cluster](#) local [Bsw-Module Init](#) function, in dedicated context of the [Host Software Cluster](#).

[SWS_SwCluC_02220]{DRAFT} [The [Os High Proxy](#) shall contain a required service resource [SwCluCOsProxyOsBaseSocket](#). This collects all hooks needed to be linked to the [Host Software Cluster](#). It also contains [Applicative Software Cluster](#) local [Os](#) specific functions for [StartOs](#), [initialize](#) and [shutdown](#). It should also cover the local [IOC](#), if present. It is recommended that this resource is marked with the mandatory flag ([SwCluCBManifIsMandatory](#)).]()

Please note: This is highly vendor specific.

[SWS_SwCluC_02221]{DRAFT} [Since a [Applicative Software Cluster](#) can contain several [OS Applications](#), the [Os High Proxy](#) shall be able to provide a local [IOC](#) implementation for the cluster internal communication paths between [OS Applications](#). The link to the [Host Software Cluster](#) shall be established as part of the [SwCluCOsProxyOsBaseSocket](#) service resource.]()

[SWS_SwCluC_02222]{DRAFT} [The [Os High Proxy](#) shall require a service resource [SwCluCOsProxyOsApplication](#), which is used in the [Applicative Software Cluster](#), and needs to be matched to the [Host Software Cluster](#).]()

This service resource represents the actual configuration of the [Applicative Software Cluster](#)'s local [OS Application](#), with its parameters and linked local [OS objects](#).

[SWS_SwCluC_02223]{DRAFT} [The [Os High Proxy](#) shall require a service resource [SwCluCOsProxyOsResource](#), which is used in the [Applicative Software Cluster](#), and needs to be matched to the [Host Software Cluster](#).]()

[SWS_SwCluC_02224]{DRAFT} [The [Os High Proxy](#) shall require a service resource [SwCluCOsProxyOsSpinlock](#), which is used in the [Applicative Software Cluster](#), and needs to be matched to the [Host Software Cluster](#).]()

[SWS_SwCluC_02225]{DRAFT} [The [Os High Proxy](#) shall provide the following APIs for the [Applicative Software Cluster](#):

- [GetResource](#)
- [ReleaseResource](#)
- [SuspendOSInterrupts](#)
- [ResumeOSInterrupts](#)
- [GetSpinlock](#)
- [ReleaseSpinlock](#)
- [GetApplicationID](#)

Each of these APIs corresponds to a service resource in the [Os Low Proxy](#). There will be an entry in the [Binary Manifest](#), to require these service resources, which are described here: [SwCluCOsProxy](#).]()

[SWS_SwCluC_02226]{DRAFT} [The [Os High Proxy](#) shall NOT support the following APIs for the [Applicative Software Cluster](#).

- [StartScheduleTable](#)
- [StopScheduleTable](#)
- [Schedule](#)
- [DisableAllInterrupts](#)
- [EnableAllInterrupts](#)
- [SuspendAllInterrupts](#)
- [ResumeAllInterrupts](#)

]()

[SWS_SwCluC_02227]{DRAFT} [The configuration of the [Os High Proxy](#) for a complete [Applicative Software Cluster](#) shall be specified with a set of ECUC-parameters defined here: [SwCluCOsProxy](#).]()

7.4.3.1.6 OS Low Proxy

[SWS_SwCluC_02250]{DRAFT} [The *Os Low Proxy* shall only be provided by the *Host Software Cluster*.]()

[SWS_SwCluC_02251]{DRAFT} [The configuration of the *Os Low Proxy* for a complete *Host Software Cluster* shall be specified with a set of ECUC-parameters defined here: *SwCluCOsProxy*. The configuration of the *Os Low Proxy* shall be derived from the implemented configuration for the *Os* in the *Host Software Cluster*.]()

[SWS_SwCluC_02252]{DRAFT} [The *Os Low Proxy* shall provide a socket for the *Os High Proxys* of the *Applicative Software Clusters*. For each *Applicative Software Cluster*, a corresponding service resource *SwCluCOsProxyOsBaseSocket* and configuration shall be provided.]()

[SWS_SwCluC_02253]{DRAFT} [The *Os Low Proxy* shall provide for each *SwCluCOsProxyOsBaseSocket* a *SwCluC_OsProxy_Init* API, to initialize the *Os High Proxy* of an *Applicative Software Cluster*. This shall be one element of the service resource *SwCluCOsProxyOsBaseSocket*.]()

For example, an Init-Callback could be provided for each *Applicative Software Cluster*, and called from the *Os Low Proxy* during initialization phase of the *Host Software Cluster*.

[SWS_SwCluC_02254]{DRAFT} [The *Os Low Proxy* shall provide an API to start the *Os High Proxy* of an *Applicative Software Cluster*. This shall be one element of the service resource *SwCluCOsProxyOsBaseSocket*.]()

For example, a Startup hook could be provided for each *Applicative Software Cluster*.

[SWS_SwCluC_02255]{DRAFT} [The *Os Low Proxy* shall provide an API to shut-down the *Os High Proxy* of an *Applicative Software Cluster*. This shall be one element of the service resource *SwCluCOsProxyOsBaseSocket*.]()

For example, a shutdown hook could be provided for each *Applicative Software Cluster*.

[SWS_SwCluC_02256]{DRAFT} [The *Os Low Proxy* shall be able to provide an API to connect IOC from *Os High Proxy* of an *Applicative Software Cluster*. This shall be one element of the service resource *SwCluCOsProxyOsBaseSocket*.]()

For example, an IOC-Init callback could be provided for each *Applicative Software Cluster*, and called from the *Os Low Proxy*, if a connection is established.

Dispatchers

[SWS_SwCluC_02270] [The *Os Low Proxy* shall have a service resource *SwCluCOsProxyOsTaskDispatcher*, for each task dispatcher provided in the *Host Software Cluster*. For this resource, an entry in the *Binary Manifest* shall be

created, to be able to link the task dispatcher to its `Os High Proxy` implementation in the `Applicative Software Cluster`.]()

Please note section 'Overview of Os Proxy' 7.11.

Example: The `Host Software Cluster` has a full configuration of the `Os`, and is prepared to host several `Applicative Software Cluster`, with their local RTE, to be able to execute the runnables of the `Applicative Software Clusters` `Software` components. This preconfiguration will preserve positions in an `OsTask`, like for events of runnables. Instead of executing a runnable, a so called dispatcher will be executed. This dispatcher acts as a link to `Applicative Software Clusters` task container, which will then execute all mapped runnables for this specific part of the task.

A connection between `Software Clusters` is established, when `ResourceIDs` in the `Binary Manifests` matches. This is necessary for service resources of the task dispatcher and the task itself. It is advised to check the connection status of these, in the implementation of the task dispatcher.

[SWS_SwCluC_02259]{DRAFT} [The `Os Low Proxy` shall provide a service resource `SwCluCOsProxyOsTask`, for each `Applicative Software Clusters` task configuration that the system shall be prepared for. For this resource, an entry in the `Binary Manifest` shall be created, to be able to link it to the `Os High Proxy` task in the `Host Software Cluster`.]()

Please note that the `SwCluCOsProxyOsTask` container has a reference to a `OsTask` container. This reference shall be used in the `Host Software Cluster` and the `Applicative Software Cluster` to obtain the `OsTask` configuration relevant for the guard value calculation. Hence, it is ensured that `Host Software Cluster` and `Applicative Software Cluster`'s `OsTask` configuration is compatible. See as well **[SWS_SwCluC_90004]**.

[SWS_SwCluC_02271] [The `Os Low Proxy` shall have a service resource `SwCluCOsProxyFunctionDispatcher`, for each function dispatcher provided in the `Host Software Cluster`. For this resource, an entry in the `Binary Manifest` shall be created, to be able to link the function dispatcher to its `Os High Proxy` implementation in the `Applicative Software Cluster`.]()

Example: The `Host Software Cluster` has a full configuration of the BSW. To initialize all modules properly, a call to `Applicative Software Cluster` might be necessary. For example, to initialize a local RAM area there. Since this function or runnable might not be known by the RTE as runnable of a `Software Component`, this function can be called via the `Os proxy function dispatcher`. This dispatcher can be configured in the `Host Software Cluster`, to be called from ECUM or an `Ini-Task`. Each dispatcher provides its own service resource in the `Binary Manifest`. If a successful link to another `Software Cluster` is established, the call can be executed from the `Host Software Cluster` into the `Applicative Software Cluster`.

Scheduling of dispatchers in Host Software Cluster

[SWS_SwCluC_02272] [The `Os Low Proxy` shall create one partition specific `Service Software Component` per configured `EcucPartition` for the dispatcher, with the name `SwCluC_Dispatcher_<EcucPartition shortName>`.]()

[SWS_SwCluC_02273] [The `Os Low Proxy` shall create one `RunnableEntity` per configured `SwCluCOsProxyOsTaskDispatcher` container in the partition specific `Service Software Component` according **[SWS_SwCluC_02272]**.]

The `EcucPartition` is determined by the parameter `SwCluCOsProxyEcucPartitionRef`, if configured or alternatively by the `EcucPartition` to which the related `OsTask` belongs. The attributes of the `RunnableEntity` shall be set as following:

- `minimumStartInterval` is set to the parameter value `SwCluCOsProxyMinimumStartInterval`
- `symbol` is set to the name of the task dispatcher's C-function used for implementation of **[SWS_SwCluC_02270]**
- `canBeInvokedConcurrently` is set to `false`

]()

[SWS_SwCluC_02274] [The `Os Low Proxy` shall create one `BackgroundEvent` per configured `SwCluCOsProxyDispatcherBackgroundEvent` container in the partition specific `Service Software Component` according **[SWS_SwCluC_02272]**. The attributes of the `BackgroundEvent` shall be set as following:

- `shortName` is set according the container's `shortName`
- `startOnEvent` is set to the `RunnableEntity` according **[SWS_SwCluC_02273]** related to the owning `SwCluCOsProxyOsTaskDispatcher` container.
- `disabledMode(s)` are created according the `SwCluCOsProxyDisabledModeInstanceRef(s)` where for each different `PPortPrototype` an `RPortPrototype` is created at the owing `Service Software Component` according **[SWS_SwCluC_02272]** and used as `contextPort`.

]()

[SWS_SwCluC_02275] [The `Os Low Proxy` shall create one `TimingEvent` per configured `SwCluCOsProxyDispatcherTimingEvent` container in the partition specific `Service Software Component` according **[SWS_SwCluC_02272]**. The attributes of the `TimingEvent` shall be set as following:

- `shortName` is set according the container's `shortName`
- `startOnEvent` is set to the `RunnableEntity` according **[SWS_SwCluC_02273]** related to the owning `SwCluCOsProxyOsTaskDispatcher` container

- `period` value is set according the `SwCluCOsProxyTimingEventPeriod` parameter
- `disabledMode(s)` are created according the `SwCluCOsProxyDisabledModeInstanceRef(s)` where for each different `PPortPrototype` an `RPortPrototype` is created at the owing Service Software Component according [SWS_SwCluC_02272] and used as `contextPort`.

]()

[SWS_SwCluC_02276] [The `Os Low Proxy` shall create one `SwcModeSwitchEvent` per configured `SwCluCOsProxyDispatcherModeSwitchEvent` container in the partition specific Service Software Component according [SWS_SwCluC_02272]. The attributes of the `SwcModeSwitchEvent` shall be set as following:

- `shortName` is set according the container's `shortName`
- `startOnEvent` is set to the `RunnableEntity` according [SWS_SwCluC_02273] related to the owning `SwCluCOsProxyOsTaskDispatcher` container
- `activation` value is set according the `SwCluCOsProxyDispatcherActivation` parameter
- `modes` are created according the `SwCluCOsProxyModeInstanceRef(s)` where for each different `PPortPrototype` an `RPortPrototype` is created at the owing Service Software Component according [SWS_SwCluC_02272] and used as `contextPort`.
- `disabledMode(s)` are created according the `SwCluCOsProxyDisabledModeInstanceRef(s)` where for each different `PPortPrototype` an `RPortPrototype` is created at the owing Service Software Component according [SWS_SwCluC_02272] and used as `contextPort`.

]()

[SWS_SwCluC_02277] [The `Os Low Proxy` shall create one `ExternalTriggerOccurredEvent` per configured `SwCluCOsProxyDispatcherExternalTriggerOccurredEvent` container in the partition specific Service Software Component according [SWS_SwCluC_02272]. The attributes of the `ExternalTriggerOccurredEvent` shall be set as following:

- `shortName` is set according the container's `shortName`
- `startOnEvent` is set to the `RunnableEntity` according [SWS_SwCluC_02273] related to the owning `SwCluCOsProxyOsTaskDispatcher` container
- `trigger` is set according the `SwCluCOsProxyTriggerInstanceRef` where for each different `PPortPrototype` an `RPortPrototype` is created at the

owing Service Software Component according [SWS_SwCluC_02272] and used as `contextRPort`.

- `disabledMode(s)` are created according the `SwCluCOsProxyDisabledModeInstanceRef(s)` where for each different `PPortPrototype` an `RPortPrototype` is created at the owing Service Software Component according [SWS_SwCluC_02272] and used as `contextPort`.

]()

OsApplication

[SWS_SwCluC_02262]{DRAFT} [The `Os Low Proxy` shall provide a service resource `SwCluCOsProxyOsApplication`, for each `OsApplication` that is configured in the `Host Software Cluster`, and prepared for use in `Applicative Software Clusters`.]()

OsResource

[SWS_SwCluC_02263]{DRAFT} [The `Os Low Proxy` shall provide a service resource `SwCluCOsProxyOsResource`, for each `OsResource` used in `Applicative Software Clusters`.]()

OsSpinlock

[SWS_SwCluC_02264]{DRAFT} [The `Os Low Proxy` shall provide a service resource `SwCluCOsProxyOsSpinlock`, for each `OsSpinlock` used in `Applicative Software Clusters`.]()

OS APIs

[SWS_SwCluC_02265]{DRAFT} [The `Os Low Proxy` shall provide a basic service implementation for the following APIs, to be used and connected with an `Applicative Software Cluster`:

- `GetResource`
- `ReleaseResource`
- `SuspendOSInterrupts`
- `ResumeOSInterrupts`
- `ReleaseResource`
- `GetSpinlock`
- `ReleaseSpinlock`
- `GetApplicationID`

These APIs shall be provided with a corresponding service resource from `SwCluCOsProxy`.]()

Unconnected Service Resources

[SWS_SwCluC_02266]{DRAFT} [The [Os Low Proxy](#) shall provide default handles for resources, in case a connection could not be established successfully between [Applicative Software Cluster](#) and the [Host Software Cluster](#).]()

For example, if an [Applicative Software Cluster](#) is missing, the task-dispatcher in the host calls an empty function.

Please note: One property of a clustered system is that parts shall be independently buildable. Another property is that some parts can be absent. Therefore, "default stubs" are needed for non-local APIs. But in case of missing interfaces to [Os](#), [Applicative Software Clusters](#) cannot run successfully. The usage of a suitable mechanism to detect this case is advised. To achieve a robust system, the [Host Software Cluster](#) is still running in such a state.

7.4.3.2 NvM Proxy

The specified NvM Proxy has the following underlying design principle.

In the [Applicative Software Cluster](#), NV Blocks supported by the NvM High Proxy are configured with [EcucModuleConfigurationValues](#) for a NvM [EcucModuleDef](#), with according [NvMBlockDescriptors](#) containers. In the [Applicative Software Cluster](#) and in the [Host Software Cluster](#), the configuration of [SwCluC_NvMProxyNvBlock](#) containers map the [NvMBlockDescriptors](#) to [CpSoftwareClusterServiceResources](#), defining each the [globalResourceId](#) and the [isMandatory](#) attribute, relevant for the corresponding [Resource Entry](#) in the [Binary Manifest](#).

The connection between the [NvM High Proxy](#) and the [NvM Low Proxy](#) utilizes the explicit synchronization, as specified in document [13]. The explicit synchronization has the advantage that the addresses of RAM Blocks and ROM blocks of [Applicative Software Cluster](#) do not have to be known by the NVRAM Manager in the [Host Software Cluster](#). In addition, the function interface can be used to introduce additional functionality, like the NV block length adjustment ([[SWS_SwCluC_02105](#)]). On the other hand, this design principle cannot support temporary RAM blocks, since the NvM configuration in the [Host Software Cluster](#) may hold a different size than the NvM configuration in the [Applicative Software Cluster](#).

[SWS_SwCluC_CONSTR_02134]{DRAFT} [The NvM High Proxy shall reject configurations of temporary RAM blocks.]([SRS_SwCluC_00206](#))

Note: this means only NVRAM blocks configured with a permanent RAM block or explicit synchronization callbacks are supported in the NvM High Proxy.

In addition, the [Host Software Cluster](#) should not have a functional dependency to the content stored in NV Blocks of the [Applicative Software Cluster](#). Furthermore, the NvM High Proxy shall be able to initialize the RAM Block, even if the

related NV Block is not connected to a [Host Software Cluster](#). This requires the existence of ROM block default values.

[SWS_SwCluC_CONSTR_02135]{DRAFT} [The NvM High Proxy shall reject configurations, where no ROM block is configured via the parameter [NvMRomBlockDataAddress](#), or the parameter [NvMInitBlockCallback](#).] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_CONSTR_02141]{DRAFT} [The NvM High Proxy shall reject configurations, where the NV Block length given by [NvMNvBlockLength](#) is larger than the maximum size foreseen in the [Host Software Cluster](#), given by [SwCluCnVMPProxyNvBlockMaxLength](#).] ([SRS_SwCluC_00206](#))

Please note that [SwCluCnVMPProxyNvBlockMaxLength](#) is also part of the guard value, and therefore protected from unilateral changes.

7.4.3.2.1 Enable NvM Proxy Generation

[SWS_SwCluC_02101]{DRAFT} [The NvM High Proxy code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter [SwCluCProxyGenerationNvM](#) is set to `HIGH_PROXY`.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02144]{DRAFT} [The NvM Low Proxy code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter [SwCluCProxyGenerationNvM](#) is set to `LOW_PROXY`.] ([SRS_SwCluC_00206](#))

7.4.3.2.2 General NvM Proxy functionality

[SWS_SwCluC_02104]{DRAFT} [The NvM Proxy shall support that the BlockIds of the NvM of [Host Software Cluster](#) can change, without reconfiguration or rebuild of the [Applicative Software Cluster](#) using these NV blocks.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02105]{DRAFT} [The NvM Proxy shall support that the NV block length used in the [Applicative Software Cluster](#) can be \leq the NV block length configured in the NvM of the [Host Software Cluster](#). In case NV block length in [Applicative Software Cluster](#) is smaller, the NV block is filled with default values. The default fill value of a single byte is defined by the configuration parameter [SwCluCnVMPProxyNvBlockFillValue](#).] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02106]{DRAFT} [The NvM Proxy shall support that the NV block length used in the [Applicative Software Cluster](#) can be changed (in the range $0 \leq$ NV block length configured in NvM), without reconfiguration or rebuild of the [Host Software Cluster](#).] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02112]{DRAFT} [In case it is configured in the NvM High Proxy, the NvM Proxy shall invoke the `NvM_SingleBlockCallbackFunction` or `JobFinished` operation, when the according instance of the NvM Low Proxy is called.] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02122]{DRAFT} [In case it is configured in the NvM High Proxy, the NvM Proxy shall invoke the `NvM_InitBlockCallbackFunction` or `InitBlock` operation, when the according instance of the NvM Low Proxy is called.] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02123]{DRAFT} [In case it is configured in the NvM High Proxy, the NvM Proxy shall invoke the `NvM_ReadRamBlockFromNvm` function or `ReadRamBlockFromNvm` operation, when the according instance of the NvM Low Proxy is called.] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02124]{DRAFT} [In case it is configured in the NvM High Proxy, the NvM Proxy shall invoke the `NvM_WriteRamBlockToNvm` function or `WriteRamBlockToNvm` operation, when the according instance of the NvM Low Proxy is called.] (*SRS_SwCluC_00206*)

7.4.3.2.3 NvM High Proxy

[SWS_SwCluC_02102]{DRAFT} [The NvM High Proxy shall provide the header file `NvM.h`.] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02103]{DRAFT} [The NvM High Proxy shall provide the symbolic name values for NvBlocks, for NvM users in the *Applicative Software Cluster*.] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02125]{DRAFT} [The NvM High Proxy shall provide the functions for single block requests

- `NvM_SetDataIndex`
- `NvM_GetDataIndex`
- `NvM_SetBlockProtection`
- `NvM_GetErrorStatus`
- `NvM_SetRamBlockStatus`
- `NvM_SetBlockLockStatus`
- `NvM_CancelJobs`
- `NvM_ReadBlock`
- `NvM_WriteBlock`
- `NvM_RestoreBlockDefaults`

- [NvM_EraseNvBlock](#)
- [NvM_InvalidateNvBlock](#)
- [NvM_ReadPRAMBlock](#)
- [NvM_WritePRAMBlock](#)
- [NvM_RestorePRAMBlockDefaults](#)

for NvM users in the [Applicative Software Cluster](#).] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02126]{DRAFT} [The NvM High Proxy shall provide the functions for single block requests, according to [\[SWS_SwCluC_02103\]](#), once per configured [SwCluCNvMBaseSocket](#). In doing so, the APIs are provided with the original Mip NvM at the [EcucPartition](#), where the [SwCluCNativeBswApi](#) is set to `true`. For all the other [SwCluCNvMBaseSocket](#) instances, the Mip NvM is replaced by `NvM_<shortName of EcucPartition>`.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02136]{DRAFT} [The functions

- [NvM_ReadBlock](#)
- [NvM_WriteBlock](#)
- [NvM_RestoreBlockDefaults](#)

of the NvM High Proxy shall only accept single block requests, where the pointer to the RAM data block is set to `NULL_PTR`. Otherwise, `E_NOT_OK` is returned, and the request is discarded.] ([SRS_SwCluC_00206](#))

For example, the function `NvM_GetErrorStatus` can be provided as

```

1  /* EcucPartition "Core0_QM", SwCluCNativeBswApi == true */
2  Std_ReturnType NvM_GetErrorStatus (
3      NvM_BlockIdType BlockId,
4      NvM_RequestResultType * RequestResultPtr
5  )
6  ...
7
8  /* EcucPartition "Core1_QM", SwCluCNativeBswApi == false */
9  Std_ReturnType NvM_Core1_QM_GetErrorStatus (
10     NvM_BlockIdType BlockId,
11     NvM_RequestResultType * RequestResultPtr
12 )
13 ...

```

Note: [SwCluCNativeBswApi](#) can only be set to `true`, for at most one [SwCluCNvMBaseSocket](#) container in an [Applicative Software Cluster](#) configuration.

Callback Functions

[SWS_SwCluC_02109]{DRAFT} [The NvM High Proxy shall support the optional configuration of a [NvM_SingleBlockCallbackFunction](#), according to [\[ECUC_NvM_00562\]](#), [\[ECUC_NvM_00506\]](#) and [\[ECUC_NvM_00559\]](#).] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02110]{DRAFT} [The NvM High Proxy shall support the optional configuration of a `NvM_PNJF_{Block}` port for `NvMNotifyJobFinished` notification.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02115]{DRAFT} [The NvM High Proxy shall support the optional configuration of a `NvM_InitBlockCallbackFunction`, according to [ECUC_NvM_00561], [ECUC_NvM_00116] and [ECUC_NvM_00559].] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02116]{DRAFT} [The NvM High Proxy shall support the optional configuration of a `NvM_PNIB_{Block}` port for `NvMNotifyInitBlock` notification.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02117]{DRAFT} [The NvM High Proxy shall support the optional configuration of a `NvM_ReadRamBlockFromNvm`, according to [ECUC_NvM_00521].] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02118]{DRAFT} [The NvM High Proxy shall support the optional configuration of a `NvM_WriteRamBlockToNvm`, according to [ECUC_NvM_00520].] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02119]{DRAFT} [The NvM High Proxy shall support the optional configuration of a `NvM_PM_{Block}` port for `NvMMirror` notification.] ([SRS_SwCluC_00206](#))

Multiple Partitions

[SWS_SwCluC_02120]{DRAFT} [The NvM High Proxy shall create one partition specific Service Software Component per configured `EcucPartition`, with the name `NvM_<EcucPartition shortName>`.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02121]{DRAFT} [The NvM High Proxy shall provide the Ports belonging to a specific Nv Block

- `NvM_PS_{Block}`
- `NvM_PAdmin_{Block}`
- `NvM_PNIB_{Block}`
- `NvM_PNJF_{Block}`
- `NvM_PM_{Block}`

at the partition specific Service Software Component, to which the using `SwComponentPrototype` is mapped.] ([SRS_SwCluC_00206](#))

Handling of unconnected blocks

[SWS_SwCluC_02107]{DRAFT} [In case a NV block configured in the NvM High Proxy is not connected to any NvM Low Proxy, the NvM High Proxy shall initialize the Nv RAM Block, with the default values given by `NvMRomBlockDataAddress` and / or `NvMInitBlockCallback`.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02113]{DRAFT} [In case a NV block configured in the NvM High Proxy is not connected to any NvM Low Proxy, the function `NvM_GetErrorStatus` of the NvM High Proxy shall still return `E_OK`, and the request result is set to `NVM_REQ_RESTORED_FROM_ROM`.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02114]{DRAFT} [In case a NV block configured in the NvM High Proxy is not connected to any NvM Low Proxy, the functions

- `NvM_SetDataIndex`
- `NvM_GetDataIndex`
- `NvM_SetBlockProtection`
- `NvM_SetRamBlockStatus`
- `NvM_SetBlockLockStatus`
- `NvM_CancelJobs`
- `NvM_ReadBlock`
- `NvM_WriteBlock`
- `NvM_RestoreBlockDefaults`
- `NvM_EraseNvBlock`
- `NvM_InvalidateNvBlock`
- `NvM_ReadPRAMBlock`
- `NvM_WritePRAMBlock`
- `NvM_RestorePRAMBlockDefaults`

of the NvM High Proxy shall return `E_NOT_OK`.] ([SRS_SwCluC_00206](#))

7.4.3.2.4 NvM Low Proxy

As shown in figure 7.10, the NvM Low Proxy shall be able to invoke APIs of the BSW modules in the `Host Software Cluster`, without a partition change in the `Applicative Software Clusters`. This partition change is only needed, if the BSW module does not offer a satellite on this partition. The availability of the BSW modules on dedicated `EcucPartitions` is a property of the `Host Software Cluster`.

[SWS_SwCluC_02132]{DRAFT} [The NvM Low Proxy shall provide the entry functions for single block requests, according to [\[SWS_SwCluC_02103\]](#), once per configured `SwCluCnVMBaseSocket` at the `EcucPartition` configured by `SwCluCnVMBaseSocket.SwCluCProxyEcucPartitionRef`.

In case the `SwCluCnVMProxyUsedSatelliteRef` is set, the `EcucPartition` shall be changed, before the NVM API is called.] ([SRS_SwCluC_00206](#))

[SWS_SwCluC_02111]{DRAFT} [The NvM Low Proxy shall provide an instance of a `NvM_SingleBlockCallbackFunction` per configured NV block (`SwCluCnVMPProxyNvBlock`) in the NvM Low Proxy.] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02127]{DRAFT} [The NvM Low Proxy shall provide an instance of a `NvM_InitBlockCallbackFunction` per configured NV block (`SwCluCnVMPProxyNvBlock`) in the NvM Low Proxy.] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02128]{DRAFT} [The NvM Low Proxy shall provide an instance of a `NvM_ReadRamBlockFromNvm` per configured NV block (`SwCluCnVMPProxyNvBlock`) in the NvM Low Proxy.] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02129]{DRAFT} [The NvM Low Proxy shall provide an instance of a `NvM_WriteRamBlockToNvm` per configured NV block (`SwCluCnVMPProxyNvBlock`) in the NvM Low Proxy.] (*SRS_SwCluC_00206*)

Contrary to the call of BSW module's APIs from an `Applicative Software Cluster`, the invoked callback notifications need to change the partition in the `Host Software Cluster`, before the calls are ending up in the `Applicative Software Cluster`, if the BSW module does not offer a satellite on this partition.

[SWS_SwCluC_02133]{DRAFT} [The callback instances according to **[SWS_SwCluC_02111]**, **[SWS_SwCluC_02127]**, **[SWS_SwCluC_02128]**, and **[SWS_SwCluC_02129]**, are provided

- at the `EcucPartition` referenced by `SwCluCnVMPProxyNvBlock.SwCluCnVMPProxyNvBlockEcucPartitionRef`, if a `SwCluCnVMBaseSocket` is directly available at this `EcucPartition`

OR

- at the `EcucPartition` of the `SwCluCnVMBaseSocket` referenced by the `SwCluCnVMBaseSocket` - via `SwCluCnVMPProxyUsedSatelliteRef`, which is on the `EcucPartition` given by `SwCluCnVMPProxyNvBlock.SwCluCnVMPProxyNvBlockEcucPartitionRef`. In this case, the `EcucPartition` shall be changed, before the callback in the `High Proxy` is called.

] (*SRS_SwCluC_00206*)

[SWS_SwCluC_02108]{DRAFT} [In case a NV block configured in the NvM Low Proxy is not connected to any NvM High Proxy, the NvM Low Proxy implementation shall ensure that the content of the NV block is not modified by the NvM Low Proxy.] (*SRS_SwCluC_00206*)

Note: An access by Dcm might still change the NV block content in this case.

[SWS_SwCluC_02142]{DRAFT} [The NvM Low Proxy implementation shall describe the NV blocks provided to NvM High Proxies via `NvBlockNeeds`, derived from the `CpSoftwareClusterServiceResource.resourceNeeds`.] (*SRS_SwCluC_00206*)

Note: **[SWS_SwCluC_02142]** ensures an automated configuration of the NVRAM Manager.

[SWS_SwCluC_02143]{DRAFT} [The *NvM Low Proxy* shall provide, for each *SwCluCnVMBaseSocket*, a *SwCluC_NvMProxy_Init* API, to initialize the *NvM High Proxy* of an *Applicative Software Cluster*. This shall be one element of the service resource *SwCluCnVMBaseSocket*.]()

7.4.3.2.5 Error detection

[SWS_SwCluC_02137]{DRAFT} [If development error detection is enabled for *NvM High Proxy*, the APIs

- *NvM_SetDataIndex*
- *NvM_GetDataIndex*
- *NvM_SetBlockProtection*
- *NvM_GetErrorStatus*
- *NvM_SetRamBlockStatus*
- *NvM_SetBlockLockStatus*
- *NvM_CancelJobs*
- *NvM_ReadBlock*
- *NvM_WriteBlock*
- *NvM_RestoreBlockDefaults*
- *NvM_EraseNvBlock*
- *NvM_InvalidateNvBlock*
- *NvM_ReadPRAMBlock*
- *NvM_WritePRAMBlock*
- *NvM_RestorePRAMBlockDefaults*

shall report the DET error *NVM_E_UNINIT*, when *NvM High Proxy* is not yet initialized.]
(*SRS_SwCluC_00206*, *SRS_BSW_00369*, *SRS_BSW_00323*)

[SWS_SwCluC_02138]{DRAFT} [If development error detection is enabled for *NvM High Proxy*, the APIs

- *NvM_SetDataIndex*
- *NvM_GetDataIndex*
- *NvM_SetBlockProtection*
- *NvM_GetErrorStatus*
- *NvM_SetRamBlockStatus*

- [NvM_SetBlockLockStatus](#)
- [NvM_CancelJobs](#)
- [NvM_ReadBlock](#)
- [NvM_WriteBlock](#)
- [NvM_RestoreBlockDefaults](#)
- [NvM_EraseNvBlock](#)
- [NvM_InvalidateNvBlock](#)
- [NvM_ReadPRAMBlock](#)
- [NvM_WritePRAMBlock](#)
- [NvM_RestorePRAMBlockDefaults](#)

shall report the DET error `NVM_E_PARAM_BLOCK_ID`, when the passed `BlockId` is out of range of the `BlockIds` configured in the NvM High Proxy.]([SRS_SwCluC_00206](#), [SRS_BSW_00369](#), [SRS_BSW_00323](#))

[SWS_SwCluC_02139]{DRAFT} [If development error detection is enabled for NvM High Proxy, the APIs

- [NvM_ReadBlock](#)
- [NvM_WriteBlock](#)
- [NvM_RestoreBlockDefaults](#)

shall report the DET error `NVM_E_PARAM_ADDRESS`, when a RAM block address different than `NULL_PTR` is passed.]([SRS_SwCluC_00206](#), [SRS_BSW_00369](#), [SRS_BSW_00323](#))

7.4.3.3 Com Proxy

7.4.3.3.1 Enable Com Proxy Generation

[SWS_SwCluC_02600]{DRAFT} [The [Com High Proxy](#) code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter `SwCluCProxyGenerationCom` is set to `HIGH_PROXY`.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_02601]{DRAFT} [The [Com Low Proxy](#) code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter `SwCluCProxyGenerationCom` is set to `LOW_PROXY`.]([SRS_SwCluC_00211](#))

7.4.3.3.2 General Com Proxy functionality

7.4.3.3.3 Overview

The specified Com Proxy has the following underlying design principle.

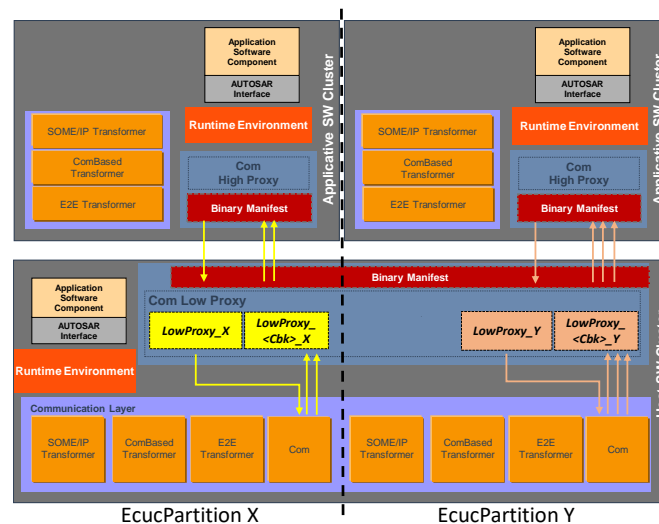


Figure 7.12: Overview of Com Proxy

The [Applicative Software Cluster](#) provides proxies for those parts of the Com implementation that are required to complete its functionality. This is called [Com High Proxy](#).

The [Host Software Cluster](#) contains the real implementation of the Com functionality and makes it available to the [Com High Proxy](#) via the respective [Com Low Proxy](#). Both [Com High Proxy](#) and [Com Low Proxy](#) are linked together based on the information available in [Binary Manifest](#), which enables the execution of Com (Callback) functions from [Applicative Software Cluster](#) to [Host Software Cluster](#) and vice-versa.

The Com Proxy as an [ComUserModule](#) is represented by one or several [ComUserModuleCnf](#). A [ComUserModuleCnf](#) is required for each [EcucPartition](#), in which the Com Proxy

- requires direct access to the Com APIs initiating transmission/reception requests of [ComSignals](#) and/or [ComSignalGroups](#)
- provides transmission/reception notification callbacks of [ComSignals](#) and/or [ComSignalGroups](#)

Effectively, a [ComUserModuleCnf](#) links a fixed set of notification callbacks in the Com to a specific [EcucPartition](#) in the Com Proxy. As consequence, the

Com Proxy has to map each `ComUserSignal` / `ComUserSignalGroup` via `ComUserSystemTemplateSystemSignalRef` / `ComUserSystemTemplateSignalGroupRef` to an `ComSignal` resp. `ComSignalGroup`.

The Com module shall provide its APIs for transmission/reception requests of the relevant `ComSignals` and/or `ComSignalGroups` on the `EcucPartition` configured in the Com Proxy.

The Com Proxy as `ComUserModule` shall provide a compatible configuration structure and content for the RTE in the `Applicative Software Cluster`. It derives its configuration of `ComSignals` and/or `ComSignalGroups` from the Com. For the `EcucPartition` assignment, the Com Proxy creates virtual main functions (Rx/Tx) and maps the respective `ComIPdus` to them. These main functions exist only in the configuration but do not have an implementation.

Even if the concept of Com user provides a lot of flexibility to support access by multiple users including their notifications, some limitations have to be considered.

In general, multiple writers can cause race conditions, if the writers are not coordinated. In addition, the required sequence of `Com_SendSignal` (of group signals) and `Com_SendSignalGroup` calls cause the risk of data inconsistency in case of preemption by multiple writers.

Since such a coordination across different Software Clusters is hardly achievable, the following restriction applies:

[SWS_SwCluC_CONSTR_02653]{DRAFT} [Sent signals or signal groups shall be provided to at most one `Applicative Software Cluster`.] (*SRS_SwCluC_00211*)

Nevertheless, reading a signal or signal group by several `Applicative Software Clusters` is possible.

7.4.3.3.4 General requirements

[SWS_SwCluC_02602]{DRAFT} [The Com Proxy shall support that the `ComHandleId` of `ComSignals` configured in Com of the `Host Software Cluster` can change, without reconfiguration or rebuild of the `Applicative Software Cluster` using these `ComSignals`.] (*SRS_SwCluC_00211*)

[SWS_SwCluC_02603]{DRAFT} [The Com Proxy shall support that the `ComHandleId` of `ComSignalGroups` resp. `ComGroupSignals` configured in the Com of the `Host Software Cluster` can change, without reconfiguration or rebuild of the `Applicative Software Cluster` using these `ComSignalGroups`.] (*SRS_SwCluC_00211*)

[SWS_SwCluC_02652]{DRAFT} [The Com Proxy shall support that the `ComUserCbkHandleId` of `ComUserSignals` resp. `ComUserSignalGroups` configured in

Com users of the [Applicative Software Cluster](#) can change, without reconfiguration or rebuild of the [Host Software Cluster](#).] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02604]{DRAFT} [The Com Proxy shall support the invocation of Com APIs from an [Applicative Software Cluster](#) in context of the [Host Software Cluster](#).] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02605]{DRAFT} [The Com Proxy shall support the invocation of Com callbacks from the [Host Software Cluster](#) in context of the [Applicative Software Cluster](#).] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02639]{DRAFT} [If the `<ComUser_CbkRxAck>` notification callback is configured in the [Com High Proxy](#) (via the [ComUserCallbackName](#) (See [ECUC_Com_10036]), where the respective [ComUserCallbackType](#) (See [ECUC_Com_10034]) is set to `COM_RX_ACK`) the configured `<ComUser_CbkRxAck>` callback shall be invoked when the associated [SwCluC_ComProxy_CbkRxAck](#) instance of the [Com Low Proxy](#) is called.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02640]{DRAFT} [If the invalid `<ComUser_CbkInv>` notification callback is configured in the [Com High Proxy](#) (via the [ComUserCallbackName](#) (See [ECUC_Com_10036]), where the respective [ComUserCallbackType](#) (See [ECUC_Com_10034]) is set to `COM_RX_INV`) the configured `<ComUser_CbkInv>` callback shall be invoked when the associated [SwCluC_ComProxy_CbkInv](#) instance of the [Com Low Proxy](#) is called.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02642]{DRAFT} [If the `<ComUser_CbkRxTOut>` notification callback is configured in the [Com High Proxy](#) (via the [ComUserCallbackName](#) (See [ECUC_Com_10036]), where the respective [ComUserCallbackType](#) (See [ECUC_Com_10034]) is set to `COM_RX_TOUT`) the configured `<ComUser_CbkRxTOut>` callback shall be invoked when the associated [SwCluC_ComProxy_CbkRxTOut](#) instance of the [Com Low Proxy](#) is called.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02643]{DRAFT} [If the `<ComUser_CbkTxAck>` notification callback is configured in the [Com High Proxy](#) (via the [ComUserCallbackName](#) (See [ECUC_Com_10036]), where the respective [ComUserCallbackType](#) (See [ECUC_Com_10034]) is set to `COM_TX_ACK`) the configured `<ComUser_CbkTxAck>` callback shall be invoked when the associated instance of the [Com Low Proxy](#) is called.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02644]{DRAFT} [If the `<ComUser_CbkTxErr>` notification callback is configured in the [Com High Proxy](#) (via the [ComUserCallbackName](#) (See [ECUC_Com_10036]), where the respective [ComUserCallbackType](#) (See [ECUC_Com_10034]) is set to `COM_TX_ERR`) the configured `<ComUser_CbkTxErr>` callback shall be invoked when the associated [SwCluC_ComProxy_COMCbkTErr](#) instance of the [Com Low Proxy](#) is called.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02645]{DRAFT} [If the `<ComUser_CbkTxTOut>` timeout notification callback is configured in the [Com High Proxy](#) (via the [ComUserCallbackName](#) (See [ECUC_Com_10036]), where the respective [ComUserCallbackType](#)

(See [ECUC_Com_10034]) is set to `COM_TX_TOUT`) the configured `<ComUser_Cbk-TxTOut>` callback shall be invoked when the associated `SwCluC_ComProxy_COM-CbkTxTOut` instance of the `Com Low Proxy` is called.]([SRS_SwCluC_00211](#))

7.4.3.3.5 Com High Proxy

[[SWS_SwCluC_02606](#)]{DRAFT} [The `Com High Proxy` shall provide the header file `Com.h`.]([SRS_SwCluC_00211](#))

[[SWS_SwCluC_02607](#)]{DRAFT} [The `Com High Proxy` shall provide the symbolic name values for the `ComSignals`, for `Com` users in the `Applicative Software Cluster`.]([SRS_SwCluC_00211](#))

[[SWS_SwCluC_02608](#)]{DRAFT} [The `Com High Proxy` shall provide the symbolic name values for the `ComSignalGroups` resp. `ComGroupSignals`, for `Com` users in the `Applicative Software Cluster`.]([SRS_SwCluC_00211](#))

[[SWS_SwCluC_02609](#)]{DRAFT} [The `Com High Proxy` shall provide the functions for transmission requests of `ComSignals` and `ComSignalGroups`

- `Com_SendSignal`
- `Com_SendDynSignal`
- `Com_SendSignalGroup`
- `Com_SendSignalGroupArray`
- `Com_SendSignalWithMetaData`
- `Com_SendDynSignalWithMetaData`
- `Com_SendSignalGroupWithMetaData`
- `Com_SendSignalGroupArrayWithMetaData`

required by the `Com` users in the `Applicative Software Cluster`.

]([SRS_SwCluC_00211](#))

[[SWS_SwCluC_02610](#)]{DRAFT} [The `Com High Proxy` shall provide the functions for reception requests of a `ComSignal` and `ComSignalGroup`

- `Com_ReceiveSignal`
- `Com_ReceiveDynSignal`
- `Com_ReceiveSignalGroup`
- `Com_ReceiveSignalGroupArray`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`

- `Com_ReceiveSignalGroupWithMetaData`
- `Com_ReceiveSignalGroupArrayWithMetaData`

required by the Com users in the `Applicative Software Cluster`

](*SRS_SwCluC_00211*)

[SWS_SwCluC_02648]{DRAFT} [The function `Com_ReceiveSignalGroup` provided by the `Com High Proxy` shall

- invoke the `Com_ReceiveSignalGroupArray` API to fetch the data as uint8 array representation, AND
- perform a transformation on the uint8 array to output the original data which will be passed to the RTE in the `Applicative Software Cluster`.

](*SRS_SwCluC_00211*)

[SWS_SwCluC_02649]{DRAFT} [The function `Com_ReceiveSignalGroupWithMetaData` provided by the `Com High Proxy` shall

- invoke the `Com_ReceiveSignalGroupArrayWithMetaData` API to fetch the data as uint8 array representation, AND
- perform a transformation on the uint8 array to output the original data which will be passed to the RTE in the `Applicative Software Cluster`.

](*SRS_SwCluC_00211*)

[SWS_SwCluC_02611]{DRAFT} [The `Com High Proxy` shall provide the functions for invalidation requests of a `ComSignal` and `ComSignalGroup`

- `Com_InvalidateSignal`
- `Com_InvalidateSignalGroup`

required by the Com users in the `Applicative Software Cluster`

](*SRS_SwCluC_00211*)

[SWS_SwCluC_02612]{DRAFT} [If the `Com High Proxy` is not connected to the `Com Low Proxy`, the functions w.r.t transmission requests of `ComSignals` and `ComSignalGroups`

- `Com_SendSignal`
- `Com_SendDynSignal`
- `Com_SendSignalGroup`
- `Com_SendSignalGroupArray`
- `Com_SendSignalWithMetaData`
- `Com_SendDynSignalWithMetaData`

- `Com_SendSignalGroupWithMetaData`
- `Com_SendSignalGroupArrayWithMetaData`

of the `Com High Proxy` shall return `COM_SERVICE_NOT_AVAILABLE`.

](*SRS_SwCluC_00211*)

[SWS_SwCluC_02613]{DRAFT} [If the `Com High Proxy` is not connected to the `Com Low Proxy`, the functions w.r.t reception requests of `ComSignals` and `ComSignalGroups`

- `Com_ReceiveSignal`
- `Com_ReceiveDynSignal`
- `Com_ReceiveSignalGroup`
- `Com_ReceiveSignalGroupArray`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`
- `Com_ReceiveSignalGroupWithMetaData`
- `Com_ReceiveSignalGroupArrayWithMetaData`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`

of the `Com High Proxy` shall return `COM_SERVICE_NOT_AVAILABLE`.

](*SRS_SwCluC_00211*)

[SWS_SwCluC_02614]{DRAFT} [If the `Com High Proxy` is not connected to the `Com Low Proxy`, the functions w.r.t invalidation requests of `ComSignals` and `ComSignalGroups`

- `Com_InvalidateSignal`
- `Com_InvalidateSignalGroup`

of the `Com High Proxy` shall return `COM_SERVICE_NOT_AVAILABLE`.

](*SRS_SwCluC_00211*)

[SWS_SwCluC_02615]{DRAFT} [The `Com High Proxy` shall provide the functions for transmission request of a signal(group)s, according to [\[SWS_SwCluC_02609\]](#), once per configured `SwCluCComProxyBaseSocket`. At the `EcucPartition` where the `SwCluCNativeBswApi` is set to `true`, For all other `SwCluCComProxyBaseSocket` instances, the `Mip Com` is replaced by `Com_<EcucPartition.ShortName>`.](*SRS_SwCluC_00211*)

[SWS_SwCluC_02616]{DRAFT} [The [Com High Proxy](#) shall provide the functions for reception request of a signal(group)s, according to [\[SWS_SwCluC_02610\]](#), once per configured [SwCluCComProxyBaseSocket](#). At the [EcucPartition](#) where the [SwCluCNativeBswApi](#) is set to `true`, the API is provided with the original `Mip Com`. For all the other [SwCluCComProxyBaseSocket](#) instances, the `Mip Com` is replaced by `Com_<EcucPartition.ShortName>`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02617]{DRAFT} [The [Com High Proxy](#) shall provide the functions for invalidation request of a signal(group)s, according to [\[SWS_SwCluC_02611\]](#), once per configured [SwCluCComProxyBaseSocket](#). At the [EcucPartition](#) where the [SwCluCNativeBswApi](#) is set to `true`, the API is provided with the original `Mip Com`. For all the other [SwCluCComProxyBaseSocket](#) instances, the `Mip Com` is replaced by `Com_<EcucPartition.ShortName>`.] ([SRS_SwCluC_00211](#))

Callback Functions

[SWS_SwCluC_02618]{DRAFT} [The [Com High Proxy](#) shall support the optional `<ComUser_CbkRxAck>` notification callback configured via the [ComUserCallbackName](#) (See [\[ECUC_Com_10036\]](#)) parameter where the [ComUserCallbackType](#) (See [\[ECUC_Com_10034\]](#)) is set to `COM_RX_ACK`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02619]{DRAFT} [The [Com High Proxy](#) shall support the optional `<ComUser_CbkInv>` notification callback configured via the [ComUserCallbackName](#) (See [\[ECUC_Com_10036\]](#)) where the [ComUserCallbackType](#) (See [\[ECUC_Com_10034\]](#)) is set to `COM_RX_INV`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02620]{DRAFT} [The [Com High Proxy](#) shall support the optional `<ComUser_CbkRxTOut>` timeout notification callback configured via the [ComUserCallbackName](#) (See [\[ECUC_Com_10036\]](#)) where the [ComUserCallbackType](#) (See [\[ECUC_Com_10034\]](#)) is set to `COM_RX_TOUT`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02621]{DRAFT} [The [Com High Proxy](#) shall support the optional `<ComUser_CbkTxAck>` notification callback configured via the [ComUserCallbackName](#) (See [\[ECUC_Com_10036\]](#)) where the [ComUserCallbackType](#) (See [\[ECUC_Com_10034\]](#)) is set to `COM_TX_ACK`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02622]{DRAFT} [The [Com High Proxy](#) shall support the optional `<ComUser_CbkTxErr>` error notification callback configured via the [ComUserCallbackName](#) (See [\[ECUC_Com_10036\]](#)) where the [ComUserCallbackType](#) (See [\[ECUC_Com_10034\]](#)) is set to `COM_TX_ERR`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02623]{DRAFT} [The [Com High Proxy](#) shall support the optional `<ComUser_CbkTxTOut>` timeout notification callback configured via the [ComUserCallbackName](#) (See [\[ECUC_Com_10036\]](#)) where the [ComUserCallbackType](#) (See [\[ECUC_Com_10034\]](#)) is set to `COM_TX_TOUT`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02647]{DRAFT} [The [Com High Proxy](#) shall create for each [SwCluCComProxyBaseSocket](#) a required resource entry in the [Binary Manifest](#).] ([SRS_SwCluC_00211](#))

Please note that the callbacks of the [Applicative Software Cluster](#)

- <ComUser_CbkRxAck>
- <ComUser_CbkInv>
- <ComUser_CbkRxTOut>
- <ComUser_CbkTxAck>
- <ComUser_CbkTxErr>
- <ComUser_CbkTxTOut>

are connected to the [Host Software Cluster](#) via the [Com base socket](#). See [\[SWS_SwCluC_02661\]](#)

In the [Applicative Software Cluster](#), the relationship between a Signal or a Signal Group in Com and the related [Com base socket](#) is given indirectly, via the common [EcucPartition\(s\)](#) of the [SwCluCComProxyBaseSocket.SwCluCProxyEcucPartitionRef](#) and the [ComMainFunctionRx.ComMainRxPartitionRef](#) and [ComMainFunctionTx.ComMainTxPartitionRef](#) respectively. Those relationships are illustrated in figure 7.13.

The reason not to define a dedicated EcuC configuration structure for this purpose, is the assumption that the same tools are used to derive the Com configuration for an [Applicative Software Cluster](#) and a [Host Software Cluster](#).

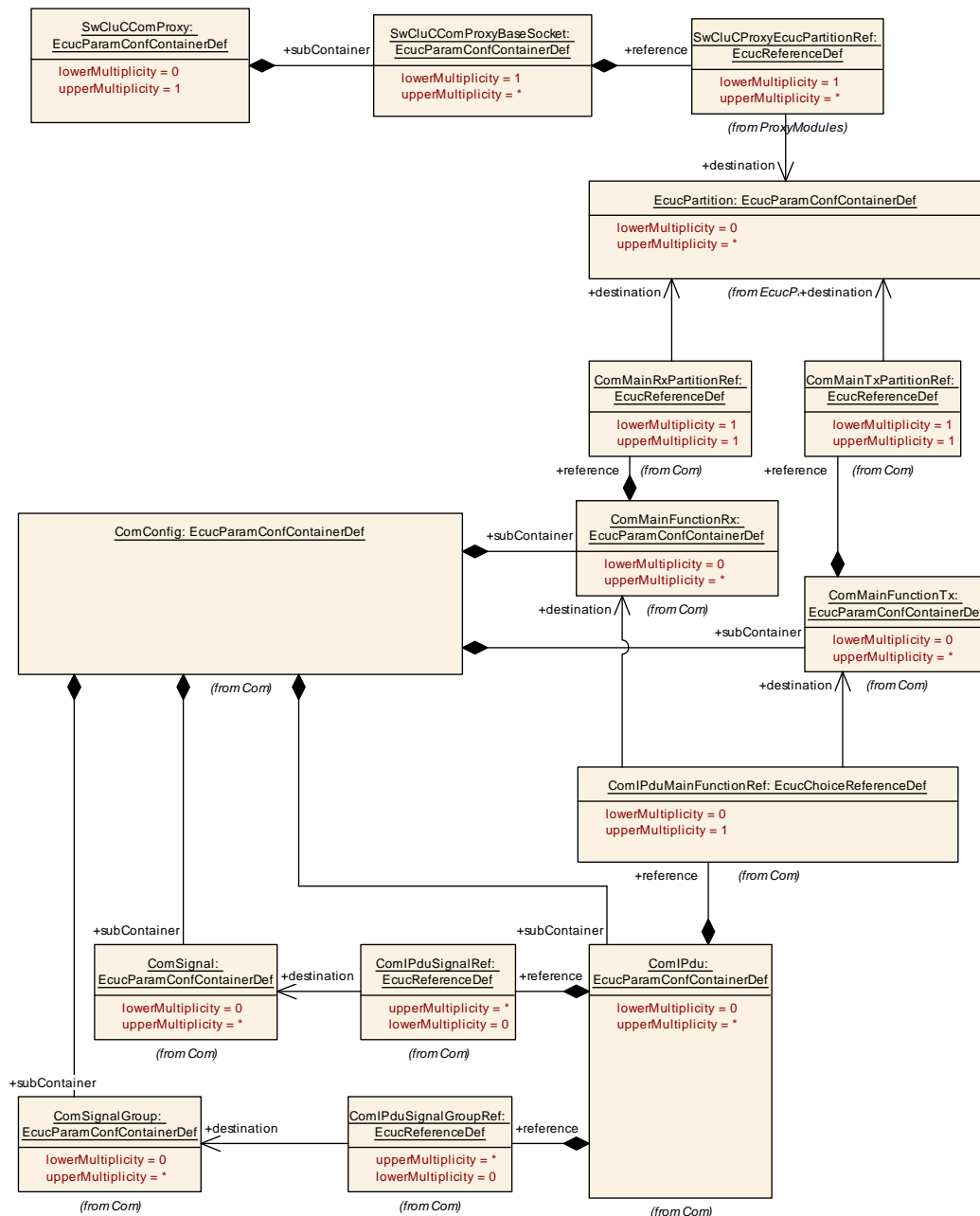


Figure 7.13: Signal or a Signal Group and their relationships to Com base sockets via EcucPartition

[SWS_SwCluC_02662]{DRAFT} [The `Com High Proxy` shall create for each `ComSignal` and `ComSignalGroup` configured in `Com` a required resource entry in the `Binary Manifest`. Where the `ComMainFunctionRx.ComMainRxPartitionRef` and `ComMainFunctionTx.ComMainTxPartitionRef` are relevant to denote the `EcucPartition`.] (*SRS_SwCluC_00211*)

7.4.3.3.6 Com Low Proxy

[SWS_SwCluC_02624]{DRAFT} [The `Com Low Proxy` shall provide the header file `SwCluC_ComProxy_Cbk.h`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_CONSTR_02654]{DRAFT} [The `ComUserHeaderInclude` parameter shall be configured as `SwCluC_ComProxy_Cbk.h`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02651]{DRAFT} [The `Com Low Proxy` shall provide in its `ComUserModuleCnf` containers the `ComUserCbkJd` values for the configured `ComUserSignals` and `ComUserSignalGroups`, according to its internal implementation requirements (e.g. to address the `Com Low Proxy`'s data structures).] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02625]{DRAFT} [The `Com Low Proxy` shall provide the entry function for transmission, reception and invalidation requests of `ComSignal(Group)s`, according to [\[SWS_SwCluC_02609\]](#), [\[SWS_SwCluC_02610\]](#) and [\[SWS_SwCluC_02611\]](#), once per configured `SwCluCComProxyBaseSocket`, at the `EcucPartition` configured by `SwCluCComProxyBaseSocket.SwCluCProxyEcucPartitionRef`.

In case the `SwCluCComProxyUsedSatelliteRef` is set, the `EcucPartition` shall be changed, before the Com API is called.] ([SRS_SwCluC_00211](#))

The implementation of the `Com Low Proxy` may provide some basic infrastructure for the `Com High Proxy` in order to be operational. Such generic infrastructure services (including the access to COM APIs or the callbacks) can be linked between `Applicative Software Cluster` and `Host Software Clusters` by means of a Com base socket.

[SWS_SwCluC_02626]{DRAFT} [The `Com Low Proxy` shall provide for each `SwCluCComProxyBaseSocket` a `SwCluC_ComProxy_Init` API, to initialize the `Com High Proxy` of an `Applicative Software Cluster`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02627]{DRAFT} [The `Com Low Proxy` shall provide an instance of all `SwCluC_ComProxy_CbkRxAck` notification callbacks, configured via the `ComUserCallbackName` (See [\[ECUC_Com_10036\]](#)) parameter, with the `ComUserCallbackType` (See [\[ECUC_Com_10034\]](#)) set to `COM_RX_ACK`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_CONSTR_02655]{DRAFT} [Each notification callback, where the parameter `ComUserCallbackType` (See [\[ECUC_Com_10034\]](#)) is set to `COM_RX_ACK`, shall be configured as `SwCluC_ComProxy_<BS>_CbkJd`, where `<BS>` is the `shortName` of the `SwCluCComProxyBaseSocket` container.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_02628]{DRAFT} [The `Com Low Proxy` shall provide an instance of all `SwCluC_ComProxy_COMCbkJd` notification callbacks, configured on each `ComSignal(Group)` via the `ComUserCallbackName` (See [\[ECUC_Com_10036\]](#)) parameter, with the `ComUserCallbackType` (See [\[ECUC_Com_10034\]](#)) set to `COM_RX_INV`.] ([SRS_SwCluC_00211](#))

[SWS_SwCluC_CONSTR_02656]{DRAFT} [Each notification callback, where the parameter `ComUserCallbackType` (See [ECUC_Com_10034]) is set to `COM_RX_INV`, shall be configured as `SwCluC_ComProxy_<BS>_CbkJnv`, where `<BS>` is the `shortName` of the `SwCluCComProxyBaseSocket` container.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_02629]{DRAFT} [The `Com Low Proxy` shall provide an instance of all `SwCluC_ComProxy_COMCbkJRxTOut` timeout notification callbacks configured on each `ComSignal(Group)`, via the `ComUserCallbackName` (See [ECUC_Com_10036]), with the `ComUserCallbackType` (See [ECUC_Com_10034]) set to `COM_RX_TOUT`.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_CONSTR_02657]{DRAFT} [Each notification callback where the parameter `ComUserCallbackType` (See [ECUC_Com_10034]) is set to `COM_RX_TOUT` shall be configured as `SwCluC_ComProxy_<BS>_CbkJRxTOut` where `<BS>` is the `shortName` of the `SwCluCComProxyBaseSocket` container.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_02630]{DRAFT} [The `Com High Proxy` shall provide an instance of all `<ComUser_CbkJTxAck>` notification callbacks, configured via the `ComUserCallbackName` (See [ECUC_Com_10036]), with the `ComUserCallbackType` (See [ECUC_Com_10034]) set to `COM_TX_ACK`.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_CONSTR_02658]{DRAFT} [Each notification callback, where the parameter `ComUserCallbackType` (See [ECUC_Com_10034]) is set to `COM_TX_ACK`, shall be configured as `SwCluC_ComProxy_<BS>_CbkJTxAck`, where `<BS>` is the `shortName` of the `SwCluCComProxyBaseSocket` container.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_02631]{DRAFT} [The `Com High Proxy` shall provide an instance of all `<ComUser_CbkJTxErr>` notification callbacks, configured via the `ComUserCallbackName` (See [ECUC_Com_10036]), with the `ComUserCallbackType` (See [ECUC_Com_10034]) set to `COM_TX_ERR`.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_CONSTR_02659]{DRAFT} [Each notification callback, where the parameter `ComUserCallbackType` (See [ECUC_Com_10034]) is set to `COM_TX_ERR`, shall be configured as `SwCluC_ComProxy_<BS>_CbkJTErr`, where `<BS>` is the `shortName` of the `SwCluCComProxyBaseSocket` container.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_02632]{DRAFT} [The `Com High Proxy` shall provide an instance of all `<ComUser_CbkJTxTOut>` notification callbacks, configured via the `ComUserCallbackName` (See [ECUC_Com_10036]), with the `ComUserCallbackType` (See [ECUC_Com_10034]) set to `COM_TX_TOUT`.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_CONSTR_02660]{DRAFT} [Each notification callback, where the parameter `ComUserCallbackType` (See [ECUC_Com_10034]) is set to `COM_TX_TOUT`, shall be configured as `SwCluC_ComProxy_<BS>_CbkJTxTOut`, where `<BS>` is the `shortName` of the `SwCluCComProxyBaseSocket` container.]([SRS_SwCluC_00211](#))

[SWS_SwCluC_02633]{DRAFT} [The notification callback instances according to [[SWS_SwCluC_02626](#)], [[SWS_SwCluC_02627](#)], [[SWS_SwCluC_02628](#)],

[SWS_SwCluC_02629], [SWS_SwCluC_02630], [SWS_SwCluC_02631] and [SWS_SwCluC_02632] are provided

- at the `EcucPartition` of the `SwCluCComProxyBaseSocket` referenced by the `SwCluCComProxyBaseSocket` - via `SwCluCComProxyUsedSateliteRef`. In this case, the `EcucPartition` shall be changed, before the call-back in the `High Proxy` is called.

](SRS_SwCluC_00211)

[SWS_SwCluC_02646]{DRAFT} [The `Com Low Proxy` shall create for each `SwCluCComProxyBaseSocket` a provided resource entry in the `Binary Manifest`.

](SRS_SwCluC_00211)

[SWS_SwCluC_02661]{DRAFT} [The `Com base socket` shall implement the connection of the related callback functions

- `SwCluC_ComProxy_CbkRxAck`
- `SwCluC_ComProxy_COMCbkInv`
- `SwCluC_ComProxy_COMCbkRxTOut`
- `SwCluC_ComProxy_COMCbkTAck`
- `SwCluC_ComProxy_COMCbkTErr`
- `SwCluC_ComProxy_COMCbkTxTOut`

](SRS_SwCluC_00211)

[SWS_SwCluC_02634]{DRAFT} [If the `Com Low Proxy` is not connected to the `Com High Proxy`, the callback functions

- `SwCluC_ComProxy_CbkRxAck`
- `SwCluC_ComProxy_COMCbkInv`
- `SwCluC_ComProxy_COMCbkRxTOut`
- `SwCluC_ComProxy_COMCbkTAck`
- `SwCluC_ComProxy_COMCbkTErr`
- `SwCluC_ComProxy_COMCbkTxTOut`

of the `Com Low Proxy` shall return without effect.](SRS_SwCluC_00211)

Conceptually, the `Com Proxy` provides `Signals` and `Signal Groups` with their corresponding notifications for a set of specific `EcucPartitions`. In the simplest case, this is a single `EcucPartition`, where also the `Com` module handles the underlying `ComIPdu`. In case the `Com` users as well as the `Com API` are unspecific to the `EcucPartition`, the `Com Proxy` can provide the same `Signals` and `Signal Groups` with their corresponding notifications for more than one `EcucPartition`. Nevertheless,

it's possible that the `Com Proxy` changes the partition for `Com API` access and notifications. But be aware that this may have a severe performance impact, if this is used for a larger number of `Signals` or `Signal Groups`.

In the `Host Software Cluster`, the set of `Signals` and `Signal Groups`, with their corresponding notifications, is configured by means of the `SwCluCComProxyBaseSocket.ComUserModuleCnf`. For those `Signals` and `Signal Groups`, the related entries in the `Binary Manifest` are created by the `Com Low Proxy`.

[SWS_SwCluC_02663]{DRAFT} [The `Com Low Proxy` shall create for each `ComUserSignal` and `ComUserSignalGroup` configured in a `SwCluCComProxyBaseSocket.ComUserModuleCnf` a provided resource entry in the `Binary Manifest`.] (*SRS_SwCluC_00211*)

7.4.3.3.7 Error Detection

[SWS_SwCluC_02636]{DRAFT} [If the development error detection is enabled for the `Com High Proxy`, the APIs

- `Com_SendSignal`
- `Com_SendDynSignal`
- `Com_SendSignalGroup`
- `Com_SendSignalGroupArray`
- `Com_SendSignalWithMetaData`
- `Com_SendDynSignalWithMetaData`
- `Com_SendSignalGroupWithMetaData`
- `Com_SendSignalGroupArrayWithMetaData`
- `Com_ReceiveSignal`
- `Com_ReceiveDynSignal`
- `Com_ReceiveSignalGroup`
- `Com_ReceiveSignalGroupArray`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`
- `Com_ReceiveSignalGroupWithMetaData`
- `Com_ReceiveSignalGroupArrayWithMetaData`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`

- `Com_InvalidateSignal`
- `Com_InvalidateSignalGroup`

shall report the DET error `COM_E_UNINIT`, in case the `Com High Proxy` is not initialized.]([SRS_SwCluC_00211](#), [SRS_BSW_00369](#), [SRS_BSW_00323](#))

[SWS_SwCluC_02637]{DRAFT} [If the development error detection is enabled for the `Com High Proxy`, the APIs

- `Com_SendSignal`
- `Com_SendDynSignal`
- `Com_SendSignalGroup`
- `Com_SendSignalGroupArray`
- `Com_SendSignalWithMetaData`
- `Com_SendDynSignalWithMetaData`
- `Com_SendSignalGroupWithMetaData`
- `Com_SendSignalGroupArrayWithMetaData`
- `Com_ReceiveSignal`
- `Com_ReceiveDynSignal`
- `Com_ReceiveSignalGroup`
- `Com_ReceiveSignalGroupArray`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`
- `Com_ReceiveSignalGroupWithMetaData`
- `Com_ReceiveSignalGroupArrayWithMetaData`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`
- `Com_InvalidateSignal`
- `Com_InvalidateSignalGroup`

shall, when invoked with a `NULL_PTR`, report the DET error `COM_E_PARAM_POINTER`.]([SRS_SwCluC_00211](#), [SRS_BSW_00369](#), [SRS_BSW_00323](#))

[SWS_SwCluC_02638]{DRAFT} [If the development error detection is enabled for the `Com High Proxy`, the APIs

- `Com_SendSignal`

- `Com_SendDynSignal`
- `Com_SendSignalGroup`
- `Com_SendSignalGroupArray`
- `Com_SendSignalWithMetaData`
- `Com_SendDynSignalWithMetaData`
- `Com_SendSignalGroupWithMetaData`
- `Com_SendSignalGroupArrayWithMetaData`
- `Com_ReceiveSignal`
- `Com_ReceiveDynSignal`
- `Com_ReceiveSignalGroup`
- `Com_ReceiveSignalGroupArray`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`
- `Com_ReceiveSignalGroupWithMetaData`
- `Com_ReceiveSignalGroupArrayWithMetaData`
- `Com_ReceiveSignalWithMetaData`
- `Com_ReceiveDynSignalWithMetaData`
- `Com_InvalidateSignal`
- `Com_InvalidateSignalGroup`

shall, when invoked with a wrong parameter, report the DET error `COM_E_PARAM.`]
([SRS_SwCluC_00211](#), [SRS_BSW_00369](#), [SRS_BSW_00323](#))

7.4.3.4 LdCom Proxy

7.4.3.4.1 Enable LdCom Proxy Generation

[SWS_SwCluC_02500]{DRAFT} [The `LdCom High Proxy` code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter `SwCluCProxyGenerationLdCom` is set to `HIGH_PROXY.`]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02501]{DRAFT} [The `LdCom Low Proxy` code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter `SwCluCProxyGenerationLdCom` is set to `LOW_PROXY.`]([SRS_SwCluC_00207](#))

7.4.3.4.2 General LdCom Proxy functionality

7.4.3.4.3 Overview

With the concept of Software Clusters, which enables the splitting of the software of an AUTOSAR Classic Platform Architecture into smaller units, the **Applicative Software Cluster** provides proxies for those parts of the LdCom implementation that are required to complete its functionality. This is called **LdCom High Proxy**.

The **Host Software Cluster** contains the real implementation of the LdCom functionality and makes it available to the **LdCom High Proxy** via the respective **LdCom Low Proxy**. Both **LdCom High Proxy** and **LdCom Low Proxy** are linked together based on the information available in **Binary Manifest**, which enables the execution of LdCom (Callback) functions from **Applicative Software Cluster** to **Host Software Cluster** and vice-versa.

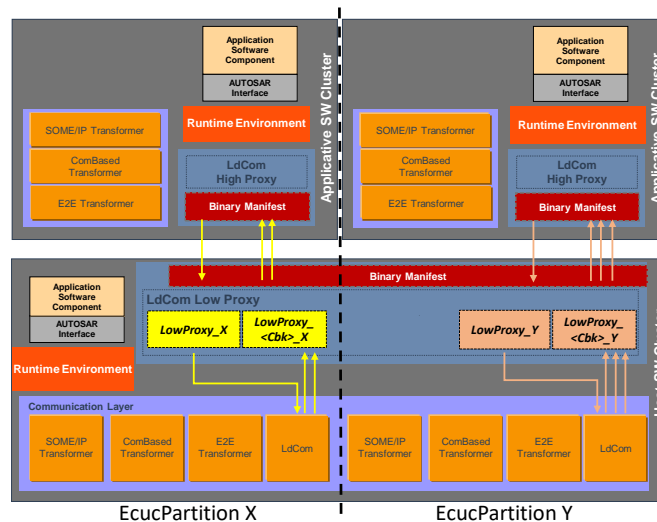


Figure 7.14: Overview of LdCom Proxy

The LdCom module can now have an arbitrary of users with the **LdCom Low Proxy** being one of those. It therefore relies on the usage of URI References (See [10], Section URI Reference) to be linked with its respective users in the model.

To guarantee the compatibility between configurations of the LdCom module and the LdCom Proxy, required parameters and containers are defined by the **LdComUserUriDefSet**. This means that the LdCom Proxy as a LdCom user shall configure **LdComUserModuleCnf** container (including its sub-containers), which holds the configuration of the **LdComIPdu**s it transmits and receives (via dedicated notification callbacks).

There are two alternatives, how the LdCom Proxy can be configured:

ECUC Partition specific LdCom Proxy Callbacks

Here, the LdCom Proxy as an LdComUserModule provides dedicated instances for each configured partition, on which LdCom (notification callback) invocations take place. This, however, mandates that it provides multiple main functions, each one bound to the relevant EcucPartition. The configured notification callbacks are invoked in the context of one EcucPartition only.

ECUC Partition agnostic LdCom Proxy Callbacks

Here, the LdCom Proxy provides partition independent notification callback instances, and therefore has to provide one common set of notification callbacks, which are invoked in the context of different EcucPartitions. Furthermore, it shall ensure a re-entrant implementation of the notification callbacks for different LdComIPdus on different EcucPartition.

The LdCom Proxy as an LdComUserModule is represented by one or several SwCluCLdComProxyBaseSockets, configured via a dedicated LdComUserModuleCnf. A SwCluCLdComProxyBaseSocket is required for each EcucPartition, in which the LdCom Proxy

- requires direct access to the LdCom APIs initiating transmission requests
- provides notification callbacks w.r.t transmission and reception

Effectively, a SwCluCLdComProxyBaseSocket links a fixed set of notification callbacks in the LdCom to a specific EcucPartition in the LdCom Proxy. As a consequence, the LdCom Proxy has to map each LdComUserIPdu via LdComUserSystemTemplateSignalRef to an LdComIPdu.

The LdCom shall provide its APIs for transmission requests of the relevant LdComIPdus on the EcucPartition configured in the LdCom Proxy. Even if the concept of LdCom user provides a lot of flexibility to support access by multiple users, including their notifications, some limitations need to be considered. In general, multiple writers can cause race conditions, if the writers are not coordinated. In addition, the behavior of <LdComUser_LdComCbkJTriggerTransmit> does not support multiple notified users for the same LdComIPdu. Since such a coordination across different Software Clusters is hardly achievable, the following restriction applies:

[SWS_SwCluC_CONSTR_02546]{DRAFT} [Sent LdComIPdu shall be provided to at most one *Applicative Software Cluster*.] (*SRS_SwCluC_00207*)

In addition, the LdCom Proxy does not support a fan out of LdComIPdus sent or received via transport protocol API.

[SWS_SwCluC_CONSTR_02547]{DRAFT} [LdComIPdus sent or received via transport protocol API shall be provided to at most one *Applicative Software Cluster*.] (*SRS_SwCluC_00207*)

Please note that the PduR supports a fan-out of TP-connections to upper layers.

7.4.3.4.4 General requirements

[SWS_SwCluC_02502]{DRAFT} [The LdCom Proxy shall support that the LdComIPdu Ids of the LdCom of the Host Software Cluster can change, without reconfiguration or rebuild of the Applicative Software Cluster using these LdComIPdus.](SRS_SwCluC_00207)

[SWS_SwCluC_02548]{DRAFT} [The LdCom Proxy shall support that the LdComUserCbkJHandleId of LdComUserIPdus configured in LdCom users of the Applicative Software Cluster can change, without reconfiguration or rebuild of the Host Software Cluster.](SRS_SwCluC_00211)

[SWS_SwCluC_02503]{DRAFT} [The LdCom Proxy shall support the invocation of LdCom API from an Applicative Software Cluster in context of the Host Software Cluster.](SRS_SwCluC_00207)

[SWS_SwCluC_02504]{DRAFT} [The LdCom Proxy shall support the invocation of LdCom Callback APIs from the Host Software Cluster in context of the Applicative Software Cluster.](SRS_SwCluC_00207)

[SWS_SwCluC_02535]{DRAFT} [If the <LdComUser_LdComCbkJRxIndication> notification callback is configured in the LdCom High Proxy (via the LdComUserCallbackName (See [ECUC_LdCom_00023]) parameter, where the LdComUserCallbackType (See [ECUC_LdCom_00025]) is set to LDCOM_RX_INDICATION), all the configured <LdComUser_LdComCbkJRxIndication> callbacks shall be invoked, when the associated instance of the SwCluC_LdComProxy_LdComCbkJRxIndication from the LdCom Low Proxy is called.](SRS_SwCluC_00207)

[SWS_SwCluC_02536]{DRAFT} [If the <LdComUser_LdComCbkJTxConfirmation> notification callback is configured in the LdCom High Proxy (via the LdComUserCallbackName (See [ECUC_LdCom_00023]) parameter, where the LdComUserCallbackType (See [ECUC_LdCom_00025]) is set to LDCOM_TX_CONFIRMATION), the configured <LdComUser_LdComCbkJTxConfirmation> callback shall be invoked. when the associated instance of the SwCluC_LdComProxy_LdComCbkJTxConfirmation from the LdCom Low Proxy is called.](SRS_SwCluC_00207)

[SWS_SwCluC_02537]{DRAFT} [If the <LdComUser_LdComCbkJTriggerTransmit> notification callback is configured in the LdCom High Proxy (via the LdComUserCallbackName (See [ECUC_LdCom_00023]) parameter, where the LdComUserCallbackType (See [ECUC_LdCom_00025]) is set to LDCOM_TX_TRIGGER_TRANSMIT), the configured <LdComUser_LdComCbkJTriggerTransmit> callback shall be invoked, when the associated instance of the SwCluC_LdComProxy_LdComCbkJTriggerTransmit from the LdCom Low Proxy is called.](SRS_SwCluC_00207)

[SWS_SwCluC_02538]{DRAFT} [If the <LdComUser_LdComCbkJCopyRxData> notification callback is configured in the LdCom High Proxy (via the LdComUserCallbackName (See [ECUC_LdCom_00023]) parameter, where the LdComUserCallbackType (See [ECUC_LdCom_00025]) is set to LDCOM_TP_COPY_RX_DATA), the

configured `<LdComUser_LdComCbkJCopyRxData>` callback shall be invoked, when the associated instance of the `SwCluC_LdComProxy_LdComCbkJCopyRxData` from the `LdCom Low Proxy` is called.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02539]{DRAFT} [If the `<LdComUser_LdComCbkJStartOfReception>` notification callback is configured in the `LdCom High Proxy` (via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_RX_START_OF_RECEPTION`), the configured `<LdComUser_LdComCbkJStartOfReception>` callback shall be invoked, when the associated instance of the `SwCluC_LdComProxy_LdComCbkJStartOfReception` from the `LdCom Low Proxy` is called.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02540]{DRAFT} [If the `<LdComUser_LdComCbkJTpRxIndication>` notification callback is configured in the `LdCom High Proxy` (via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_RX_INDICATION`), the configured `<LdComUser_LdComCbkJTpRxIndication>` callback shall be invoked, when the associated instance of the `SwCluC_LdComProxy_LdComCbkJTpRxIndication` from the `LdCom Low Proxy` is called.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02541]{DRAFT} [If the `<LdComUser_LdComCbkJTpTxConfirmation>` notification callback is configured in the `LdCom High Proxy` (via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_TX_CONFIRMATION`), the configured `<LdComUser_LdComCbkJTpTxConfirmation>` callback shall be invoked, when the associated instance of the `SwCluC_LdComProxy_LdComCbkJTpTxConfirmation` from the `LdCom Low Proxy` is called.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02542]{DRAFT} [If the `<LdComUser_LdComCbkJCopyTxData>` notification callback is configured in the `LdCom High Proxy` (via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_COPY_TX_DATA`), the configured `<LdComUser_LdComCbkJCopyTxData>` callback shall be invoked, when the associated instance of the `SwCluC_LdComProxy_LdComCbkJCopyTxData` from the `LdCom Low Proxy` is called.]([SRS_SwCluC_00207](#))

7.4.3.4.5 LdCom High Proxy

[SWS_SwCluC_02505]{DRAFT} [The `LdCom High Proxy` shall provide the header file `LdCom.h`.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02506]{DRAFT} [The `LdCom High Proxy` shall provide the symbolic name values for the `LdComIPdu` IDs, for `LdCom` users in the `Applicative Software Cluster`.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02507]{DRAFT} [The `LdCom High Proxy` shall provide the function w.r.t transmission request of a `LdComIPdu`

- `LdCom_Transmit`

required by the `Applicative Software Cluster`

]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02508]{DRAFT} [If the `LdCom High Proxy` is not connected to the `LdCom Low Proxy`, the function

- `LdCom_Transmit`

of the `LdCom High Proxy` shall return `E_NOT_OK`.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02509]{DRAFT} [The `LdCom High Proxy` shall provide the function for transmission request of a signal, according to [[SWS_SwCluC_02506](#)], once per configured `SwCluCLdComProxyBaseSocket`. At the `EcucPartition`, where the `SwCluCNativeBswApi` is set to `true`, the API is provided with the original `Mip LdCom`. For all other `SwCluCLdComProxyBaseSocket` instances, the `Mip LdCom` is replaced by `LdCom_<EcucPartition.ShortName>`.]([SRS_SwCluC_00207](#))

Callback Functions

[SWS_SwCluC_02510]{DRAFT} [The `LdCom High Proxy` shall support the `<LdComUser_LdComCbkJRxIndication>` notification callback, configured via the `LdComUserCallbackName` (See [[ECUC_LdCom_00023](#)]) parameter, where the `LdComUserCallbackType` (See [[ECUC_LdCom_00025](#)]) is set to `LDCOM_RX_INDICATION`]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02511]{DRAFT} [The `LdCom High Proxy` shall support the optional `<LdComUser_LdComCbkJTxConfirmation>` notification callback, configured via the `LdComUserCallbackName` (See [[ECUC_LdCom_00023](#)]) parameter, where the `LdComUserCallbackType` (See [[ECUC_LdCom_00025](#)]) is set to `LDCOM_TX_CONFIRMATION`]([SRS_SwCluC_00207](#))

Please note that only a single `LdCom` user can be notified with `<LdComUser_LdComCbkJTxConfirmation>`.

[SWS_SwCluC_02512]{DRAFT} [The `LdCom High Proxy` shall support the optional `<LdComUser_LdComCbkJTriggerTransmit>` notification callback, configured via the `LdComUserCallbackName` (See [[ECUC_LdCom_00023](#)]) parameter, where the `LdComUserCallbackType` (See [[ECUC_LdCom_00025](#)]) is set to `LDCOM_TX_TRIGGER_TRANSMIT`]([SRS_SwCluC_00207](#))

Please note that only a single `LdCom` user can be notified with `<LdComUser_LdComCbkJTriggerTransmit>`.

[SWS_SwCluC_02513]{DRAFT} [The [LdCom High Proxy](#) shall support the `<LdComUser_LdComCbkJCopyRxData>` notification callback, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_COPY_RX_DATA`] ([SRS_SwCluC_00207](#))

Please note that only a single LdCom user can be notified with `<LdComUser_LdComCbkJCopyRxData>`.

[SWS_SwCluC_02514]{DRAFT} [The [LdCom High Proxy](#) shall support the `<LdComUser_LdComCbkJStartOfReception>` notification callback configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_RX_START_OF_RECEPTION`] ([SRS_SwCluC_00207](#))

Please note that only a single LdCom user can be notified with `<LdComUser_LdComCbkJStartOfReception>`.

[SWS_SwCluC_02515]{DRAFT} [The [LdCom High Proxy](#) shall support the `<LdComUser_LdComCbkJTpRxIndication>` notification callback, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_RX_INDICATION`] ([SRS_SwCluC_00207](#))

Please note that only a single LdCom user can be notified with `<LdComUser_LdComCbkJTpRxIndication>`.

[SWS_SwCluC_02516]{DRAFT} [The [LdCom High Proxy](#) shall support the `<LdComUser_LdComCbkJTpTxConfirmation>` notification callback, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_TX_CONFIRMATION`] ([SRS_SwCluC_00207](#))

Please note that only a single LdCom user can be notified with `<LdComUser_LdComCbkJTpTxConfirmation>`.

[SWS_SwCluC_02517]{DRAFT} [The [LdCom High Proxy](#) shall support the `<LdComUser_LdComCbkJCopyTxData>` notification callback, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, where the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_COPY_TX_DATA`] ([SRS_SwCluC_00207](#))

Please note that only a single LdCom user can be notified with `<LdComUser_LdComCbkJCopyTxData>`.

[SWS_SwCluC_02543]{DRAFT} [The [Com High Proxy](#) shall create, for each `SwCluC_LdComProxyBaseSocket`, a required resource entry in the [Binary Manifest](#).] ([SRS_SwCluC_00207](#))

Please note that the callbacks of the [Applicative Software Cluster](#)

- `<LdComUser_LdComCbkJStartOfReception>`

- <LdComUser_LdComCbkJCopyRxData>
- <LdComUser_LdComCbkJCopyTxData>
- <LdComUser_LdComCbkJTriggerTransmit>
- <LdComUser_LdComCbkJRxIndication>
- <LdComUser_LdComCbkJTpRxIndication>
- <LdComUser_LdComCbkJTxConfirmation>
- <LdComUser_LdComCbkJTpTxConfirmation>

are connected to the [Host Software Cluster](#) via the [LdCom base socket](#). See [\[SWS_SwCluC_02550\]](#)

In the [Applicative Software Cluster](#), the relationship between a [LdComUserIPdu](#) and the related [Base Socket](#) is given indirectly, via the common [EcucPartition\(s\)](#) of the [SwCluCProxyBaseSocket.SwCluCProxyEcucPartitionRef](#) and the [EcucPartition](#) of the [LdComIPdu](#)'s referenced [Pdu](#). Where the [EcucPartition](#) of the [Pdu](#) is given via [Pdu.EcucPduDefaultPartitionRef](#) or [Pdu.EcucPduDedicatedPartition.EcucPduDedicatedPartitionRef](#), where [EcucPduDedicatedPartition](#) applies for [LdCom](#) configuration. Those relationships are illustrated in [figure 7.15](#). The reason not to define a dedicated [EcuC](#) configuration structure for this purpose, is the assumption that the same tools are used to derive the [LdCom](#) configuration for an [Applicative Software Cluster](#) and a [Host Software Cluster](#).

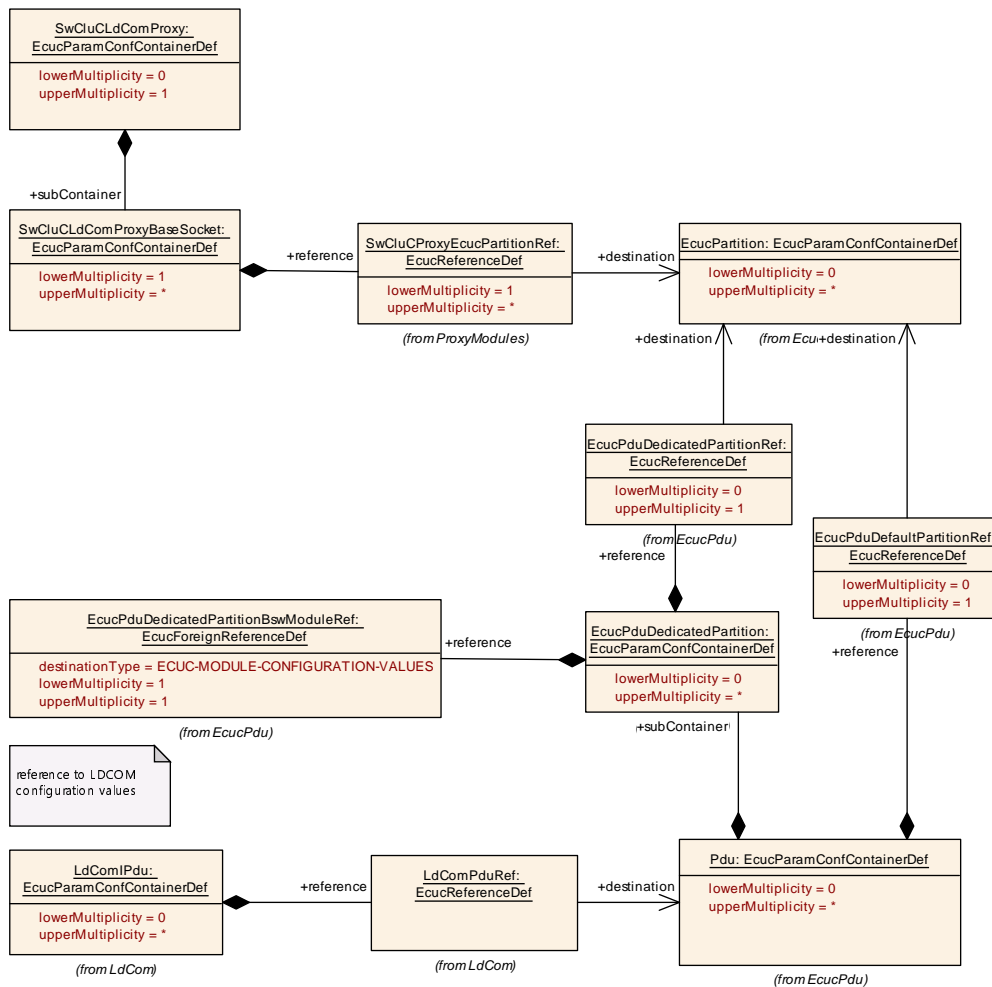


Figure 7.15: LdComIPdus and their relationships to LdCom base sockets via EcucPartition

[SWS_SwCluC_02551]{DRAFT} [The LdCom High Proxy shall create for each LdComIPdu configured in LdCom a required resource entry in the Binary Manifest. Where the EcucPartition of the LdComIPdu’s Pdu given via Pdu.EcucPduDefaultPartitionRef or Pdu.EcucPduDedicatedPartition.EcucPduDedicatedPartitionRef are relevant to denote the EcucPartition.](SRS_SwCluC_00207)

7.4.3.4.6 LdCom Low Proxy

[SWS_SwCluC_02518]{DRAFT} [The LdCom Low Proxy shall provide the header file SwCluC_LdComProxy_Cbk.h.](SRS_SwCluC_00207)

[SWS_SwCluC_CONSTR_02554]{DRAFT} [The LdComUserHeaderInclude parameter shall be configured as SwCluC_LdComProxy_Cbk.h.](SRS_SwCluC_00211)

[SWS_SwCluC_02549]{DRAFT} [The `LdCom Low Proxy` shall provide in its `LdComUserModuleCnf` containers the `LdComUserCbkJHandleId` values for the configured `LdComUserIPdus`, according to its internal implementation requirements (e.g. to address the `LdCom Low Proxy`'s data structures).] (*SRS_SwCluC_00211*)

The implementation of the `LdCom Low Proxy` may provide some basic infrastructure for the `LdCom High Proxy` in order to be operational. Such generic infrastructure services (including the access to `LdCom` APIs or the callbacks) can be linked between `Applicative Software Cluster` and `Host Software Clusters` by means of a `LdCom` base socket.

[SWS_SwCluC_02519]{DRAFT} [The `LdCom Low Proxy` shall provide the entry function for transmission requests, according to [\[SWS_SwCluC_02506\]](#), once per configured `SwCluC_LdComProxyBaseSocket` at the `EcucPartition` configured by `SwCluC_LdComProxyBaseSocket.SwCluCProxyEcucPartitionRef`.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_02520]{DRAFT} [The `LdCom Low Proxy` shall provide for each `SwCluC_LdComProxyBaseSocket` a `SwCluC_LdComProxy_Init` API, to initialize the `LdCom High Proxy` of an `Applicative Software Cluster`.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_02521]{DRAFT} [The `LdCom Low Proxy` shall provide an instance of all `SwCluC_LdComProxy_LdComCbkJRxIndication` notification callbacks, configured via the `LdComUserCallbackName` (See [\[ECUC_LdCom_00023\]](#)) parameter, with `LdComUserCallbackType` (See [\[ECUC_LdCom_00025\]](#)) set to `LDCOM_RX_INDICATION`.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_CONSTR_02555]{DRAFT} [Each `SwCluC_LdComProxy_LdComCbkJRxIndication` notification callback, where the parameter `LdComUserCallbackType` (See [\[ECUC_LdCom_00025\]](#)) is set to `LDCOM_RX_INDICATION`, shall be configured as `SwCluC_LdComProxy_<BS>_LdComCbkJRxIndication`, where `<BS>` is the `shortName` of the `SwCluC_LdComProxyBaseSocket` container.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_02522]{DRAFT} [The `LdCom Low Proxy` shall provide an instance of all `LdComCbkJTxConfirmation` notification callbacks, configured via the `LdComUserCallbackName` (See [\[ECUC_LdCom_00023\]](#)) parameter, with `LdComUserCallbackType` (See [\[ECUC_LdCom_00025\]](#)) set to `LDCOM_TX_CONFIRMATION`.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_CONSTR_02556]{DRAFT} [Each `SwCluC_LdComProxy_LdComCbkJTxConfirmation` notification callback, where the parameter `LdComUserCallbackType` (See [\[ECUC_LdCom_00025\]](#)) is set to `LDCOM_TX_CONFIRMATION`, shall be configured as `SwCluC_LdComProxy_<BS>_LdComCbkJTxConfirmation`, where `<BS>` is the `shortName` of the `SwCluC_LdComProxyBaseSocket` container.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_02523]{DRAFT} [The `LdCom Low Proxy` shall provide an instance of all `LdComCbkJTriggerTransmit` notification callbacks, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, with `LdComUserCallbackType` (See [ECUC_LdCom_00025]) set to `LDCOM_TX_TRIGGER_TRANSMIT`.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_CONSTR_02557]{DRAFT} [Each `SwCluC_LdComProxy_LdComCbkJTriggerTransmit` notification callback, where the parameter `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TX_TRIGGER_TRANSMIT`, shall be configured as `SwCluC_LdComProxy_<BS>_LdComCbkJTriggerTransmit`, where `<BS>` is the `shortName` of the `SwCluCLdComProxyBaseSocket` container.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_02524]{DRAFT} [The `LdCom Low Proxy` shall provide an instance of all `LdComCbkJCopyRxData` notification callbacks, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, with `LdComUserCallbackType` (See [ECUC_LdCom_00025]) set to `LDCOM_TP_COPY_RX_DATA`.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_CONSTR_02558]{DRAFT} [Each `SwCluC_LdComProxy_LdComCbkJCopyRxData` notification callback, where the parameter `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_COPY_RX_DATA`, shall be configured as `SwCluC_LdComProxy_<BS>_LdComCbkJCopyRxData`, where `<BS>` is the `shortName` of the `SwCluCLdComProxyBaseSocket` container.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_02525]{DRAFT} [The `LdCom Low Proxy` shall provide an instance of all `LdComCbkJStartOfReception` notification callbacks, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, with `LdComUserCallbackType` (See [ECUC_LdCom_00025]) set to `LDCOM_RX_START_OF_RECEPTION`.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_CONSTR_02559]{DRAFT} [Each `SwCluC_LdComProxy_LdComCbkJStartOfReception` notification callback, where the parameter `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_COPY_RX_DATA`, shall be configured as `SwCluC_LdComProxy_<BS>_LdComCbkJCopyRxData`, where `<BS>` is the `shortName` of the `SwCluCLdComProxyBaseSocket` container.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_02526]{DRAFT} [The `LdCom Low Proxy` shall provide an instance of all `LdComCbkJTpRxIndication` notification callbacks, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, with `LdComUserCallbackType` (See [ECUC_LdCom_00025]) set to `LDCOM_TP_RX_INDICATION`.] (*SRS_SwCluC_00207*)

[SWS_SwCluC_CONSTR_02560]{DRAFT} [Each `SwCluC_LdComProxy_LdComCbkJTpRxIndication` notification callback, where the parameter `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_RX_INDICATION`, shall be configured as `SwCluC_LdComProxy_<BS>_LdComCbkJTpRxIndication`, where

<BS> is the `shortName` of the `SwCluCLdComProxyBaseSocket` container.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02527]{DRAFT} [The `LdCom Low Proxy` shall provide an instance of all `LdComCbkJpTxConfirmation` notification callbacks, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, with the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) set to `LDCOM_TP_TX_CONFIRMATION`.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_CONSTR_02561]{DRAFT} [Each `SwCluC_LdComProxy_LdComCbkJpTxConfirmation` notification callback, where the parameter `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_TX_CONFIRMATION`, shall be configured as `SwCluC_LdComProxy_<BS>_LdComCbkJpTxConfirmation`, where <BS> is the `shortName` of the `SwCluCLdComProxyBaseSocket` container.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02528]{DRAFT} [The `LdCom Low Proxy` shall provide an instance of all `SwCluC_LdComProxy_LdComCbkJCopyTxData` notification callbacks, configured via the `LdComUserCallbackName` (See [ECUC_LdCom_00023]) parameter, with the `LdComUserCallbackType` (See [ECUC_LdCom_00025]) set to `LDCOM_TP_COPY_TX_DATA`.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_CONSTR_02562]{DRAFT} [Each `SwCluC_LdComProxy_LdComCbkJCopyTxData` notification callback, where the parameter `LdComUserCallbackType` (See [ECUC_LdCom_00025]) is set to `LDCOM_TP_COPY_TX_DATA`, shall be configured as `SwCluC_LdComProxy_<BS>_LdComCbkJCopyTxData`, where <BS> is the `shortName` of the `SwCluCLdComProxyBaseSocket` container.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02529]{DRAFT} [The notification callback instances according to [[SWS_SwCluC_02520](#)], [[SWS_SwCluC_02521](#)], [[SWS_SwCluC_02522](#)], [[SWS_SwCluC_02523](#)], [[SWS_SwCluC_02524](#)], [[SWS_SwCluC_02525](#)], [[SWS_SwCluC_02526](#)], [[SWS_SwCluC_02527](#)] and [[SWS_SwCluC_02528](#)] are provided

- at the `EcucPartition` of the `SwCluCLdComProxyBaseSocket`, referenced by the `SwCluCLdComProxyBaseSocket` - via `SwCluCLdComProxyUsedSatelliteRef`.

]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02544]{DRAFT} [The `LdCom Low Proxy` shall create for each `SwCluCLdComProxyBaseSocket` a provided resource entry in the `Binary Manifest`.]([SRS_SwCluC_00207](#))

[SWS_SwCluC_02550]{DRAFT} [The `LdCom base socket` shall implement the connection of the related callback functions

- `SwCluC_LdComProxy_LdComCbkJStartOfReception`

- SwCluC_LdComProxy_LdComCbkJCopyRxData
- SwCluC_LdComProxy_LdComCbkJCopyTxData
- SwCluC_LdComProxy_LdComCbkJTriggerTransmit
- SwCluC_LdComProxy_LdComCbkJRxIndication
- SwCluC_LdComProxy_LdComCbkJTpRxIndication
- SwCluC_LdComProxy_LdComCbkJTxConfirmation
- SwCluC_LdComProxy_LdComCbkJTpTxConfirmation

](SRS_SwCluC_00207)

[SWS_SwCluC_02530]{DRAFT} [If the LdCom Low Proxy is not connected to the LdCom High Proxy, the notification callback functions

- SwCluC_LdComProxy_LdComCbkJStartOfReception
- SwCluC_LdComProxy_LdComCbkJCopyRxData
- SwCluC_LdComProxy_LdComCbkJCopyTxData

of the LdCom Low Proxy shall return BUFREQ_E_NOT_OK.](SRS_SwCluC_00207)

[SWS_SwCluC_02531]{DRAFT} [If the LdCom Low Proxy is not connected to the LdCom High Proxy, the notification callback function

- SwCluC_LdComProxy_LdComCbkJTriggerTransmit

of the LdCom Low Proxy shall return E_NOT_OK.](SRS_SwCluC_00207)

[SWS_SwCluC_02532]{DRAFT} [If the LdCom Low Proxy is not connected to the LdCom High Proxy, the notification callback functions

- SwCluC_LdComProxy_LdComCbkJRxIndication
- SwCluC_LdComProxy_LdComCbkJTpRxIndication
- SwCluC_LdComProxy_LdComCbkJTxConfirmation
- SwCluC_LdComProxy_LdComCbkJTpTxConfirmation

of the LdCom Low Proxy shall return without effect.](SRS_SwCluC_00207)

Conceptually, the LdCom Proxy provides LdComIPdus with their corresponding notifications for a set of specific EcucPartitions. In the simplest case, this is a single EcucPartition, where also the LdCom module handles the underlying LdComIPdu. In case the LdCom users as well as the LdCom API are unspecific to the EcucPartitions, the LdCom Proxy provides the same LdComIPdu on multiple EcucPartitions. Nevertheless, it is possible that the LdCom Proxy changes the partition. But be aware that this may have a severe performance impact, if this is used for a larger number of LdComIPdus.

In the *Host Software Cluster*, the set of *LdComIPdus* with their corresponding notifications is configured by means of the *SwCluCldComProxyBaseSocket.LdComUserModuleCnf*. For those *LdComIPdus*, the related entries in the *Binary Manifest* are created by the *LdCom Low Proxy*.

[SWS_SwCluC_02552]{DRAFT} [The *LdCom Low Proxy* shall create, for each *LdComUserIPdu* configured in a *SwCluCldComProxyBaseSocket.LdComUserModuleCnf*, a provided resource entry in the *Binary Manifest*.] (*SRS_SwCluC_00207*)

7.4.3.4.7 Error Detection

[SWS_SwCluC_02533]{DRAFT} [If the development error detection is enabled for the *LdCom High Proxy*, the API

- *LdCom_Transmit*

shall report the DET error `LDCOM_E_UNINIT`, in case the *LdCom High Proxy* is not initialized.] (*SRS_SwCluC_00207*, *SRS_BSW_00369*, *SRS_BSW_00323*)

[SWS_SwCluC_02534]{DRAFT} [If the development error detection is enabled for the *LdCom High Proxy*, the API

- *LdCom_Transmit*

shall, when invoked with a `NULL_PTR`, report the DET error `LDCOM_E_PARAM_POINTER`.

] (*SRS_SwCluC_00207*, *SRS_BSW_00369*, *SRS_BSW_00323*)

7.4.3.5 Dcm Proxy

The specified *Dcm Proxy* has the following underlying design principle:

For diagnostic services accessing data of the ECU's applications, the *Dcm* requires interfaces to the Software Components, to read or write data values or to control specific functionality. Those interfaces need to be provided by the Software Components that are implementing the regular functionality of the ECU. Therefore, the *Dcm Proxy* provides the functionality to route these interfaces into the *Applicative SW Cluster*. The protocol, session and authentication handling stay in the *Dcm* of the *Host Software Cluster*. The configuration whether a sub-service is available at all, including its properties (e.g. a data size or the required diagnostic session), is still located in the *Dcm*.

For some diagnostic services, the *Dcm* offers a wide configurability with regard to how the interface to the software components is exactly implemented. E.g., the access to a diagnostic data can be done either with a client server interface for synchronous

access, or a client server interface for asynchronous access, or a sender receiver interface. Furthermore, additional sub-options are possible.

To avoid configuration dependencies from the [Host Software Cluster](#) to detailed interface decisions in the [Applicative Software Clusters](#), the Dcm Proxy uses a more generic interface to connect to the Dcm. The configuration of interface specialization is done in the [Applicative Software Clusters](#).

In general, this specification refers to the Standardized AUTOSAR Interfaces of the Dcm

7.4.3.5.1 Enable Dcm Proxy Generation

[SWS_SwCluC_02338]{DRAFT} [The [Dcm High Proxy](#) code, and related AUTOSAR model descriptions, shall only be created if the configuration parameter [SwCluCProxyGenerationDcm](#) is set to [HIGH_PROXY](#).] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02339]{DRAFT} [The [Dcm Low Proxy](#) code, and related AUTOSAR model descriptions, shall only be created if the configuration parameter [SwCluCProxyGenerationDcm](#) is set to [LOW_PROXY](#).] ([SRS_SwCluC_00208](#))

7.4.3.5.2 Dcm High Proxy

[SWS_SwCluC_02300]{DRAFT} [The [Dcm High Proxy](#) shall provide the header file [Dcm.h](#).] ([SRS_SwCluC_00208](#))

Multiple Partitions

[SWS_SwCluC_02341]{DRAFT} [The [Dcm High Proxy](#) shall create one partition specific Service Software Component per configured [EcucPartition](#), with the name [Dcm_<EcucPartition shortName>](#).] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02342]{DRAFT} [The [Dcm High Proxy](#) shall provide the Ports belonging to a specific interface connection at the partition specific Service Software Component, at the given [EcucPartition](#) of the proxy channel, configured via

- [SwCluCDcmProxyDiagnosticData.SwCluCProxyEcucPartitionRef](#)
- OR
- [SwCluCDcmProxyRoutineControl.SwCluCProxyEcucPartitionRef](#)
- OR
- [SwCluCDcmProxyServiceRequestNotification.SwCluCProxyEcucPartitionRef](#)

] ([SRS_SwCluC_00208](#))

Binary Manifest

The implementation of the [Dcm High Proxy](#) may require some basic infrastructure from the [Dcm Low Proxy](#) in order to be operational. Such generic infrastructure services can be linked between [Applicative Software Cluster](#) and [Host Software Cluster](#) by means of a `Dcm` base socket.

[SWS_SwCluC_02390]{DRAFT} [The [Dcm High Proxy](#) shall create for each [SwCluCdcmProxyBaseSocket](#) a required resource entry in the [Binary Manifest](#).]()

7.4.3.5.2.1 Diagnostic Data proxy

The diagnostic data proxy provides access to diagnostic data in Application Software Clusters, which can be accessed by diagnostic services via the `Dcm` in the [Host Software Cluster](#).

Service ID	Service name
UDS 0x22	ReadDataByIdentifier
UDS 0x24	ReadScalingDataByIdentifier
UDS 0x2E	WriteDataByIdentifier
UDS 0x2F	InputOutputControlByIdentifier
OBD \$01	Request Current Powertrain Diagnostic Data
OBD \$09	Request Vehicle Information

Table 7.3: Diagnostic Services for diagnostic data proxy

In addition, the [diagnostic data proxy](#) is also used by the `Dem` to access diagnostic data for freeze frame capturing.

DataServices_{Data} ports

The `DataServices_{Data}` ports are used to access single diagnostic data elements.

[SWS_SwCluC_02301]{DRAFT} [The [Dcm High Proxy](#) shall support the configuration of `DataServices_{Data}` [RPortPrototypes](#) for synchronous [ClientServerInterface](#) `DataServices_{Data}` for a [SwCluCdcmProxyDiagnosticData](#) referencing via [SwCluCdcmProxyDataRef](#) a [DcmDspData](#) or a [DcmDspPidService01](#).] ([SRS_SwCluC_00208](#))

Note: In the corresponding `Dcm` configuration, [DcmDspDataUsePort](#) will usually be set to `USE_DATA_SYNCH_CLIENT_SERVER`.

[SWS_SwCluC_02302]{DRAFT} [The [Dcm High Proxy](#) shall support the configuration of `DataServices_{Data}` [RPortPrototypes](#) for asynchronous [ClientServerInterface](#) `DataServices_{Data}` without parameter `ErrorCode`.] ([SRS_SwCluC_00208](#))

Note: In the corresponding `Dcm` configuration, [DcmDspDataUsePort](#) will usually be set to `USE_DATA_ASYNC_CLIENT_SERVER`.

[SWS_SwCluC_02303]{DRAFT} [The `Dcm High Proxy` shall support the configuration of `DataServices_{Data} RPortPrototypes` for asynchronous `ClientServerInterface DataServices_{Data}` with parameter `ErrorCode`.]
(*SRS_SwCluC_00208*)

Note: In the corresponding `Dcm` configuration, `DcmDspDataUsePort` will usually be set to `USE_DATA_ASYNC_CLIENT_SERVER_ERROR`.

[SWS_SwCluC_02304]{DRAFT} [In case the `SwCluCdcmProxyDiagnosticData` channel related to a `DataServices_{Data} PortPrototypes` is connected, the `Dcm Proxy` shall invoke the

- `ReadData operation`, when the corresponding `Xxx_ReadData API` is called in the `Dcm Low Proxy`,
OR
- `WriteData operation`, when the corresponding `Xxx_WriteData API` is called in the `Dcm Low Proxy`,
OR
- `ReadDataLength operation`, when the corresponding `Xxx_ReadDataLength API` is called in the `Dcm Low Proxy`,
OR
- `ConditionCheckRead operation`, when the corresponding `Xxx_ConditionCheckRead API` is called in the `Dcm Low Proxy`,
OR
- `GetScalingInformation operation`, when the corresponding `Xxx_GetScalingInformation API` is called in the `Dcm Low Proxy`,
OR
- `ReturnControlToEcu operation`, when the corresponding `Xxx_ReturnControlToEcu API` is called in the `Dcm Low Proxy`,
OR
- `ResetToDefault operation`, when the corresponding `Xxx_ResetToDefault API` is called in the `Dcm Low Proxy`,
OR
- `FreezeCurrentState operation`, when the corresponding `Xxx_FreezeCurrentState API` is called in the `Dcm Low Proxy`,
OR
- `ShortTermAdjustment operation`, when the corresponding `Xxx_ShortTermAdjustment API` is called in the `Dcm Low Proxy`.

In that case, IN arguments (Data (if applicable), OpStatus (if applicable), controlMask (if applicable) and ControlStateInfo (if applicable)) are passed to the client, and OUT arguments (Data (if applicable), ScalingInfo (if applicable), ErrorCode (if applicable)) and application errors are transferred back to the caller of the [Dcm Low Proxy](#).] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02305]{DRAFT} [In case the [operations](#) in the [DataServices_{Data} ClientServerInterface](#) do not have a [controlMask](#) argument configured, the value is discarded without checking the passed [controlMaskLength](#).] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02306]{DRAFT} [In case the [operations](#) in the [DataServices_{Data} ClientServerInterface](#) do not have an [ErrorCode](#) argument configured, the default value [DCM_POS_RESP](#) is transferred back to the caller of the [Dcm Low Proxy](#) in case the caller returned [E_OK](#).] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02307]{DRAFT} [In case the [DataServices_{Data} RPortPrototype](#) is unconnected, the application error [E_NOT_OK](#) and the [ErrorCode](#) [DCM_E_REQUESTOUTOFRANGE](#) is transferred back to the caller of the [Dcm Low Proxy](#).] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02308]{DRAFT} [In case the passed [DataLength](#), [ScalingInfoLength](#), or [controlMaskLength](#) (if applicable, see [\[SWS_SwCluC_02305\]](#)) of the caller of the [Dcm Low Proxy](#) do not match the configured data size in the [Dcm](#) configuration in the [Applicative Software Cluster](#), the application error [E_NOT_OK](#) and the [ErrorCode](#) [DCM_E_REQUESTOUTOFRANGE](#) is transferred back to the caller of the [Dcm Low Proxy](#). No data, [scalingInfo](#) or [controlMask](#) values shall be transferred in this case.] ([SRS_SwCluC_00208](#))

Please note: In case the data has dynamic length configured in the [Applicative Software Cluster](#) for this diagnostic data element, the [DataLength](#) cannot be checked.

DataServices_{Data} via sender-receiver / nv-data interfaces

[SWS_SwCluC_02309]{DRAFT} [The [Dcm High Proxy](#) shall support the configuration of [DataServices_{Data} RPortPrototypes](#) for [SenderReceiverInterface/NvDataInterface](#) [DataServices_{Data}](#).] ([SRS_SwCluC_00208](#))

Note: In the corresponding [Dcm](#) configuration, [DcmDspDataUsePort](#) will usually be set to [USE_DATA_SENDER_RECEIVER](#) or [USE_DATA_SENDER_RECEIVER_AS_SERVICE](#) AND the related parameters [DcmDspDidRead](#) will not exist AND [DcmDspDidWrite](#) will not exist.

[SWS_SwCluC_02310]{DRAFT} [The [Dcm High Proxy](#) shall support the configuration of [DataServices_{Data} PPortPrototypes](#) for [SenderReceiverInterface/NvDataInterface](#) [DataServices_{Data}](#).] ([SRS_SwCluC_00208](#))

Note: In the corresponding Dcm configuration, `DcmDspDataUsePort` will usually be set to `USE_DATA_SENDER_RECEIVER` or `USE_DATA_SENDER_RECEIVER_AS_SERVICE` AND the related parameters `DcmDspDidRead` will not exist AND `DcmDspDidWrite` will exist.

[SWS_SwCluC_02311]{DRAFT} [The `Dcm High Proxy` shall support the configuration of `DataServices_{Data} PRPortPrototypes` for `SenderReceiverInterface/NvDataInterface DataServices_{Data}`.] (*SRS_SwCluC_00208*)

Note: In the corresponding Dcm configuration, `DcmDspDataUsePort` will usually be set to `USE_DATA_SENDER_RECEIVER` or `USE_DATA_SENDER_RECEIVER_AS_SERVICE` AND the related parameters `DcmDspDidRead` will exist AND `DcmDspDidWrite` will not exist.

[SWS_SwCluC_02312]{DRAFT} [In case the `SwCluCcmProxyDiagnosticData` channel related to a `DataServices_{Data} PortPrototypes` is connected, the Dcm Proxy shall

- read the `dataElement/nvData`, when the corresponding `Xxx_ReadData` API is called in the `Dcm Low Proxy`,
- OR
- write the `dataElement/nvData`, when the corresponding `Xxx_WriteData` API is called in the `Dcm Low Proxy`.

In that case IN argument `Data` (if applicable) is sent to the receiver, and OUT argument `Data` (if applicable) is transferred back to the caller of the `Dcm Low Proxy`.] (*SRS_SwCluC_00208*)

[SWS_SwCluC_02313]{DRAFT} [In case the `DataServices_{Data} SenderReceiverInterface/NvDataInterface` is configured, the default value `DCM_POS_RESP` as `ErrorCode` is transferred back to the caller of the `Dcm Low Proxy` for any connected `RPortPrototype` and any connected or unconnected `PPortPrototype` or `PRPortPrototype`.] (*SRS_SwCluC_00208*)

[SWS_SwCluC_02314]{DRAFT} [In case the `DataServices_{Data} RPortPrototype`, `PPortPrototype` or `PRPortPrototype` is unconnected, the return value `E_NOT_OK` and the `ErrorCode DCM_E_REQUESTOUTOFRANGE` is transferred back to the caller of the `Dcm Low Proxy`.] (*SRS_SwCluC_00208*)

[SWS_SwCluC_02315]{DRAFT} [In case the passed `DataLength` of the caller of the `Dcm Low Proxy` does not match the configured data size in the Dcm configuration in the `Applicative Software Cluster`, the return value `E_NOT_OK` and the `ErrorCode DCM_E_REQUESTOUTOFRANGE` is transferred back to the caller of the `Dcm Low Proxy`. No data shall be transferred in this case.] (*SRS_SwCluC_00208*)

DataServices_{DID} ports

The `DataServices_{DID}` ports are used to access entire data belonging to a DID.

[SWS_SwCluC_02316]{DRAFT} [The *Dcm High Proxy* shall support the configuration of *DataServices_{DID}* *RPortPrototypes* for *SenderReceiverInterface* *DataServices_{DID}*.] (*SRS_SwCluC_00208*)

Note: In the corresponding *Dcm* configuration, *DcmDspDidUsePort* will usually be set to *USE_ATOMIC_SENDER_RECEIVER_INTERFACE* or *USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE* AND the related parameters *DcmDspDidRead* will exist AND *DcmDspDidWrite* will not exist OR *DcmDspDidControl* will exist.

[SWS_SwCluC_02317]{DRAFT} [The *Dcm High Proxy* shall support the configuration of *DataServices_{DID}* *PPortPrototypes* for *SenderReceiverInterface* *DataServices_{DID}*.] (*SRS_SwCluC_00208*)

Note: In the corresponding *Dcm* configuration, *DcmDspDidUsePort* will usually be set to *USE_ATOMIC_SENDER_RECEIVER_INTERFACE* or *USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE* AND the related parameters *DcmDspDidRead* will not exist AND *DcmDspDidWrite* will exist OR *DcmDspDidControl* will exist.

[SWS_SwCluC_02318]{DRAFT} [The *Dcm High Proxy* shall support the configuration of *DataServices_{DID}* *PRPortPrototypes* for *SenderReceiverInterface* *DataServices_{DID}*.] (*SRS_SwCluC_00208*)

Note: In the corresponding *Dcm* configuration, *DcmDspDidUsePort* will usually be set to *USE_ATOMIC_SENDER_RECEIVER_INTERFACE*, *USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE*, or *USE_ATOMIC_NV_DATA_INTERFACE* AND the related parameters *DcmDspDidRead* will exist AND *DcmDspDidWrite* will exist OR *DcmDspDidControl* will exist.

[SWS_SwCluC_02319]{DRAFT} [In case the *SwCluCdcmProxyDiagnosticData* channel related to a *DataServices_{DID}* *PortPrototypes* is connected, the *Dcm Proxy* shall

- read the *dataElement*, when the corresponding *Xxx_ReadData* API is called in the *Dcm Low Proxy*,
- OR
- write the *dataElement*, when the corresponding *Xxx_WriteData* API is called in the *Dcm Low Proxy*,

In that case, IN argument *Data* (if applicable) is sent to the receiver and OUT argument *Data* (if applicable) is transferred back to the caller of the *Dcm Low Proxy*.] (*SRS_SwCluC_00208*)

[SWS_SwCluC_02320]{DRAFT} [In case the *DataServices_{DID}* *SenderReceiverInterface* is configured, the default value *DCM_POS_RESP* as *ErrorCode* is transferred back to the caller of the *Dcm Low Proxy* for any connected *RPortPrototype* and any connected or unconnected *PPortPrototype* or *PRPortPrototype*.] (*SRS_SwCluC_00208*)

[SWS_SwCluC_02321]{DRAFT} [In case the *DataServices_{DID}* *RPortPrototype* is unconnected, the return value *E_NOT_OK* and the *ErrorCode*

DCM_E_REQUESTOUTOFRANGE is transferred back to the caller of the `Dcm Low Proxy`.](SRS_SwCluC_00208)

[SWS_SwCluC_02322]{DRAFT} [In case the passed `DataLength` of the caller of the `Dcm Low Proxy` does not match the configured data size in the `Dcm` configuration in the `Applicative Software Cluster`, the return value `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTOUTOFRANGE` is transferred back to the caller of the `Dcm Low Proxy`. No data shall be transferred in this case.](SRS_SwCluC_00208)

IOControlRequest_{DID} and IOControlResponse_{DID} ports

The `IOControlRequest_{DID}` and `IOControlResponse_{DID}` sender-receiver ports are used for the `Service 0x2F - InputOutputControlByIdentifier` as an alternative to the `DataServices_{Data}` client-server ports

[SWS_SwCluC_02323]{DRAFT} [The `Dcm High Proxy` shall support the configuration of `IOControlRequest_{DID}` `RPortPrototypes` for `SenderReceiverInterface` `IOControlRequest_{DID}`.](SRS_SwCluC_00208)

[SWS_SwCluC_02324]{DRAFT} [The `Dcm High Proxy` shall support the configuration of `IOControlResponse_{DID}` `RPortPrototypes` for `SenderReceiverInterface` `IOControlResponse_{DID}`.](SRS_SwCluC_00208)

Note: In the corresponding `Dcm` configuration, `DcmDspDidUsePort` will usually be set to `USE_ATOMIC_SENDER_RECEIVER_INTERFACE` or `USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE` and the related container `DcmDspDid` will exist.

[SWS_SwCluC_02325]{DRAFT} [In case the `SwCluCdcmProxyDiagnosticData` channel, related to a `IOControlRequest_{DID}` and `IOControlResponse_{DID}` sender-receiver port, is connected, the `Dcm Proxy` shall:

- calculate the `underControl`, `IOOperationRequest` and `controlState` data elements, according to [SWS_Dcm_01435], [SWS_Dcm_00682], [SWS_Dcm_01436], [SWS_Dcm_01437], [SWS_Dcm_01438], [SWS_Dcm_01275], [SWS_Dcm_01277] AND
- return the `IOOperationResponse`, according to [SWS_Dcm_01439], [SWS_Dcm_01440], and [SWS_Dcm_01441], when the corresponding `Xxx_ReadData`, `Xxx_WriteData`, `Xxx_ReturnControlToEcu`, `Xxx_ResetToDefault`, `Xxx FreezeCurrentState`, or `Xxx_ShortTermAdjustment` API is called in the `Dcm Low Proxy`.

](SRS_SwCluC_00208)

Binary Manifest

[SWS_SwCluC_02391]{DRAFT} [The `Dcm High Proxy` shall create for each `diagnostic data proxy` - configured via `SwCluCdcmProxyDiagnosticData` container - a provided resource entry in the `Binary Manifest`.]()

7.4.3.5.2.2 Routine Control proxy

The routine control proxy provides access to routine control services in Application Software Clusters, which can be accessed by UDS diagnostic services 0x31 RoutineControl via the Dcm in the Host Software Cluster.

RoutineServices_{RoutineName} ports

[SWS_SwCluC_02330]{DRAFT} [The Dcm High Proxy shall support the configuration of RoutineServices_{RoutineName} RPortPrototypes for ClientServerInterface RoutineServices_{RoutineName}.] (SRS_SwCluC_00208)

[SWS_SwCluC_02331]{DRAFT} [In case the SwCluC DcmProxyRoutineControl channel related to a RoutineServices_{RoutineName} RPortPrototypes is connected, the Dcm Proxy shall invoke the

- Start operation, when the corresponding Xxx_Start API is called in the Dcm Low Proxy,
OR
- StartConfirmation operation, when the corresponding Xxx_StartConfirmation API is called in the Dcm Low Proxy,
OR
- Stop operation, when the corresponding Xxx_Stop API is called in the Dcm Low Proxy,
OR
- StopConfirmation operation, when the corresponding Xxx_StopConfirmation API is called in the Dcm Low Proxy,
OR
- RequestResults operation, when the corresponding Xxx_RequestResults API is called in the Dcm Low Proxy,
OR
- RequestResultsConfirmation operation, when the corresponding Xxx_RequestResultsConfirmation API is called in the Dcm Low Proxy,

In this case, INOUT arguments (dataInOut (if applicable)) are used to transfer the individual dataIn signals to the client, and to transfer back the individual dataOut signals as joint array to the caller of the Dcm Low Proxy. The configuration of the operations in the ClientServerInterfaces is according to the Dcm configuration of the Applicative Software Cluster. Additionally, IN arguments OpStatus (if applicable) and ConfirmationStatus (if applicable) are passed to the client, and OUT arguments ErrorCode (if applicable) and application errors are transferred back to the caller of the Dcm Low Proxy.] (SRS_SwCluC_00208)

[SWS_SwCluC_02332]{DRAFT} [In case the RoutineServices_{RoutineName} RPortPrototype is unconnected, the return value E_NOT_OK and the ErrorCode DCM_E_REQUESTOUTOFRANGE (if applicable) is transferred back to the caller of the Dcm Low Proxy.](SRS_SwCluC_00208)

[SWS_SwCluC_02333]{DRAFT} [In case the passed DataLength of the caller of the Dcm Low Proxy does not match the configured data size in the Dcm configuration in the Applicative Software Cluster, the return value E_NOT_OK and the ErrorCode DCM_E_REQUESTOUTOFRANGE is transferred back to the caller of the Dcm Low Proxy. No data shall be transferred in this case.](SRS_SwCluC_00208)

Please note: In case dynamic length signals are configured in the Applicative Software Cluster for this routine control, the DataLength cannot be checked.

Binary Manifest

[SWS_SwCluC_02392]{DRAFT} [The Dcm High Proxy shall create for each routine control proxy - configured via SwCluCdcmProxyRoutineControl container - a provided resource entry in the Binary Manifest.](/)

7.4.3.5.2.3 Service Request Notification proxy

The service request notification proxy provides access to the service request manufacturer notification and service request supplier notification

ServiceRequestManufacturerNotification_{Name} ports

[SWS_SwCluC_02334]{DRAFT} [The Dcm High Proxy shall support the configuration of ServiceRequestManufacturerNotification_{Name} RPortPrototypes for ClientServerInterface ServiceRequestManufacturerNotification_{Name}.](SRS_SwCluC_00208)

ServiceRequestSupplierNotification_{Name} ports

[SWS_SwCluC_02335]{DRAFT} [The Dcm High Proxy shall support the configuration of ServiceRequestSupplierNotification_{Name} RPortPrototypes for ClientServerInterface ServiceRequestSupplierNotification_{Name}.](SRS_SwCluC_00208)

[SWS_SwCluC_02336]{DRAFT} [In case the SwCluCdcmProxyServiceRequestNotification channel related to a ServiceRequestManufacturerNotification_{Name} or ServiceRequestSupplierNotification_{Name} RPortPrototypes is connected, the Dcm Proxy shall invoke the

- Confirmation operation, when the corresponding Xxx_Confirmation API is called in the Dcm Low Proxy,

OR

- Indication `operation`, when the corresponding `Xxx_Indication` API is called in the `Dcm Low Proxy`.

In this case, IN arguments [`SID` (if applicable), `ReqType` (if applicable), `ConnectionId` (if applicable), `ConfirmationStatus` (if applicable), `ProtocolType` (if applicable), `TesterSourceAddress` (if applicable), `RequestData` (if applicable), `DataSize` (if applicable)] are passed to the client and OUT arguments [`ErrorCode` (if applicable)] and application errors are transferred back to the caller of the `Dcm Low Proxy`.] ([SRS_SwCluC_00208](#))

[[SWS_SwCluC_02337](#)]{DRAFT} [In case the `ServiceRequestManufacturerNotification_{Name}` or `ServiceRequestSupplierNotification_{Name}` `RPortPrototype` is unconnected, the return value `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTTOUTOFRANGE` (if applicable) is transferred back to the caller of the `Dcm Low Proxy`.] ([SRS_SwCluC_00208](#))

Binary Manifest

[[SWS_SwCluC_02393](#)]{DRAFT} [The `Dcm High Proxy` shall create for each `service request notification proxy` - configured via `SwCluCdcmProxyServiceRequestNotification` container - a provided resource entry in the `Binary Manifest`.] ()

7.4.3.5.3 Dcm Low Proxy

7.4.3.5.3.1 Diagnostic Data proxy

ReadData

[[SWS_SwCluC_02350](#)]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_ReadData` function per configured `SwCluCdcmProxyDiagnosticData` container in the `Dcm Low Proxy`, if the `diagnostic data element read access` is enabled.] ([SRS_SwCluC_00208](#))

[[SWS_SwCluC_02351](#)]{DRAFT} [The `Xxx_ReadData` function shall be named `SwCluC_DcmProxy_ReadData_<DDSN>`, where `<DDSN>` is the `shortName` of the `SwCluCdcmProxyRoutineControl` container.] ([SRS_SwCluC_00208](#))

[[SWS_SwCluC_02352](#)]{DRAFT} [The `diagnostic data element read access` for a `SwCluCdcmProxyDiagnosticData` is enabled, if:

- `SwCluCdcmProxyDataRef` references a `DcmDspData` AND the corresponding `DcmDspDid` has a `DcmDspDidRead` configured in its related `DcmDspDidInfo`

OR

- `SwCluCdcmProxyDataRef` references a `DcmDspDid`, which has a `DcmDspDidRead` configured in its related `DcmDspDidInfo`

OR

- `SwCluCDcmProxyDataRef` references a `DcmDspPidService01`

OR

- `SwCluCDcmProxyDataRef` references a `DemDataElementClass`

](*SRS_SwCluC_00208*)

[SWS_SwCluC_02380]{DRAFT} [The `Xxx_ReadData` function of `Dcm Low Proxy` shall return `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTOUTOFRANGE`, if the corresponding `diagnostic data proxy` is not connected to an instance in the `Dcm High Proxy`. No data is returned in this case.](*SRS_SwCluC_00208*)

WriteData

[SWS_SwCluC_02353]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_WriteData` function per configured `SwCluCDcmProxyDiagnosticData` container in the `Dcm Low Proxy`, if the `diagnostic data element write access` is enabled.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02354]{DRAFT} [The `Xxx_WriteData` function shall be named `SwCluC_DcmProxy_WriteData_<DDSN>`, where `<DDSN>` is the `shortName` of the `SwCluCDcmProxyRoutineControl` container.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02355]{DRAFT} [The `diagnostic data element write access` for a `SwCluCDcmProxyDiagnosticData` is enabled, if:

- `SwCluCDcmProxyDataRef` references a `DcmDspData` AND the corresponding `DcmDspDid` has a `DcmDspDidWrite` configured in its related `DcmDspDidInfo`

OR

- `SwCluCDcmProxyDataRef` references a `DcmDspDid`, which has a `DcmDspDidWrite` configured in its related `DcmDspDidInfo`

](*SRS_SwCluC_00208*)

[SWS_SwCluC_02381]{DRAFT} [The `Xxx_WriteData` function of `Dcm Low Proxy` shall return `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTOUTOFRANGE`, if the corresponding `diagnostic data proxy` is not connected to an instance in the `Dcm High Proxy`.](*SRS_SwCluC_00208*)

ReadDataLength

[SWS_SwCluC_02356]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_ReadDataLength` function per configured `SwCluCDcmProxyDiagnosticData` container in the `Dcm Low Proxy`, if the `diagnostic data element read data length access` is enabled.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02357]{DRAFT} [The `Xxx_ReadDataLength` function shall be named `SwCluC_DcmProxy_ReadDataLength_<DDSN>`, where `<DDSN>` is the `shortName` of the `SwCluCDcmProxyRoutineControl` container.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02358]{DRAFT} [The diagnostic data element read data length access for a [SwCluCDcmProxyDiagnosticData](#) is enabled, if:

- [SwCluCDcmProxyDataRef](#) references a [DcmDspData](#) AND the corresponding [DcmDspDid](#) has a [DcmDspDidRead](#) configured in its related [DcmDspDidInfo](#)
- AND
- the [DcmDspDataType](#) is set to `UINT8_DYN`.

]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02382]{DRAFT} [The `Xxx_ReadDataLength` function of [Dcm Low Proxy](#) shall return `E_NOT_OK`, if the corresponding [diagnostic data proxy](#) is not connected to an instance in the [Dcm High Proxy](#).]([SRS_SwCluC_00208](#))

ConditionCheckRead

[SWS_SwCluC_02359]{DRAFT} [The [Dcm Low Proxy](#) shall provide an instance of a `Xxx_ConditionCheckRead` function per configured [SwCluCDcmProxyDiagnosticData](#) container in the [Dcm Low Proxy](#), if the [diagnostic data element condition check read access](#) is enabled.]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02360]{DRAFT} [The `Xxx_ConditionCheckRead` function shall be named `SwCluC_DcmProxy_ConditionCheckRead_<DDSN>`, where `<DDSN>` is the [shortName](#) of the [SwCluCDcmProxyRoutineControl](#) container.]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02361]{DRAFT} [The diagnostic data element condition check read access for a [SwCluCDcmProxyDiagnosticData](#) is enabled, if:

- [SwCluCDcmProxyDataRef](#) references a [DcmDspData](#) AND the corresponding [DcmDspDid](#) has a [DcmDspDidRead](#) configured in its related [DcmDspDidInfo](#)
- AND
- the parameter [DcmDspDataConditionCheckReadFncUsed](#) is set to `true`.

]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02383]{DRAFT} [The `Xxx_ConditionCheckRead` function of [Dcm Low Proxy](#) shall return `E_NOT_OK`, if the corresponding [diagnostic data proxy](#) is not connected to an instance in the [Dcm High Proxy](#).]([SRS_SwCluC_00208](#))

GetScalingInformation

[SWS_SwCluC_02362]{DRAFT} [The [Dcm Low Proxy](#) shall provide an instance of a `Xxx_GetScalingInformation` function per configured [SwCluCDcmProxyDiagnosticData](#) container in the [Dcm Low Proxy](#), if the [diagnostic data element scaling information access](#) is enabled.]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02363]{DRAFT} [The `Xxx_GetScalingInformation` function shall be named `SwCluC_DcmProxy_GetScalingInformation_<DDSN>`, where

<DDSN> is the `shortName` of the `SwCluCdcProxyRoutineControl` container.]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02364]{DRAFT} [The diagnostic data element scaling information access for a `SwCluCdcProxyDiagnosticData` is enabled, if:

- `SwCluCdcProxyDataRef` references a `DcmDspData`

AND

- the corresponding `DcmDspDataScalingInfoSize` is configured in its related `DcmDspDataInfo`.

]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02384]{DRAFT} [The `Xxx_GetScalingInformation` function of `Dcm Low Proxy` shall return `E_NOT_OK`, if the corresponding `diagnostic data proxy` is not connected to an instance in the `Dcm High Proxy`.]([SRS_SwCluC_00208](#))

ReturnControlToECU

[SWS_SwCluC_02365]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_ReturnControlToECU` function per configured `SwCluCdcProxyDiagnosticData` container in the `Dcm Low Proxy`, if the `diagnostic data element return control to ECU` is enabled.]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02366]{DRAFT} [The `Xxx_ReturnControlToECU` function shall be named `SwCluC_DcmProxy_ReturnControlToECU_<DDSN>`, where `<DDSN>` is the `shortName` of the `SwCluCdcProxyRoutineControl` container.]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02367]{DRAFT} [The diagnostic data element return control to ECU for a `SwCluCdcProxyDiagnosticData` is enabled, if:

- `SwCluCdcProxyDataRef` references a `DcmDspData`

AND

- the corresponding `DcmDspDid` has a `DcmDspDidControl` configured in its related `DcmDspDidInfo`

AND

- the parameter `DcmDspDidFreezeCurrentState` is set to `true`.

]([SRS_SwCluC_00208](#))

[SWS_SwCluC_02385]{DRAFT} [The `Xxx_ReturnControlToECU` function of `Dcm Low Proxy` shall return `E_NOT_OK`, if the corresponding `diagnostic data proxy` is not connected to an instance in the `Dcm High Proxy`.]([SRS_SwCluC_00208](#))

ResetToDefault

[SWS_SwCluC_02368]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_ResetToDefault` function per configured `SwCluCdcmProxyDiagnosticData` container in the `Dcm Low Proxy`, if the diagnostic data element `reset to default` is enabled.](SRS_SwCluC_00208)

[SWS_SwCluC_02369]{DRAFT} [The `Xxx_ResetToDefault` function shall be named `SwCluC_DcmProxy_ResetToDefault_<DDSN>`, where `<DDSN>` is the `shortName` of the `SwCluCdcmProxyRoutineControl` container.](SRS_SwCluC_00208)

[SWS_SwCluC_02370]{DRAFT} [The diagnostic data element `reset to default` for a `SwCluCdcmProxyDiagnosticData` is enabled, if:

- `SwCluCdcmProxyDataRef` references a `DcmDspData`
AND
- the corresponding `DcmDspDid` has a `DcmDspDidControl` configured in its related `DcmDspDidInfo`
AND
- the parameter `DcmDspDidResetToDefault` is set to `true`.

](SRS_SwCluC_00208)

[SWS_SwCluC_02386]{DRAFT} [The `Xxx_ResetToDefault` function of `Dcm Low Proxy` shall return `E_NOT_OK`, if the corresponding `diagnostic data proxy` is not connected to an instance in the `Dcm High Proxy`.](SRS_SwCluC_00208)

FreezeCurrentState

[SWS_SwCluC_02371]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_FreezeCurrentState` function per configured `SwCluCdcmProxyDiagnosticData` container in the `Dcm Low Proxy`, if the diagnostic data element `freeze current state` is enabled.](SRS_SwCluC_00208)

[SWS_SwCluC_02372]{DRAFT} [The `Xxx_FreezeCurrentState` function shall be named `SwCluC_DcmProxy_FreezeCurrentState_<DDSN>`, where `<DDSN>` is the `shortName` of the `SwCluCdcmProxyRoutineControl` container.](SRS_SwCluC_00208)

[SWS_SwCluC_02373]{DRAFT} [The diagnostic data element `freeze current state` for a `SwCluCdcmProxyDiagnosticData` is enabled, if

- `SwCluCdcmProxyDataRef` references a `DcmDspData`
AND
- the corresponding `DcmDspDid` has a `DcmDspDidControl` configured in its related `DcmDspDidInfo`
AND

- the parameter `DcmDspDidFreezeCurrentState` is set to `true`.

](SRS_SwCluC_00208)

[SWS_SwCluC_02387]{DRAFT} [The `Xxx_FreezeCurrentState` function of `Dcm Low Proxy` shall return `E_NOT_OK`, if the corresponding `diagnostic data proxy` is not connected to an instance in the `Dcm High Proxy`.](SRS_SwCluC_00208)

ShortTermAdjustment

[SWS_SwCluC_02374]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_ShortTermAdjustment` function per configured `SwCluCdcmProxyDiagnosticData` container in the `Dcm Low Proxy`, if the `diagnostic data element short term adjustment` is enabled.](SRS_SwCluC_00208)

[SWS_SwCluC_02375]{DRAFT} [The `Xxx_ShortTermAdjustment` function shall be named `SwCluC_DcmProxy_ShortTermAdjustment_<DDSN>`, where `<DDSN>` is the `shortName` of the `SwCluCdcmProxyRoutineControl` container.](SRS_SwCluC_00208)

[SWS_SwCluC_02376]{DRAFT} [The `diagnostic data element short term adjustment` for a `SwCluCdcmProxyDiagnosticData` is enabled, if

- `SwCluCdcmProxyDataRef` references a `DcmDspData`

AND

- the corresponding `DcmDspDid` has a `DcmDspDidControl` configured in its related `DcmDspDidInfo`

AND

- the parameter `DcmDspDidShortTermAdjustment` is set to `true`.

](SRS_SwCluC_00208)

[SWS_SwCluC_02388]{DRAFT} [The `Xxx_ShortTermAdjustment` function of `Dcm Low Proxy` shall return `E_NOT_OK`, if the corresponding `diagnostic data proxy` is not connected to an instance in the `Dcm High Proxy`.](SRS_SwCluC_00208)

Binary Manifest

[SWS_SwCluC_02394]{DRAFT} [The `Dcm Low Proxy` shall create for each `diagnostic data proxy` - configured via `SwCluCdcmProxyDiagnosticData` container - a required resource entry in the `Binary Manifest`.]()

7.4.3.5.3.2 Routine Control proxy

Start Operation

[SWS_SwCluC_02400]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_Start` function per configured `SwCluCdcmProxyRoutineControl` container in the `Dcm Low Proxy`.] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02401]{DRAFT} [The `Xxx_Start` function shall be named `SwCluC_DcmProxy_Start_<RCSN>`, where `<RCSN>` is the `shortName` of the `SwCluCdcmProxyRoutineControl` container.] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02402]{DRAFT} [The `Xxx_Start` function of `Dcm Low Proxy` shall return `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTOUTOFRANGE`, if the corresponding `routine control proxy` is not connected to an instance in the `Dcm High Proxy`.] ([SRS_SwCluC_00208](#))

StartConfirmation Operation

[SWS_SwCluC_02403]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_StartConfirmation` function per configured `SwCluCdcmProxyRoutineControl` container in the `Dcm Low Proxy`.] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02404]{DRAFT} [The `Xxx_StartConfirmation` function shall be named `SwCluC_DcmProxy_StartConfirmation_<RCSN>`, where `<RCSN>` is the `shortName` of the `SwCluCdcmProxyRoutineControl` container.] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02405]{DRAFT} [The `Xxx_StartConfirmation` function of `Dcm Low Proxy` shall return `E_NOT_OK`, if the corresponding `routine control proxy` is not connected to an instance in the `Dcm High Proxy`.] ([SRS_SwCluC_00208](#))

Stop Operation

[SWS_SwCluC_02406]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_Stop` function per configured `SwCluCdcmProxyRoutineControl` container in the `Dcm Low Proxy`.] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02407]{DRAFT} [The `Xxx_Stop` function shall be named `SwCluC_DcmProxy_Stop_<RCSN>`, where `<RCSN>` is the `shortName` of the `SwCluCdcmProxyRoutineControl` container.] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02408]{DRAFT} [The `Xxx_Stop` function of `Dcm Low Proxy` shall return `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTOUTOFRANGE`, if the corresponding `routine control proxy` is not connected to an instance in the `Dcm High Proxy`.] ([SRS_SwCluC_00208](#))

StopConfirmation Operation

[SWS_SwCluC_02409]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_StopConfirmation` function per configured `SwCluCdcmProxyRoutineControl` container in the `Dcm Low Proxy`.] ([SRS_SwCluC_00208](#))

[SWS_SwCluC_02410]{DRAFT} [The `Xxx_StopConfirmation` function shall be named `SwCluC_DcmProxy_StopConfirmation_<RCSN>`, where `<RCSN>` is the

`shortName` of the `SwCluCDCmProxyRoutineControl` container.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02411]{DRAFT} [The `Xxx_StopConfirmation` function of `Dcm Low Proxy` shall return `E_NOT_OK` if the corresponding `routine control proxy` is not connected to an instance in the `Dcm High Proxy`.](*SRS_SwCluC_00208*)

RequestResults Operation

[SWS_SwCluC_02412]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_RequestResults` function per configured `SwCluCDCmProxyRoutineControl` container in the `Dcm Low Proxy`.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02413]{DRAFT} [The `Xxx_RequestResults` function shall be named `SwCluC_DcmProxy_RequestResults_<RCSN>`, where `<RCSN>` is the `shortName` of the `SwCluCDCmProxyRoutineControl` container.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02414]{DRAFT} [The `Xxx_RequestResults` function of `Dcm Low Proxy` shall return `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTOUTOFRANGE`, if the corresponding `routine control proxy` is not connected to an instance in the `Dcm High Proxy`.](*SRS_SwCluC_00208*)

RequestResultsConfirmation Operation

[SWS_SwCluC_02415]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_RequestResultsConfirmation` function per configured `SwCluCDCmProxyRoutineControl` container in the `Dcm Low Proxy`.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02416]{DRAFT} [The `Xxx_RequestResultsConfirmation` function shall be named `SwCluC_DcmProxy_RequestResultsConfirmation_<RCSN>`, where `<RCSN>` is the `shortName` of the `SwCluCDCmProxyRoutineControl` container.](*SRS_SwCluC_00208*)

[SWS_SwCluC_02417]{DRAFT} [The `Xxx_RequestResultsConfirmation` function of `Dcm Low Proxy` shall return `E_NOT_OK`, if the corresponding `routine control proxy` is not connected to an instance in the `Dcm High Proxy`.](*SRS_SwCluC_00208*)

Binary Manifest

[SWS_SwCluC_02395]{DRAFT} [The `Dcm Low Proxy` shall create for each `routine control proxy` - configured via `SwCluCDCmProxyRoutineControl` container - a required resource entry in the `Binary Manifest`.]()

7.4.3.5.3.3 Service Request Notification proxy

Indication Operation

[SWS_SwCluC_02418]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_Indication` function per configured `SwCluCdcmProxyServiceRequestNotification` container in the `Dcm Low Proxy`.](SRS_SwCluC_00208)

[SWS_SwCluC_02419]{DRAFT} [The `Xxx_Indication` function shall be named `SwCluC_DcmProxy_Indication_<SRNSN>`, where `<SRNSN>` is the `short-Name` of the `SwCluCdcmProxyServiceRequestNotification` container.](SRS_SwCluC_00208)

[SWS_SwCluC_02420]{DRAFT} [The `Xxx_Indication` function of `Dcm Low Proxy` shall return `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTOUTOFRANGE`, if the corresponding `routine control proxy` is not connected to an instance in the `Dcm High Proxy`.](SRS_SwCluC_00208)

Confirmation Operation

[SWS_SwCluC_02421]{DRAFT} [The `Dcm Low Proxy` shall provide an instance of a `Xxx_Confirmation` function per configured `SwCluCdcmProxyServiceRequestNotification` container in the `Dcm Low Proxy`.](SRS_SwCluC_00208)

[SWS_SwCluC_02422]{DRAFT} [The `Xxx_Confirmation` function shall be named `SwCluC_DcmProxy_Confirmation_<SRNSN>`, where `<SRNSN>` is the `short-Name` of the `SwCluCdcmProxyServiceRequestNotification` container.](SRS_SwCluC_00208)

[SWS_SwCluC_02423]{DRAFT} [The `Xxx_Confirmation` function of `Dcm Low Proxy` shall return `E_NOT_OK` and the `ErrorCode` `DCM_E_REQUESTOUTOFRANGE`, if the corresponding `routine control proxy` is not connected to an instance in the `Dcm High Proxy`.](SRS_SwCluC_00208)

Binary Manifest

[SWS_SwCluC_02396]{DRAFT} [The `Dcm Low Proxy` shall create for each `service request notification proxy` - configured via `SwCluCdcmProxyServiceRequestNotification` container - a required resource entry in the `Binary Manifest`.]()

7.4.3.5.3.4 Multiple Partitions

The invoked interface functions need to change the partition in the `Host Software Cluster`, before the calls are ending up in the `Applicative Software Cluster`, if the `Dcm` module does not offer a satellite on this partition. This partition change can be implemented in the `Low Proxy`, by means of the `SchM` or with platform proprietary callouts.

[SWS_SwCluC_02425]{DRAFT} [The `Dcm Low Proxy` shall provide, for each `SwCluCdcmProxyBaseSocket`, a `SwCluC_DcmProxy_Init` API, to initialize the `Dcm High Proxy` of an `Applicative Software Cluster`. This shall be one

element of the service resource `SwCluCdcProxyBaseSocket.`](*SRS_SwCluC_00208*)

[SWS_SwCluC_02424]{DRAFT} [The interface function instance for the diagnostic data proxy, routine control proxy and service request notification proxy, according to [SWS_SwCluC_02350], [SWS_SwCluC_02353], [SWS_SwCluC_02356], [SWS_SwCluC_02359], [SWS_SwCluC_02362], [SWS_SwCluC_02365], [SWS_SwCluC_02368], [SWS_SwCluC_02371], [SWS_SwCluC_02374], [SWS_SwCluC_02400], [SWS_SwCluC_02403], [SWS_SwCluC_02406], [SWS_SwCluC_02409], [SWS_SwCluC_02412], [SWS_SwCluC_02415], are provided either:

- at the `EcucPartition`, referenced by
 - `SwCluCdcProxyDiagnosticData.SwCluCProxyEcucPartitionRef`
 - OR
 - `SwCluCdcProxyRoutineControl.SwCluCProxyEcucPartitionRef`
 - OR
 - `SwCluCdcProxyServiceRequestNotification.SwCluCProxyEcucPartitionRef`
- OR
- if a `SwCluCdcProxyBaseSocket` is directly available, at its `EcucPartition`
- OR
- at the `EcucPartition` of the `SwCluCdcProxyBaseSocket`, referenced by the `SwCluCdcProxyBaseSocket` - via `SwCluCdcProxyUsedSatelliteRef`, applicable for the `EcucPartition` requested above. In this case, the `EcucPartition` shall be changed, before the callback in the High Proxy is called.

](*SRS_SwCluC_00208*)

Binary Manifest

The implementation of the `Dcm Low Proxy` may provide some basic infrastructure for the `Dcm High Proxy` in order to be operational. Such generic infrastructure services can be linked between `Applicative Software Cluster` and `Host Software Cluster` by means of a `Dcm base socket`.

[SWS_SwCluC_02397]{DRAFT} [The `Dcm Low Proxy` shall create for each `SwCluCdcProxyBaseSocket` a provided resource entry in the `Binary Manifest`.]()

7.4.3.6 Dem Proxy

The specified `Dem Proxy` has the following underlying design principle:

In brief, the core business of `Dem` is the management of `Diagnostic Events` and the event memory, used to store failures and to capture related freeze frames with environmental data. This requires mainly interfaces to report and query status `Diagnostic Events`, as well as interfaces to read environmental data for the freeze frames. The `Dem Proxy` provides the ability implement so called diagnostic event proxies, enabling that `Diagnostic Monitors` - reporting diagnostic events - in the `Applicative Software Clusters` can access the `Dem` in the `Host Software Cluster`. APIs and AUTOSAR Service Interfaces that are not related to `Diagnostic Monitors` may not be supported by the `Dem Proxy`. Data relevant for freeze frames are (according diagnostic standards) are also accessible via `Dcm` by DID. Hence, the `diagnostic data proxy` functionality of the `Dem Proxy` has to be used.

7.4.3.6.1 Enable Dem Proxy Generation

[SWS_SwCluC_02748]{DRAFT} [The `Dem High Proxy` code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter `SwCluCProxyGenerationDem` is set to `HIGH_PROXY`.] (SRS_SwCluC_00209)

[SWS_SwCluC_02749]{DRAFT} [The `Dem Low Proxy` code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter `SwCluCProxyGenerationDem` is set to `LOW_PROXY`.] (SRS_SwCluC_00209)

7.4.3.6.2 Dem High Proxy

7.4.3.6.2.1 Dem High Proxy C-API

[SWS_SwCluC_02700]{DRAFT} [The `Dem High Proxy` shall provide the header file `Dem.h`.] (SRS_SwCluC_00209)

[SWS_SwCluC_02701]{DRAFT} [The `Dem High Proxy` shall provide the symbolic name values of `DemEventIds` for `Diagnostic Monitors` in the `Applicative Software Cluster`.] (SRS_SwCluC_00209)

[SWS_SwCluC_02702]{DRAFT} [The `Dem High Proxy` shall provide the functions for `Diagnostic Monitors` reporting `Diagnostic Events`

- `Dem_ClearPrestoredFreezeFrame`
- `Dem_PrestoreFreezeFrame`
- `Dem_ResetEventDebounceStatus`

- Dem_ResetEventStatus
- Dem_SetEventDisabled
- Dem_SetEventStatus

in the [Applicative Software Cluster](#).]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02736]{DRAFT} [The Dem High Proxy shall provide the function [Dem_SetEventStatusWithMonitorData](#), for Diagnostic Monitors reporting Diagnostic Events with monitor data in the [Applicative Software Cluster](#).]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02739]{DRAFT} [The Dem High Proxy shall provide the function [Dem_SetWIRStatus](#), for Diagnostic Monitors reporting Diagnostic Events in the [Applicative Software Cluster](#).]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02703]{DRAFT} [The Dem High Proxy shall provide the functions for Diagnostic Info of Diagnostic Events

- [Dem_GetDTCOfEvent](#)
- [Dem_GetDebouncingOfEvent](#)
- [Dem_GetEventAvailable](#)
- [Dem_GetEventExtendedDataRecordEx](#)
- [Dem_GetEventFreezeFrameDataEx](#)
- [Dem_GetEventUdsStatus](#)
- [Dem_GetFaultDetectionCounter](#)
- [Dem_GetMonitorStatus](#)

in the [Applicative Software Cluster](#).]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02751]{DRAFT} [If development error detection is enabled for Dem High Proxy, the APIs

- [Dem_ClearPrestoredFreezeFrame](#)
- [Dem_PrestoreFreezeFrame](#)
- [Dem_ResetEventDebounceStatus](#)
- [Dem_ResetEventStatus](#)
- [Dem_SetEventDisabled](#)
- [Dem_SetEventStatus](#)
- [Dem_GetDTCOfEvent](#)
- [Dem_GetDebouncingOfEvent](#)

- [Dem_GetEventAvailable](#)
- [Dem_GetEventExtendedDataRecordEx](#)
- [Dem_GetEventFreezeFrameDataEx](#)
- [Dem_GetEventUdsStatus](#)
- [Dem_GetFaultDetectionCounter](#)
- [Dem_GetMonitorStatus](#)
- [Dem_SetEventStatusWithMonitorData](#)
- [Dem_SetWIRStatus](#)

shall report the DET error `DEM_E_UNINIT`, when Dem High Proxy is not yet initialized.]([SRS_SwCluC_00209](#), [SRS_BSW_00369](#), [SRS_BSW_00323](#))

7.4.3.6.2.2 Callback Functions

[SWS_SwCluC_02704]{DRAFT} [The [Dem High Proxy](#) shall support the optional configuration of a `<Module>_DemGetFaultDetectionCounter<ForEvent>` callback, according to [ECUC_Dem_00630] and [ECUC_Dem_00631].]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02705]{DRAFT} [The [Dem High Proxy](#) shall support the optional configuration of a `<Module>_DemInitMonitorFor<EventName>` callback, according to [ECUC_Dem_00632] and [ECUC_Dem_00601].]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02706]{DRAFT} [The [Dem High Proxy](#) shall support the optional configuration of a `<Module>_DemTriggerOnMonitorStatus` callback, according to [ECUC_Dem_00936] and [ECUC_Dem_00937].]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02707]{DRAFT} [The [Dem High Proxy](#) shall support the optional configuration of a `<Module>_DemClearEventAllowed<ForCondition>` callback, according to [ECUC_Dem_00607] and [ECUC_Dem_00609].]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02737]{DRAFT} [The [Dem High Proxy](#) shall support the optional configuration of a `<Module>_DemTriggerOnEventData` callback, according to [ECUC_Dem_00606] and [ECUC_Dem_00608].]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02738]{DRAFT} [The [Dem High Proxy](#) shall support the optional configuration of a `<Module>_DemTriggerOnEventUdsStatus` callback, according to [ECUC_Dem_00628] and [ECUC_Dem_00629].]([SRS_SwCluC_00209](#))

7.4.3.6.2.3 Service Ports

[SWS_SwCluC_02723]{DRAFT} [The [Dem High Proxy](#) shall provide the ports belonging to a specific `Diagnostic Event`

- Event_{Name}
- EventStatus_{Name}

at the partition specific Service Software Component, to which the using [SwComponentPrototype](#) is mapped.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02724]{DRAFT} [The [Dem High Proxy](#) shall provide the DiagnosticMonitor_MonitorData_{Name} port for each specific Diagnostic Event, where DemEventReportingType is set to STANDARD_REPORTING_WITH_MONITOR_DATA at the partition specific Service Software Component, to which the using [SwComponentPrototype](#) is mapped.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02740]{DRAFT} [The [Dem High Proxy](#) shall provide the EventStatus_{Name} port for each specific Diagnostic Event at the partition specific Service Software Component, to which the using [SwComponentPrototype](#) is mapped.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02725]{DRAFT} [The [Dem High Proxy](#) shall provide the CB_ClrEvt_{Name} port for each specific Diagnostic Event, where DemCallback_ClearEventAllowed is configured at the partition specific Service Software Component, to which the using [SwComponentPrototype](#) is mapped.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02726]{DRAFT} [The [Dem High Proxy](#) shall provide the CB_FaultDetectCtr_{Name} port for each specific Diagnostic Event, where DemCallbackGetFDC is configured at the partition specific Service Software Component, to which the using [SwComponentPrototype](#) is mapped.]([SRS_SwCluC_00209](#))

Note: this applies only if DemDebounceAlgorithmClass is set to DebounceMonitorInternal

[SWS_SwCluC_02727]{DRAFT} [The [Dem High Proxy](#) shall provide the CB_InitEvt_{Name} port for each specific Diagnostic Event, where DemCallbackInitMForE is configured at the partition specific Service Software Component, to which the using [SwComponentPrototype](#) is mapped.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02741]{DRAFT} [The [Dem High Proxy](#) shall provide the CB_DataEvt_{Name} port for each specific Diagnostic Event, where DemCallbackEventDataChanged is configured at the partition specific Service Software Component, to which the using [SwComponentPrototype](#) is mapped.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02742]{DRAFT} [The [Dem High Proxy](#) shall provide the CB_EventUdsStatusChanged_{EventName}_{CallbackName} port for each specific Diagnostic Event, where DemCallbackEventUdsStatusChanged is configured at the partition specific Service Software Component, to which the using [SwComponentPrototype](#) is mapped.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02743]{DRAFT} [The [Dem High Proxy](#) shall provide the `CBMonitorStatusChanged_{EventName}_{CallbackName}` port for each specific Diagnostic Event, where `DemCallbackMonitorStatusChanged` is configured at the partition specific Service Software Component, to which the using `SwComponentPrototype` is mapped.]([SRS_SwCluC_00209](#))

Multiple Partitions

[SWS_SwCluC_02708]{DRAFT} [The [Dem High Proxy](#) shall create one partition specific Service Software Component per configured `EcucPartition`, with the name `Dem_{EcucPartition shortName}`.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02709]{DRAFT} [The [Dem High Proxy](#) shall provide the ports belonging to a specific interface connection at the partition specific Service Software Component, to which the using `SwComponentPrototype` is mapped.]([SRS_SwCluC_00209](#))

7.4.3.6.2.4 Binary Manifest

The implementation of the [Dem High Proxy](#) may require some basic infrastructure from the [Dem Low Proxy](#) in order to be operational. Such generic infrastructure services can be linked between [Applicative Software Cluster](#) and [Host Software Cluster](#) by means of a Dem base socket.

[SWS_SwCluC_02744]{DRAFT} [The [Dem High Proxy](#) shall create for each `SwCluCdcmProxyBaseSocket` a required resource entry in the [Binary Manifest](#).]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02745]{DRAFT} [The [Dem High Proxy](#) shall create for each diagnostic event proxy - configured via `SwCluCDemProxyDiagnosticEvent` container - a required resource entry in the [Binary Manifest](#).]([SRS_SwCluC_00209](#))

7.4.3.6.2.5 Handling of unconnected Diagnostic Events

[SWS_SwCluC_02710]{DRAFT} [In case a Diagnostic Event configured in the [Dem High Proxy](#) is not connected to any [Dem Low Proxy](#), the functions / operations

- [Dem_ClearPrestoredFreezeFrame](#) / `ClearPrestoredFreezeFrame`
- [Dem_PrestoreFreezeFrame](#) / `PrestoreFreezeFrame`
- [Dem_ResetEventDebounceStatus](#) / `ResetEventDebounceStatus`
- [Dem_ResetEventStatus](#) / `ResetEventStatus`
- [Dem_SetEventDisabled](#) / `SetEventDisabled`

- [Dem_SetEventStatus](#) / SetEventStatus
- [Dem_SetEventStatusWithMonitorData](#) / SetEventStatusWithMonitorData
- [Dem_SetWIRStatus](#) / SetWIRStatus

of the [Dem High Proxy](#) shall return E_NOT_OK.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02711]{DRAFT} [In case a Diagnostic Event configured in the [Dem High Proxy](#) is not connected to any [Dem Low Proxy](#), the [Dem High Proxy](#) shall return DEM_E_NO_DTC_AVAILABLE and the value 0 for the DTCOfEvent argument, when [Dem_GetDTCOfEvent](#) / GetDTCOfEvent is called.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02712]{DRAFT} [In case a Diagnostic Event configured in the [Dem High Proxy](#) is not connected to any [Dem Low Proxy](#), the [Dem High Proxy](#) shall return E_NOT_OK and the value 0 for the DebouncingState argument, when [Dem_GetDebouncingOfEvent](#) / GetDebouncingOfEvent is called.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02713]{DRAFT} [In case a Diagnostic Event configured in the [Dem High Proxy](#) is not connected to any [Dem Low Proxy](#), the [Dem High Proxy](#) shall return E_NOT_OK and the value FALSE for the AvailableStatus argument, when [Dem_GetEventAvailable](#) / GetEventAvailable is called.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02714]{DRAFT} [In case a Diagnostic Event configured in the [Dem High Proxy](#) is not connected to any [Dem Low Proxy](#), the [Dem High Proxy](#) shall return E_NOT_OK and the value 0 for the Bufsize argument, when [Dem_GetEventExtendedDataRecordEx](#) / GetEventExtendedDataRecordEx is called. The Bufsize argument stays unmodified.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02715]{DRAFT} [In case a Diagnostic Event configured in the [Dem High Proxy](#) is not connected to any [Dem Low Proxy](#), the [Dem High Proxy](#) shall return E_NOT_OK and the value 0 for the Bufsize argument, when [Dem_GetEventFreezeFrameDataEx](#) / GetEventFreezeFrameDataEx is called. The Bufsize argument stays unmodified.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02716]{DRAFT} [In case a Diagnostic Event configured in the [Dem High Proxy](#) is not connected to any [Dem Low Proxy](#), the [Dem High Proxy](#) shall return E_NOT_OK and the value 0 for the UDSStatusByte argument, when [Dem_GetEventUdsStatus](#) / GetEventUdsStatus is called.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02717]{DRAFT} [In case a Diagnostic Event configured in the [Dem High Proxy](#) is not connected to any [Dem Low Proxy](#), the [Dem High Proxy](#) shall return E_NOT_OK and the value 0 for the FaultDetectionCounter argument, when [Dem_GetFaultDetectionCounter](#) / GetFaultDetectionCounter is called.]([SRS_SwCluC_00209](#))

[SWS_SwCluC_02718]{DRAFT} [In case a Diagnostic Event configured in the Dem High Proxy is not connected to any Dem Low Proxy, the Dem High Proxy shall return E_NOT_OK and the value 0 for the MonitorStatus argument, when Dem_GetMonitorStatus / GetMonitorStatus is called.](SRS_SwCluC_00209)

7.4.3.6.3 Dem Low Proxy

As shown in figure 7.10, the Dem Low Proxy shall be able to invoke APIs of the BSW modules in the Host Software Cluster, without a partition change in the Applicative Software Clusters. This partition change is only needed, if the BSW module does not offer a satellite on this partition. The availability of the BSW modules on dedicated EcucPartitions is a property of the Host Software Cluster.

[SWS_SwCluC_02719]{DRAFT} [The Dem Low Proxy shall provide the entry functions for Diagnostic Event reporting and info, according to [SWS_SwCluC_02702] and [SWS_SwCluC_02702], once per configured SwCluCDemProxyBaseSocket at the EcucPartition configured by SwCluCDemProxyBaseSocket.SwCluCProxyEcucPartitionRef.

In case the SwCluCDemProxyUsedSatelliteRef is set, the EcucPartition shall be changed, before the Dem API is called.](SRS_SwCluC_00209)

[SWS_SwCluC_02728]{DRAFT} [The Dem Low Proxy shall provide, for each SwCluCDemProxyBaseSocket, a SwCluC_DemProxy_Init API, to initialize the Dem High Proxy of an Applicative Software Cluster. This shall be one element of the service resource SwCluCDemProxyBaseSocket.](SRS_SwCluC_00209)

[SWS_SwCluC_02729]{DRAFT} [The Dem Low Proxy shall provide an instance of a <Module>_DemGetFaultDetectionCounter<ForEvent> per configured Diagnostic Event (SwCluCDemProxyDiagnosticEvent), in the Dem Low Proxy, if DemDebounceAlgorithmClass is set to DebounceMonitorInternal.](SRS_SwCluC_00209)

[SWS_SwCluC_02730]{DRAFT} [The Dem Low Proxy shall provide an instance of a <Module>_DemInitMonitorFor<EventName> per configured Diagnostic Event (SwCluCDemProxyDiagnosticEvent), in the Dem Low Proxy.](SRS_SwCluC_00209)

[SWS_SwCluC_02731]{DRAFT} [The Dem Low Proxy shall provide an instance of a <Module>_DemTriggerOnMonitorStatus per configured Diagnostic Event (SwCluCDemProxyDiagnosticEvent), in the Dem Low Proxy.](SRS_SwCluC_00209)

[SWS_SwCluC_02732]{DRAFT} [The Dem Low Proxy shall provide an instance of a <Module>_DemClearEventAllowed<ForCondition> per configured Diagnostic Event (SwCluCDemProxyDiagnosticEvent), in the Dem Low Proxy.](SRS_SwCluC_00209)

[SWS_SwCluC_02733]{DRAFT} [The `Dem Low Proxy` implementation shall describe the `Diagnostic Events` provided to `Dem High Proxys` via `DiagnosticEventNeeds` and `DiagnosticEventInfoNeeds`, derived from the `CpSoftwareClusterServiceResource.resourceNeeds`.] ([SRS_SwCluC_00209](#))

Note: [SWS_SwCluC_02733] ensures a partly automated configuration of the `Diagnostic Event Manager`.

7.4.3.6.3.1 Binary Manifest

The implementation of the `Dem Low Proxy` may provide some basic infrastructure for the `Dem High Proxy` in order to be operational. Such generic infrastructure services can be linked between `Applicative Software Cluster` and `Host Software Cluster` by means of a `Dem base socket`.

[SWS_SwCluC_02746]{DRAFT} [The `Dem Low Proxy` shall create, for each `SwCluCdmProxyBaseSocket`, a provided resource entry in the `Binary Manifest`.] ([SRS_SwCluC_00209](#))

[SWS_SwCluC_02747]{DRAFT} [The `Dem Low Proxy` shall create for, each `diagnostic event proxy` - configured via `SwCluCDemProxyDiagnosticEvent` container -, a provided resource entry in the `Binary Manifest`.] ([SRS_SwCluC_00209](#))

7.4.3.6.3.2 Handling of unconnected Diagnostic Events

[SWS_SwCluC_02734]{DRAFT} [In case a `Diagnostic Event` configured in the `Dem Low Proxy` is not connected to any `Dem High Proxy`, the callback functions

- `<Module>_DemGetFaultDetectionCounter<ForEvent>`
- `<Module>_DemClearEventAllowed<ForCondition>`

of the `Dem High Proxy` shall return `E_NOT_OK`.] ([SRS_SwCluC_00209](#))

Note: The callback functions in [SWS_SwCluC_02734] indicate that the call to the diagnostic monitor has failed, in case the `Diagnostic Event` is not connected.

[SWS_SwCluC_02735]{DRAFT} [In case a `Diagnostic Event` configured in the `Dem Low Proxy` is not connected to any `Dem High Proxy`, the callback functions

- `<Module>_DemInitMonitorFor<EventName>`
- `<Module>_DemTriggerOnMonitorStatus`

of the `Dem High Proxy` shall return `E_OK`.] ([SRS_SwCluC_00209](#))

Note: The callback functions in [SWS_SwCluC_02735] do not support an error code different than `E_OK`.

7.4.3.7 FiM Proxy

The FiM Proxy provides the ability implement so called diagnostic function inhibition proxies, enabling that software in the [Applicative Software Clusters](#) can access the FiM in the [Host Software Cluster](#), to get function permissions or to set the function availability.

The specified FiM Proxy has the following underlying design principle:

In the [Applicative Software Cluster](#), Function Inhibitions supported by the FiM High Proxy are configured with [EcucModuleConfigurationValues](#) for a [FiM EcucModuleDef](#), with according [FiMFIDs](#) containers. In the [Applicative Software Cluster](#) and in the [Host Software Cluster](#), the configuration of [SwCluCfiMProxyFID](#) containers map the [FiMFIDs](#) to [CpSoftwareClusterServiceResources](#), defining each the [globalResourceId](#) and the [isMandatory](#) attribute, relevant for the corresponding [Resource Entry](#) in the [Binary Manifest](#).

The connection between the [FiM High Proxy](#) and the [FiM Low Proxy](#) utilizes the [Binary Manifest](#). Finally, the [FiM Low Proxy](#) invokes the FiM APIs in the [Host Software Cluster](#) in the partitions the FiM module is supporting.

The configuration of the [Function Inhibition](#) with the relationships between diagnostic events and [FIDs](#) resides in the [Host Software Cluster](#). To adjust the intended inhibition without rebuild of the [Host Software Cluster](#), the FiM module provides post-build configurability to large extend.

7.4.3.7.1 Enable FiM Proxy Generation

[SWS_SwCluC_03214]{DRAFT} [The [FiM High Proxy](#) code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter [SwCluCProxyGenerationFiM](#) is set to [HIGH_PROXY](#).] ([SRS_SwCluC_00210](#))

[SWS_SwCluC_03215]{DRAFT} [The [FiM Low Proxy](#) code, and related AUTOSAR model descriptions, shall only be created, if the configuration parameter [SwCluCProxyGenerationFiM](#) is set to [LOW_PROXY](#).] ([SRS_SwCluC_00210](#))

7.4.3.7.2 FiM High Proxy

7.4.3.7.2.1 FiM High Proxy C-API

[SWS_SwCluC_03200]{DRAFT} [The [FiM High Proxy](#) shall provide the header file [FiM.h](#).] ([SRS_SwCluC_00210](#))

[SWS_SwCluC_03201]{DRAFT} [The [FiM High Proxy](#) shall provide the symbolic name values for [FiMFIDs](#), for FiM users in the [Applicative Software Cluster](#).] ([SRS_SwCluC_00210](#))

[SWS_SwCluC_03202]{DRAFT} [The `FiM High Proxy` shall provide the functions for `Function Inhibitions`

- `FiM_GetFunctionPermission`
- `FiM_SetFunctionAvailable`

for `FiM` users in the `Applicative Software Cluster`.] (*SRS_SwCluC_00210*)

[SWS_SwCluC_03217]{DRAFT} [If development error detection is enabled for `FiM High Proxy`, the APIs

- `FiM_GetFunctionPermission`
- `FiM_SetFunctionAvailable`

shall report the DET error `FIM_E_UNINIT`, when `FiM High Proxy` is not yet initialized.] (*SRS_SwCluC_00210*, *SRS_BSW_00369*, *SRS_BSW_00323*)

[SWS_SwCluC_03218]{DRAFT} [If development error detection is enabled for `FiM High Proxy`, the APIs

- `FiM_GetFunctionPermission`
- `FiM_SetFunctionAvailable`

shall report the DET error `FIM_E_FID_OUT_OF_RANGE`, if the functions are called with wrong `FID` value.] (*SRS_SwCluC_00210*, *SRS_BSW_00369*, *SRS_BSW_00323*)

7.4.3.7.2.2 Service Ports

[SWS_SwCluC_03204]{DRAFT} [The `FiM High Proxy` shall provide the ports belonging to a specific `Function Inhibition`

- `Func_{Name}`
- `Control_{Name}`

at the partition specific `Service Software Component`, to which the using `SwComponentPrototype` is mapped.] (*SRS_SwCluC_00210*)

Multiple Partitions

[SWS_SwCluC_03203]{DRAFT} [The `FiM High Proxy` shall create one partition specific `Service Software Component` per configured `EcucPartition`, with the name `FiM_<EcucPartition shortName>`.] (*SRS_SwCluC_00210*)

7.4.3.7.2.3 Handling of unconnected `Function Inhibition`

[SWS_SwCluC_03205]{DRAFT} [In case a `Function Inhibition` configured in the `FiM High Proxy` is not connected to any `FiM Low Proxy`, the `FiM High`

Proxy shall return `E_NOT_OK` and the value `FALSE` for the `Permission`, when `FiM_GetFunctionPermission / GetFunctionPermission` is called.](SRS_SwCluC_00210)

[SWS_SwCluC_03206]{DRAFT} [In case a `Function Inhibition` configured in the `FiM High Proxy` is not connected to any `FiM Low Proxy`, the `FiM High Proxy` shall return `E_NOT_OK`, when `FiM_SetFunctionAvailable / SetFunctionAvailable` is called.](SRS_SwCluC_00210)

7.4.3.7.2.4 Binary Manifest

The implementation of the `FiM High Proxy` may require some basic infrastructure from the `FiM Low Proxy` in order to be operational. Such generic infrastructure services can be linked between `Applicative Software Cluster` and `Host Software Cluster` by means of a `FiM base socket`.

[SWS_SwCluC_03210]{DRAFT} [The `FiM High Proxy` shall create for each `SwCluCfiMProxyBaseSocket` a required resource entry in the `Binary Manifest`.](SRS_SwCluC_00210)

[SWS_SwCluC_03211]{DRAFT} [The `FiM High Proxy` shall create for each `diagnostic function inhibition proxy - configured via SwCluCfiMProxyFID` container - a required resource entry in the `Binary Manifest`.](SRS_SwCluC_00210)

7.4.3.7.3 FiM Low Proxy

As shown in figure 7.10, the `FiM Low Proxy` shall be able to invoke APIs of the BSW modules in the `Host Software Cluster`, without a partition change in the `Applicative Software Clusters`. This partition change is only needed, if the BSW module does not offer a satellite on this partition. The availability of the BSW modules on dedicated `EcucPartitions` is a property of the `Host Software Cluster`.

[SWS_SwCluC_03207]{DRAFT} [The `FiM Low Proxy` implementation shall describe the `Function Inhibitions` provided to `FiM High Proxies` via `FunctionInhibitionNeeds`, derived from the `CpSoftwareClusterServiceResource.resourceNeeds`.](SRS_SwCluC_00210)

[SWS_SwCluC_03208]{DRAFT} [The `FiM Low Proxy` implementation shall describe the `control functions` for `Function Inhibitions` provided to `FiM High Proxies` via `FunctionInhibitionAvailabilityNeeds`, derived from the `CpSoftwareClusterServiceResource.resourceNeeds`.](SRS_SwCluC_00210)

Note: **[SWS_SwCluC_03205]** and **[SWS_SwCluC_03206]** ensures a partly automated configuration of the `FiM Manager`.

[SWS_SwCluC_03209]{DRAFT} [The [FiM Low Proxy](#) shall provide, for each [SwCluCFiMProxyBaseSocket](#), a [SwCluC_FiMProxy_Init](#) API, to initialize the [FiM High Proxy](#) of an [Applicative Software Cluster](#). This shall be one element of the service resource [SwCluCFiMProxyBaseSocket](#).]([SRS_SwCluC_00210](#))

7.4.3.7.3.1 Binary Manifest

The implementation of the [FiM Low Proxy](#) may provide some basic infrastructure for the [FiM High Proxy](#) in order to be operational. Such generic infrastructure services can be linked between [Applicative Software Cluster](#) and [Host Software Cluster](#) by means of a .

[SWS_SwCluC_03212]{DRAFT} [The [FiM Low Proxy](#) shall create for each [SwCluCDcmProxyBaseSocket](#) a provided resource entry in the [Binary Manifest](#).]([SRS_SwCluC_00210](#))

[SWS_SwCluC_03213]{DRAFT} [The [FiM Low Proxy](#) shall create for each [diagnostic function inhibition proxy - configured via SwCluCFiMProxyFID](#) container - a provided resource entry in the [Binary Manifest](#).]([SRS_SwCluC_00210](#))

7.4.4 Error Classification

7.4.4.1 Development Errors

[SWS_SwCluC_02140]{DRAFT} **Development Error Types of SwCluC NvM Proxy**
[The development error types defined for the SwCluC NvM Proxy are listed in table [Table 7.4](#).]([SRS_BSW_00337](#), [SRS_BSW_00385](#), [SRS_BSW_00327](#), [SRS_BSW_00480](#), [SRS_BSW_00487](#))

<i>Type of error</i>	<i>Related error code</i>	<i>Value [hex]</i>
API service called with wrong parameter	NVM_E_PARAM_BLOCK_ID	0x0A
API is called with wrong parameter pointer	NVM_E_PARAM_POINTER	0x0F
API is called when NvM Proxy is not initialized yet	NVM_E_UNINIT	0x14

Table 7.4: Development Error Types for SwCluC NvM Proxy

[SWS_SwCluC_02650]{DRAFT} **Development Error Types of SwCluC Com Proxy**
[The development error types defined for the SwCluC Com Proxy are listed in table [Table 7.5](#).]([SRS_BSW_00337](#), [SRS_BSW_00385](#), [SRS_BSW_00327](#), [SRS_BSW_00480](#), [SRS_BSW_00487](#))

<i>Type of error</i>	<i>Related error code</i>	<i>Value [hex]</i>
API service called with wrong parameter	COM_E_PARAM	0x01
API service called with a NULL pointer.	COM_E_PARAM_POINTER	0x03
Error code if any API service is called before the Com High Proxy module was initialized by the Com Low Proxy	COM_E_UNINIT	0x02

Table 7.5: Development Error Types for SwCluC Com Proxy

[SWS_SwCluC_02545]{DRAFT} Development Error Types of SwCluC LdCom Proxy [The development error types defined for the SwCluC LdCom Proxy are listed in table [Table 7.6.](#)] ([SRS_BSW_00337](#), [SRS_BSW_00385](#), [SRS_BSW_00327](#), [SRS_BSW_00480](#), [SRS_BSW_00487](#))

<i>Type of error</i>	<i>Related error code</i>	<i>Value [hex]</i>
Error code, if any API service is called before the LdCom High Proxy module was initialized by the LdCom Low Proxy	LDCOM_E_UNINIT	0x02
API service called with a NULL pointer. In case of this error, the API service shall return immediately without any further action, except for reporting this development error.	LDCOM_E_PARAM_POINTER	0x03
API service called with wrong Signal-ID (LdCom IPdu ID)	LDCOM_E_INVALID_SIGNAL_ID	0x05

Table 7.6: Development Error Types for SwCluC LdCom Proxy

[SWS_SwCluC_02426]{DRAFT} Development Error Types of SwCluC Dcm Proxy [The development error types defined for the SwCluC Dcm Proxy are listed in table [Table 7.7.](#)] ([SRS_BSW_00337](#), [SRS_BSW_00385](#), [SRS_BSW_00327](#), [SRS_BSW_00480](#), [SRS_BSW_00487](#))

<i>Type of error</i>	<i>Related error code</i>	<i>Value [hex]</i>
Error code, if any API service is called before the Dcm High Proxy module was initialized by the Dcm Low Proxy	DCM_E_UNINIT	0x05

Table 7.7: Development Error Types for SwCluC Dcm Proxy

[SWS_SwCluC_02750]{DRAFT} Development Error Types of SwCluC Dem Proxy

[The development error types defined for the SwCluC Dem Proxy are listed in table [Table 7.8.](#)] ([SRS_BSW_00337](#), [SRS_BSW_00385](#), [SRS_BSW_00327](#), [SRS_BSW_00480](#), [SRS_BSW_00487](#))

Type of error	Related error code	Value [hex]
API function called with a NULL pointer.	DEM_E_PARAM_POINTER	0x11
API function called with wrong parameter value	DEM_E_PARAM_DATA	0x12
API function called with wrong length parameter value	DEM_E_PARAM_LENGTH	0x13
Error code, if any API service is called before the Dem High Proxy module was initialized by the Dem Low Proxy	DEM_E_UNINIT	0x20

Table 7.8: Development Error Types for SwCluC Dem Proxy

[SWS_SwCluC_03216]{DRAFT} Development Error Types of SwCluC FiM Proxy

[The development error types defined for the SwCluC FiM Proxy are listed in table [Table 7.9.](#)] ([SRS_BSW_00337](#), [SRS_BSW_00385](#), [SRS_BSW_00327](#), [SRS_BSW_00480](#), [SRS_BSW_00487](#))

Type of error	Related error code	Value [hex]
Error code, if any API service is called before the FiM High Proxy module was initialized by the FiM Low Proxy	FIM_E_UNINIT	0x01
FiM_GetFunctionPermission or FiM_SetFunctionAvailable called with wrong FID	FIM_E_FID_OUT_OF_RANGE	0x02

Table 7.9: Development Error Types for SwCluC FiM Proxy

7.4.4.2 Runtime Errors

Runtime errors are not applicable for the [Cross Cluster Communication of the Software Cluster Connection](#).

7.4.4.3 Transient Faults

Transient Faults are not applicable for the [Cross Cluster Communication of the Software Cluster Connection](#).

7.4.4.4 Production Errors

Production Errors are not applicable for the `Cross Cluster Communication` of the `Software Cluster Connection`.

7.4.4.5 Extended Production Errors

Extended Production Errors are not applicable for the `Cross Cluster Communication` of the `Software Cluster Connection`.

7.5 Standardized Service Resources

This section list the standardized definitions of `CpSoftwareClusterServiceResource` with the defined `category`s and the applicable `resourceNeeds`.

Table 7.10 defines the meaning of the tables for `CpSoftwareClusterServiceResource` in this section.

Column	Description
Category	<code>category</code> of the <code>CpSoftwareClusterServiceResource</code> referencing hte related <code>EcucContainerValues</code> via the <code>resourceNeeds</code> reference.
Context and Container / Parameter	path to the standardized definition
Mult.	Describes the multiplicity, which applies for the Container, Parameter, or Reference when it is used for the description of a <code>CpSoftwareClusterServiceResource</code> . <code>resourceNeeds</code> . This multicity can deviate from the multiplicity defind for the <code>Standardized Module Defintion</code>
Description / Restrictions	Describes the multiplicity, which applies for the Container, Parameter, or Reference when it is used for the description of a <code>CpSoftwareClusterServiceResource</code> . <code>resourceNeeds</code>
Guard Value relevant	Defines whether and how the Container, Parameter, or Reference is treated for the calculation of the Guard Value of a <code>Resource Entry</code> . Following list of standard definitions apply:





	No	The Container, Parameter, or Reference is only used for information or to express a configuration request which is optionally to be fulfilled and does not impact the calculated Guard Value
	Value	The value of the Parameter is considered for the Guard Value calculation
	Choice	Applicable in case the mentioned Container is child of a choice container. The selected kind chosen Container is part of the Guard Value calculation
	from ref	Take the attributes from the referenced Container. The related Container, Parameter, or Reference will be given additionally in this table

Table 7.10: How to read Standardized Service Resource tables

7.5.1 Software Cluster Base Configuration Check

For a [CpSoftwareClusterServiceResource](#), defined for a [Software Cluster Base Configuration Check](#), [\[SWS_SwCluC_90000\]](#) applies.

[\[SWS_SwCluC_90000\]](#){DRAFT} **SWCLUSTER_RES_BASE_CNF** [

Category	SWCLUSTER_RES_BASE_CNF			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.11: SWCLUSTER_RES_BASE_CNF

]()

7.5.2 Cross Cluster Communication

For a [CpSoftwareClusterServiceResource](#), defined for a [Cross Cluster Communication Base Socket](#) (see section 7.3.2.3), [\[SWS_SwCluC_90008\]](#) applies.

[\[SWS_SwCluC_90008\]](#){DRAFT} **SWCLUSTER_RES_XCC_BASE_SOCKET** [

Category	SWCLUSTER_RES_XCC_BASE_SOCKET			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.12: SWCLUSTER_RES_XCC_BASE_SOCKET

}]()

7.5.3 OS Proxy

For a [CpSoftwareClusterServiceResource](#), defined for a [Os Base Socket](#) (see section 7.4.3.1.3), [[SWS_SwCluC_90002](#)] applies.

[[SWS_SwCluC_90002](#)]{DRAFT} **SWCLUSTER_RES_OS_BASE_SOCKET** [

Category	SWCLUSTER_RES_OS_BASE_SOCKET			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.13: SWCLUSTER_RES_OS_BASE_SOCKET

}]()

For a [CpSoftwareClusterServiceResource](#), defined for a [OsApplication](#) (see section 7.4.3.1.5), [[SWS_SwCluC_90003](#)] applies.

[[SWS_SwCluC_90003](#)]{DRAFT} **SWCLUSTER_RES_OS_APPLICATION** [

Category	SWCLUSTER_RES_OS_APPLICATION			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.14: SWCLUSTER_RES_OS_APPLICATION

}]()

For a [CpSoftwareClusterServiceResource](#), defined for a [OsTask](#) (see section 7.4.3.1.5), [[SWS_SwCluC_90004](#)] applies.

[[SWS_SwCluC_90004](#)]{DRAFT} **SWCLUSTER_RES_OS_TASK** [

Category	SWCLUSTER_RES_OS_TASK			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
Os/OsTask	OsTaskPriority	1		Value
Os/OsTask	OsTaskSchedule	1		Value

Table 7.15: SWCLUSTER_RES_OS_TASK

]()

For a [CpSoftwareClusterServiceResource](#), defined for a [OsResource](#) (see section 7.4.3.1.5), [[SWS_SwCluC_90005](#)] applies.

[SWS_SwCluC_90005]{DRAFT} SWCLUSTER_RES_OS_RESOURCE [

Category	SWCLUSTER_RES_OS_RESOURCE			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.16: SWCLUSTER_RES_OS_RESOURCE

]()

For a [CpSoftwareClusterServiceResource](#), defined for a [OsSpinlock](#) (see section 7.4.3.1.5), [[SWS_SwCluC_90006](#)] applies.

[SWS_SwCluC_90006]{DRAFT} SWCLUSTER_RES_OS_SPINLOCK [

Category	SWCLUSTER_RES_OS_SPINLOCK			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.17: SWCLUSTER_RES_OS_SPINLOCK

]()

For a [CpSoftwareClusterServiceResource](#), defined for a task dispatcher (see section 7.4.3.1.5), [[SWS_SwCluC_90007](#)] applies.

[SWS_SwCluC_90007]{DRAFT} SWCLUSTER_RES_OS_TASK_DISPATCHER [

Category	SWCLUSTER_RES_OS_TASK_DISPATCHER			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
Os/OsTask	OsTaskPriority	1		Value
Os/OsTask	OsTaskSchedule	1		Value

Table 7.18: SWCLUSTER_RES_OS_TASK_DISPATCHER

}]()

For a [CpSoftwareClusterServiceResource](#), defined for a function dispatcher (see section 7.4.3.1.5), [\[SWS_SwCluC_90008\]](#) applies.

[SWS_SwCluC_90009]{DRAFT} SWCLUSTER_RES_OS_FNC_DISPATCHER [

Category	SWCLUSTER_RES_OS_FNC_DISPATCHER			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.19: SWCLUSTER_RES_OS_FNC_DISPATCHER

}]()

7.5.4 NvM Proxy

For a [CpSoftwareClusterServiceResource](#), defined for a NvM Base Socket (see section 7.4.3.2.4), [\[SWS_SwCluC_90010\]](#) applies.

[SWS_SwCluC_90010]{DRAFT} SWCLUSTER_RES_NVM_BASE_SOCKET [

Category	SWCLUSTER_RES_NVM_BASE_SOCKET			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.20: SWCLUSTER_RES_NVM_BASE_SOCKET

}]()

For a [CpSoftwareClusterServiceResource](#), defined for a Nv Block (see section 7.4.3.2), [\[SWS_SwCluC_90001\]](#) applies.

[SWS_SwCluC_90001]{DRAFT} SWCLUSTER_RES_NV_BLOCK [

<i>Category</i>	SWCLUSTER_RES_NV_BLOCK			
<i>Context</i>	<i>Container / Parameter</i>	<i>Mult.</i>	<i>Description / Restrictions</i>	<i>Guard Value relevant</i>
NvM/NvMBlockDescriptor	NvMBlockManagementType	1		Value
NvM/NvMBlockDescriptor	NvMBlockUseAutoValidation	1		Value
NvM/NvMBlockDescriptor	NvMBlockUseCrc	1		Value
NvM/NvMBlockDescriptor	NvMBlockUseSetRamBlockStatus	1		Value
NvM/NvMBlockDescriptor	NvMBlockWriteProt	1		Value
NvM/NvMBlockDescriptor	NvMCalcRamBlockCrc	1		Value
NvM/NvMBlockDescriptor	NvMResistantToChangedSw	1		Value
NvM/NvMBlockDescriptor	NvMRomBlockNum	1		Value
NvM/NvMBlockDescriptor	NvMSelectBlockForReadAll	1		Value
NvM/NvMBlockDescriptor	NvMSelectBlockForWriteAll	1		Value
NvM/NvMBlockDescriptor	NvMWriteBlockOnce	1		Value
NvM/NvMCommon	NvMSetRamBlockStatusApi	1		Value
SwCluC/SwCluCPxies/SwCluCNvMProxy/SwCluCNvMProxyNvBlock	SwCluCNvMProxyNvBlockMaxLength	1		Value

Table 7.21: SWCLUSTER_RES_NV_BLOCK

]()

7.5.5 Com Proxy

For a [CpSoftwareClusterServiceResource](#), defined for a [Com base socket](#), [\[SWS_SwCluC_90011\]](#) applies.

[SWS_SwCluC_90011]{DRAFT} SWCLUSTER_RES_COM_BASE_SOCKET [

<i>Category</i>	SWCLUSTER_RES_COM_BASE_SOCKET			
<i>Context</i>	<i>Container / Parameter</i>	<i>Mult.</i>	<i>Description / Restrictions</i>	<i>Guard Value relevant</i>
no standardized resourceNeeds				

Table 7.22: SWCLUSTER_RES_COM_BASE_SOCKET

]()

7.5.6 LdCom Proxy

For a [CpSoftwareClusterServiceResource](#), defined for a [LdCom base socket](#), [[SWS_SwCluC_90012](#)] applies.

[[SWS_SwCluC_90012](#)]{DRAFT} SWCLUSTER_RES_LDCOM_BASE_SOCKET [

Category	SWCLUSTER_RES_LDCOM_BASE_SOCKET			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.23: SWCLUSTER_RES_LDCOM_BASE_SOCKET

]()

7.5.7 Dcm Proxy

For a [CpSoftwareClusterServiceResource](#), defined for a [Dcm base socket](#), [[SWS_SwCluC_90013](#)] applies.

[[SWS_SwCluC_90013](#)]{DRAFT} SWCLUSTER_RES_DCM_BASE_SOCKET [

Category	SWCLUSTER_RES_DCM_BASE_SOCKET			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.24: SWCLUSTER_RES_DCM_BASE_SOCKET

]()

For a [CpSoftwareClusterServiceResource](#), defined for a [diagnostic data proxy](#), [[SWS_SwCluC_90014](#)] applies.

[[SWS_SwCluC_90014](#)]{DRAFT} SWCLUSTER_RES_DCM_DIAGNOSTIC_DATA [

Category	SWCLUSTER_RES_DCM_DIAGNOSTIC_DATA			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.25: SWCLUSTER_RES_DCM_DIAGNOSTIC_DATA

]()

For a `CpSoftwareClusterServiceResource`, defined for a `routine control proxy`, [SWS_SwCluC_90015] applies.

[SWS_SwCluC_90015]{DRAFT} SWCLUSTER_RES_DCM_ROUTINE_CONTROL
 [

Category	SWCLUSTER_RES_DCM_ROUTINE_CONTROL			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.26: SWCLUSTER_RES_DCM_ROUTINE_CONTROL

]()

For a `CpSoftwareClusterServiceResource`, defined for a `service request notification proxy`, [SWS_SwCluC_90016] applies.

[SWS_SwCluC_90016]{DRAFT} SWCLUSTER_RES_DCM_SERVICE_REQUEST_NOTIFICATION
 [

Category	SWCLUSTER_RES_DCM_SERVICE_REQUEST_NOTIFICATION			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.27: SWCLUSTER_RES_DCM_SERVICE_REQUEST_NOTIFICATION

]()

7.5.8 Dem Proxy

For a `CpSoftwareClusterServiceResource`, defined for a `Dem base socket`, [SWS_SwCluC_90017] applies.

[SWS_SwCluC_90017]{DRAFT} SWCLUSTER_RES_DEM_BASE_SOCKET [

Category	SWCLUSTER_RES_DEM_BASE_SOCKET			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.28: SWCLUSTER_RES_DEM_BASE_SOCKET

}]()

For a [CpSoftwareClusterServiceResource](#), defined for a [diagnostic event proxy](#), [[SWS_SwCluC_90018](#)] applies.

[SWS_SwCluC_90018]{DRAFT} SWCLUSTER_RES_DEM_DIAGNOSTIC_EVENT
[

Category	SWCLUSTER_RES_DEM_DIAGNOSTIC_EVENT			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
Dem/DemConfigSet/Dem EventParameter	DemDebounceAlgorithmClass	1	choice of debouncing without contained parameters	Choice
Dem/DemConfigSet/Dem EventParameter	DemEventReportingType	1		Value

Table 7.29: SWCLUSTER_RES_DEM_DIAGNOSTIC_EVENT

}]()

7.5.9 FiM Proxy

For a [CpSoftwareClusterServiceResource](#), defined for a [FiM base socket](#) [[SWS_SwCluC_90019](#)], applies.

[SWS_SwCluC_90019]{DRAFT} SWCLUSTER_RES_FIM_BASE_SOCKET [

Category	SWCLUSTER_RES_FIM_BASE_SOCKET			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.30: SWCLUSTER_RES_FIM_BASE_SOCKET

}]()

For a [CpSoftwareClusterServiceResource](#), defined for a [diagnostic function inhibition proxy](#), [[SWS_SwCluC_90020](#)] applies.

[SWS_SwCluC_90020]{DRAFT} SWCLUSTER_RES_FIM_FID [

Category	SWCLUSTER_RES_FIM_FID			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.31: SWCLUSTER_RES_FIM_FID

]()

7.5.10 WdgM

For a [CpSoftwareClusterServiceResource](#) defined for a WdgM base socket [\[SWS_SwCluC_90021\]](#) applies.

[SWS_SwCluC_90021]{DRAFT} SWCLUSTER_RES_WDGM_BASE_SOCKET [

Category	SWCLUSTER_RES_WDGM_BASE_SOCKET			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.32: SWCLUSTER_RES_WDGM_BASE_SOCKET

]()

Please note the definition of `WdgMBaseSocket` in document [\[14\]](#)

For a [CpSoftwareClusterServiceResource](#) defined for a cross cluster transition proxy [\[SWS_SwCluC_90022\]](#) applies.

[SWS_SwCluC_90022]{DRAFT} SWCLUSTER_RES_WDGM_TRANSITION [

Category	SWCLUSTER_RES_WDGM_TRANSITION			
Context	Container / Parameter	Mult.	Description / Restrictions	Guard Value relevant
no standardized resourceNeeds				

Table 7.33: SWCLUSTER_RES_WDGM_TRANSITION

]()

Please note the definition of `WdgMTransitionProxy` in document [\[14\]](#)

8 API specification

8.1 Imported types

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

□ [

<i>Module</i>	<i>Header File</i>	<i>Imported Type</i>
ComStack_Types	ComStack_Types.h	BufReq_ReturnType
	ComStack_Types.h	CbkHandleIdType (draft)
	ComStack_Types.h	PduInfoType
	ComStack_Types.h	PduLengthType
	ComStack_Types.h	RetryInfoType
	ComStack_Types.h	TpDataStateType
Std	Std_Types.h	Std_ReturnType

]()

Please note that the [Software Cluster Connection](#) additionally utilizes the Platform Types (as defined in document [15]) and Standard Types (as defined in document [16]).

8.2 Type definitions

8.2.1 Binary Manifest

8.2.1.1 SwCluC_BManif_SwClusterIdType

[SWS_SwCluC_10000]{DRAFT} [

Name	SwCluC_BManif_SwClusterIdType (draft)
Kind	Type
Derived from	uint8
Description	Type of the Software Cluster's ID value Tags: atp.Status=draft
Available via	SwCluC_BManif.h

] ([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.2 SwCluC_BManif_MachineIdType

[SWS_SwCluC_10001]{DRAFT} [

Name	SwCluC_BManif_MachineIdType (draft)
Kind	Type
Derived from	uint8
Description	Type for the global ID value of a resource Tags: atp.Status=draft
Available via	SwCluC_BManif.h

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.3 SwCluC_BManif_ConCtrlType

[SWS_SwCluC_10002]{DRAFT} [

Name	SwCluC_BManif_ConCtrlType (draft)		
Kind	Type		
Derived from	uint16		
Range	SWCLUC_BMANIF_DISABLE_ON_ECU_CONNECTION	0x8000	The connection of Software Clusters on ECU is disabled for this machine.
Description	Type to code control flags for the connection process. Bit 0..14 are reserved. Tags: atp.Status=draft		
Available via	SwCluC_BManif.h		

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.4 SwCluC_BManif_ResourcePropertiesType

[SWS_SwCluC_10003]{DRAFT} [

Name	SwCluC_BManif_ResourcePropertiesType (draft)		
Kind	Type		
Derived from	uint8		
Range	SWCLUC_BMANIF_MANDATORY_RESOURCE	0x40	the required resource is mandatory
	SWCLUC_BMANIF_PROVIDED_RESOURCE	0x80	the resource is provided by the Software Cluster
Description	Type to code the properties of a resource. Bit 0..5 are reserved. Tags: atp.Status=draft		
Available via	SwCluC_BManif.h		

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.5 SwCluC_BManif_ResourceTypeIdType

[SWS_SwCluC_10004]{DRAFT} [

Name	SwCluC_BManif_ResourceTypeIdType (draft)
Kind	Type
Derived from	uint8
Description	Type to code the type of a resource Tags: atp.Status=draft
Available via	SwCluC_BManif.h

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.6 SwCluC_BManif_GlobalResourceIdType

[SWS_SwCluC_10005]{DRAFT} [

Name	SwCluC_BManif_GlobalResourceIdType (draft)
Kind	Type
Derived from	uint32
Description	Type for the global ID value of a resource Tags: atp.Status=draft
Available via	SwCluC_BManif.h

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.7 SwCluC_BManif_ResourceGuardValueType

[SWS_SwCluC_10006]{DRAFT} [

Name	SwCluC_BManif_ResourceGuardValueType (draft)
Kind	Type
Derived from	uint32
Description	Type for the guard value of a resource Tags: atp.Status=draft
Available via	SwCluC_BManif.h

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.8 SwCluC_BManif_TableIndexType

[SWS_SwCluC_10007]{DRAFT} [

Name	SwCluC_BManif_TableIndexType (draft)
Kind	Type
Derived from	uint16
Description	Index type to address the immutable and modifiable interface table Tags: atp.Status=draft
Available via	SwCluC_BManif.h

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.9 SwCluC_BManif_HandleIndexType

[SWS_SwCluC_10008]{DRAFT} [

Name	SwCluC_BManif_HandleIndexType (draft)
Kind	Type
Derived from	uint8
Description	Index type to address a set of handles in the immutable and modifiable interface table Tags: atp.Status=draft
Available via	SwCluC_BManif.h

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.2.1.10 SwCluC_BManif_VoidFncPtrType

[SWS_SwCluC_10009]{DRAFT} [

Name	SwCluC_BManif_VoidFncPtrType (draft)
Kind	Function Pointer
Syntax	<pre>int (*SwCluC_BManif_VoidFncPtrType) (void)</pre>
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	int -
Description	Generic function pointer type according C default type for functions Tags: atp.Status=draft
Available via	SwCluC_BManif.h

]([SRS_SwCluC_00006](#))

8.2.1.11 SwCluC_BManif_HandleType

[SWS_SwCluC_10010]{DRAFT} [

Name	SwCluC_BManif_HandleType (draft)	
Kind	Union	
Elements	dptr	
	Type	void*
	Comment	Handle as pointer to variable
	val	
	Type	uint32
	Comment	Handle as value
	fptr	
	Type	SwCluC_BManif_VoidFncPtrType
Comment	Handle as pointer to function	
Description	Type of a handle in the interface table. Tags: atp.Status=draft	
Available via	SwCluC_BManif.h	

]([SRS_SwCluC_00006](#))

8.2.1.12 SwCluC_BManif_HeaderType

[SWS_SwCluC_10011]{DRAFT} [

Name	SwCluC_BManif_HeaderType (draft)	
Kind	Structure	
Elements	Preamble	
	Type	uint64
	Comment	magic pattern of Manifest Header begin
	ManifestMajorVersion	
	Type	uint8
	Comment	version of the Binary Manifest Layout
	ManifestMinorVersion	
	Type	uint8
	Comment	version of the Binary Manifest Layout
	SwClusterId	
	Type	SwCluC_BManif_SwClusterIdType
	Comment	Software Cluster Id
	MachineId	
	Type	SwCluC_BManif_MachineIdType
	Comment	Machine Id
	SwClusterType	
Type	uint8	





Comment	kind of Software Cluster
Reserved1	
Type	uint8
Comment	reserved
Reserved2	
Type	uint8
Comment	reserved
Reserved3	
Type	uint8
Comment	reserved
ConnectorControlFlags	
Type	SwCluC_BManif_ConCtrlType
Comment	Control flags for Software Cluster connector
NoOfInterfaceDescriptors	
Type	uint16
Comment	Number of Interface Descriptors
NoOfOfferedInterfaceHandles	
Type	uint16
Comment	Number of Handles in the Offered Interface
NoOfSubscribedInterfaceHandles	
Type	uint16
Comment	Number of Handles in the Subscribed Interface
ImmutableTablesChecksumPtr	
Type	const uint32*
Comment	Pointer to checksum about immutable descriptor and interface tables
SubscribedTablesValidityMarkerPtr	
Type	const uint32*
Comment	Pointer to validity marker indicating whether all subscribed tables are written
ResourcePropertiesDescriptorColumnPtr	
Type	const SwCluC_BManif_ResourcePropertiesType*
Comment	Pointer to descriptor column for resource properties
ResourceTypeDescriptorColumnPtr	
Type	const SwCluC_BManif_ResourceTypeIdType*
Comment	Pointer to descriptor column for resource type Ids
GlobalResourceIdDescriptorColumnPtr	
Type	const SwCluC_BManif_GlobalResourceIdType*
Comment	Pointer to descriptor column for global resource Ids
ResourceGuardValueDescriptorColumnPtr	
Type	const SwCluC_BManif_ResourceGuardValueType*
Comment	Pointer to descriptor column for guard values
OfferedInterfaceIndexDescriptorColumnPtr	
Type	const SwCluC_BManif_TableIndexType*
Comment	Pointer to descriptor column for offered interface column index
OfferedInterfaceNoOfHandlesDescriptorColumnPtr	





	Type	const SwCluC_BManif_HandleIndexType*
	Comment	Pointer to descriptor column for number of handles in offered interface column
		SubscribedInterfaceIndexDescriptorColumnPtr
	Type	const SwCluC_BManif_TableIndexType*
	Comment	Pointer to descriptor column for subscribed interface column index
		SubscribedInterfaceNoOfHandlesDescriptorColumnPtr
	Type	const SwCluC_BManif_HandleIndexType*
	Comment	Pointer to descriptor column for number of handles in subscribed interface column
		SubscribedInterfaceNoOfHandleSetsDescriptorColumnPtr
	Type	const SwCluC_BManif_HandleIndexType*
	Comment	Pointer to descriptor column for number of handle sets in subscribed interface column
		OfferedInterfaceHandleColumnPtr
	Type	const SwCluC_BManif_HandleType*
	Comment	Pointer to interface column for offered handles
		SubscribedInterfaceDefaultHandleColumnPtr
	Type	const SwCluC_BManif_HandleType*
	Comment	Pointer to default interface column for subscribed handles
		SubscribedInterfaceHandleColumnPtr
	Type	const SwCluC_BManif_HandleType*
	Comment	Pointer to interface column for subscribed handles
		SubscribedInterfaceConnectedSwClusterIdColumnPtr
	Type	const SwCluC_BManif_SwClusterIdType*
	Comment	Pointer to interface column for connected cluster Id values
Description		– Tags: atp.Status=draft
Available via		SwCluC_BManif.h

|(SRS_SwCluC_00006, SRS_BSW_00310)

8.2.2 Cross Cluster Connection

The Cross Cluster Connection implements the types of a [Cross Software Cluster Communication Plug-In](#) as specified in document [9].

8.2.3 ProxyModules

The [Proxy Modules](#) partly implement the types of the according AUTOSAR BSW modules. Those are listed in appendix D.

8.3 Function definitions

8.3.1 BinaryManifest

8.3.1.1 API Principles

None of the functions defined for the `Binary Manifest` access are utilizing the `Std_ReturnType` to provide runtime errors. Since the `Binary Manifest` itself is implemented by a set of const variables, which need to be located to Flash ROM, all the failure situations caused by a temporary unavailability of the `Binary Manifest` can be excluded. Instead, it needs to be considered that the `Binary Manifest` is involved very frequently, since it is needed to access any resource from other `Software Clusters`, including the `Host Software Cluster`.

8.3.1.1.1 Named versus indexed handle access

The APIs for the `Binary Manifest` offer basically two different kinds of access.

The functions `[SWS_SwCluC_10020]`, `[SWS_SwCluC_10021]`, and `[SWS_SwCluC_10032]` provide an entire named access to the handle of a `Resource Entry` in a `Resource Entry Group`. This kind of access is useful, if the using implementation requires a named access on a specific channel. For instance the callbacks of NvM are implemented `NvBlock` specific. In such a `NvBlock` specific callback the related handles to call the `High Proxy` could be easily accessed via name.

In opposite the functions `[SWS_SwCluC_10022]`, `[SWS_SwCluC_10023]`, and `[SWS_SwCluC_10033]` provide an access to a handle of a `Resource Entry` in a `Resource Entry Group`, where the `Resource Entry` is addressed by index. This kind of access is useful, if the using implementation requires indexed based access on channels. Usually service APIs of the `High Proxies` select the channel by an ID value. E.g. the `NvBlock` gets selected by a `BlockId` parameter. Since the order of `Resource Entries` in a `Resource Entry Group` is configurable, the proxy implementation can ensure that the 'channel id' matches to the `resIndex` values. Hence, the passed 'channel id' values can be used without further conversion to get the relevant handles in the `Binary Manifest`.

8.3.1.1.2 Implementation aspects

When an implementation of the Software Cluster Connection or a CDD utilizes the `Binary Manifest`, it is not expected that the usage of the `SwCluC_BManif_GetHandle`, `SwCluC_BManif_GetConSwClusterId`, and `SwCluC_BManif_GetNoOfHandleSets` APIs result in real function calls. From the implementation perspective of the `Binary Manifest`, this should never be required, since accesses to handles or Software Cluster Ids are accesses to primitive constant variables.

[SWS_SwCluC_00057]{DRAFT} [The functions `SwCluC_BManif_GetHandle`, `SwCluC_BManif_GetConSwClusterId`, and `SwCluC_BManif_GetNoOfHandleSets`, according to [\[SWS_SwCluC_10020\]](#), [\[SWS_SwCluC_10021\]](#), [\[SWS_SwCluC_10022\]](#), [\[SWS_SwCluC_10023\]](#), [\[SWS_SwCluC_10032\]](#), and [\[SWS_SwCluC_10033\]](#), shall each be implemented as a C-function-like macro.] ([SRS_SwCluC_00012](#))

[SWS_SwCluC_00058]{DRAFT} [`<Resource Entry Group>` is the `shortName` of either the `SwCluCBManifProvideResourceEntryGroup` or `SwCluCBManifRequireResourceEntryGroup`.] ([SRS_SwCluC_00012](#))

[SWS_SwCluC_00059]{DRAFT} [`<Resource Entry>` is the `shortName` of either the `SwCluCBManifProvideResourceEntry` or `SwCluCBManifRequireResourceEntry`.] ([SRS_SwCluC_00012](#))

[SWS_SwCluC_00060]{DRAFT} [`<Handle>` is the `shortName` of either the `SwCluCBManifProvideHandle` or `SwCluCBManifNotifierHandle` of the `SwCluCBManifResourceType`, referenced by `SwCluCBManifProvideResourceEntryGroup` or `SwCluCBManifRequireResourceEntryGroup`.] ([SRS_SwCluC_00012](#))

[SWS_SwCluC_00061]{DRAFT} [`<handleType>` is either

- the referenced `ImplementationDataType`, if the `SwCluCBManifHandleImpleTypeRef` is defined for a `Provide Handle` or `Notifier Handle`. OR
- `uint32`, if the `SwCluCBManifHandleImpleTypeRef` is not defined, and `SwCluCBManifNativeHandleType` is set to `VALUE` OR
- `void *`, if the `SwCluCBManifHandleImpleTypeRef` is not defined, and `SwCluCBManifNativeHandleType` is set to `DATA_REFERENCE` OR
- `SwCluC_BManif_VoidFncPtrType`, if the `SwCluCBManifHandleImpleTypeRef` is not defined, and `SwCluCBManifNativeHandleType` is set to `FUNCTION_REFERENCE`

] ([SRS_SwCluC_00012](#))

[SWS_SwCluC_CONSTR_00091] [In a set of `SwCluCBManifProvideHandle` in a `SwCluCBManifResourceType` it is not permitted to set `SwCluCBManifIsOptional` to `false`, if any preceding `SwCluCBManifProvideHandle` has already set `SwCluCBManifIsOptional` to `true`.] ([SRS_SwCluC_00014](#))

[SWS_SwCluC_CONSTR_00092] [In a set of `SwCluCBManifNotifierHandle` in a `SwCluCBManifResourceType` it is not permitted to set `SwCluCBManifIsOptional` to `false`, if any preceding `SwCluCBManifNotifierHandle` has already set `SwCluCBManifIsOptional` to `true`.] ([SRS_SwCluC_00014](#))

Rational: A `Resource Type` with optional handles has to define the optional ones after the non-optional ones.

[SWS_SwCluC_CONSTR_00093] [In a set of [SwCluCBManifProvideHandle](#) in a [SwCluCBManifResourceType](#) at most one optional handle shall exist.]([SRS_SwCluC_00014](#))

[SWS_SwCluC_CONSTR_00094] [In a set of [SwCluCBManifNotifierHandle](#) in a [SwCluCBManifResourceType](#) at most one optional handle shall exist.]([SRS_SwCluC_00014](#))

Rational: If more than one optional handle exist, the mapping between the filled handles and the declared handles in the resource type becomes ambiguous. If an implementation requires to vary the number of exchanged handles on a more fine grained level, those can be put into implementation specific structure behind the optional handle. But the minor configuration with the typical highest optimization need can still work without double indirection.

8.3.1.2 SwCluC_BManif_GetHandle

[SWS_SwCluC_10020]{DRAFT} [

Service Name	SwCluC_BManif_GetHandle_<ResourceEntryGroup>_<ResourceEntry>_<Handle> (draft)	
Syntax	<pre><handleType> SwCluC_BManif_GetHandle_<ResourceEntryGroup>_<ResourceEntry>_<Handle> (SwCluC_BManif_HandleIndexType notifierSetIndex)</pre>	
Service ID [hex]	0x10	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	notifierSetIndex	Optional parameter for the notifier set index in the range 0.. SWCLUC_BMANIF_MAX_NO_OF_NOTIFIER_SETS_<Resource Entry Group>_<Resource Entry> -1 It exists if for a notifier handle of a provided resource multiple notifier sets are supported
Parameters (inout)	None	
Parameters (out)	None	
Return value	<handleType>	Pointer or value stored in Binary Manifest for this handle
Description	Returns a handle of a Resource Entry in a Resource Entry Group Tags: atp.Status=draft	
Available via	SwCluC_BManif.h	

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

[SWS_SwCluC_00062]{DRAFT} [The function [SwCluC_BManif_GetHandle](#) according to [\[SWS_SwCluC_10020\]](#) shall exist for each [SwCluCBManifProvideHandle](#) and [SwCluCBManifNotifierHandle](#) defined for each [SwCluCBManifProvideResourceEntry](#) and [SwCluCBManifRequireResourceEntry](#) in a [SwCluCBManifProvideResourceEntryGroup](#) and [SwCluCBManifRequireResourceEntryGroup](#).]([SRS_SwCluC_00006](#))

8.3.1.3 SwCluC_BManif_GetHandle

[SWS_SwCluC_10022]{DRAFT} [

Service Name	SwCluC_BManif_GetHandle_<ResourceEntryGroup>_<Handle> (draft)	
Syntax	<pre><handleType> SwCluC_BManif_GetHandle_<ResourceEntryGroup>_<Handle> (SwCluC_BManif_TableIndexType resIndex, SwCluC_BManif_HandleIndexType notifierSetIndex)</pre>	
Service ID [hex]	0x11	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	resIndex	Index of the resource entry in a Resource Entry Group in the range 0.. SWCLUC_BMANIF_NO_OF_ENTRIES_<Resource Entry Group> -1.
	notifierSetIndex	Optional parameter for the notifier set index in the range 0.. SWCLUC_BMANIF_MAX_NO_OF_NOTIFIER_SETS_<Resource Entry Group>_<Resource Entry> -1 It exists if for a notifier handle of a provided resource multiple notifier sets are supported
Parameters (inout)	None	
Parameters (out)	None	
Return value	<handleType>	Id of the connected Software Cluster
Description	Returns the Id of the connected Software Cluster for a Notifier Handle of a Provide Resource Entry or a Provide Handle of a Require Resource Entry Tags: atp.Status=draft	
Available via	SwCluC_BManif.h	

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

[SWS_SwCluC_00064]{DRAFT} [The function `SwCluC_BManif_GetHandle` according to [SWS_SwCluC_10022] shall exist for each `SwCluCBManifProvideHandle` and `SwCluCBManifNotifierHandle` defined for each `SwCluCBManifProvideResourceEntryGroup` and `SwCluCBManifRequireResourceEntryGroup`.]([SRS_SwCluC_00006](#))

8.3.1.4 SwCluC_BManif_GetConSwClusterId

[SWS_SwCluC_10021]{DRAFT} [

Service Name	SwCluC_BManif_GetConSwClusterId_<ResourceEntryGroup>_<ResourceEntry>_<Handle> (draft)	
Syntax	<pre>SwCluC_BManif_SwClusterIdType SwCluC_BManif_GetConSwClusterId_< ResourceEntryGroup>_<ResourceEntry>_<Handle> (SwCluC_BManif_HandleIndexType notifierSetIndex)</pre>	
Service ID [hex]	0x12	
Sync/Async	Synchronous	





Reentrancy	Reentrant	
Parameters (in)	notifierSetIndex	Optional parameter for the notifier set index in the range 0.. SWCLUC_BMANIF_MAX_NO_OF_NOTIFIER_SETS_<Resource Entry Group>_<Resource Entry> -1 It exists if for a notifier handle of a provided resource multiple notifier sets are supported
Parameters (inout)	None	
Parameters (out)	None	
Return value	SwCluC_BManif_SwClusterIdType	Id of the connected Software Cluster
Description	Returns the Id of the connected Software Cluster for a Notifier Handle of a Provide Resource Entry or a Provide Handle of a Require Resource Entry Tags: atp.Status=draft	
Available via	SwCluC_BManif.h	

](SRS_SwCluC_00006, SRS_BSW_00310)

[SWS_SwCluC_00063]{DRAFT} [The function `SwCluC_BManif_GetConSwClusterId` according to [SWS_SwCluC_10021] shall exist

- for each `SwCluC_BManifNotifierHandle` defined for each `SwCluC_BManifProvideResourceEntry` in a `SwCluC_BManifProvideResourceEntryGroup`

AND

- for each `SwCluC_BManifProvideHandle` defined for each `SwCluC_BManifRequireResourceEntry` in a `SwCluC_BManifRequireResourceEntryGroup`.

](SRS_SwCluC_00006)

8.3.1.5 SwCluC_BManif_GetConSwClusterId

[SWS_SwCluC_10023]{DRAFT} [

Service Name	SwCluC_BManif_GetConSwClusterId_<ResourceEntryGroup>_<Handle> (draft)	
Syntax	SwCluC_BManif_SwClusterIdType SwCluC_BManif_GetConSwClusterId_<ResourceEntryGroup>_<Handle> (SwCluC_BManif_TableIndexType resIndex, SwCluC_BManif_HandleIndexType notifierSetIndex)	
Service ID [hex]	0x13	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	resIndex	Index of the resource entry in a Resource Entry Group in the range 0.. SWCLUC_BMANIF_NO_OF_ENTRIES_<Resource Entry Group> -1.





	notifierSetIndex	Optional parameter for the notifier set index in the range 0.. SWCLUC_BMANIF_MAX_NO_OF_NOTIFIER_SETS_<Resource Entry Group>_<Resource Entry> -1 It exists if for a notifier handle of a provided resource multiple notifier sets are supported.
Parameters (inout)	None	
Parameters (out)	None	
Return value	SwCluC_BManif_SwClusterIdType	Id of the connected Software Cluster
Description	Returns the Id of the connected Software Cluster for a Notifier Handle of a Provide Resource Entry or Provide Handle of a Require Resource Entry Tags: atp.Status=draft	
Available via	SwCluC_BManif.h	

](SRS_SwCluC_00006, SRS_BSW_00310)

[SWS_SwCluC_00065]{DRAFT} [The function SwCluC_BManif_GetConSwClusterId according to [SWS_SwCluC_10023] shall exist

- for each SwCluCBManifNotifierHandle defined for a SwCluCBManifProvideResourceEntryGroup

AND

- for each SwCluCBManifProvideHandle defined for a SwCluCBManifRequireResourceEntryGroup.

](SRS_SwCluC_00006)

8.3.1.6 SwCluC_BManif_GetNoOfHandleSets

[SWS_SwCluC_10032]{DRAFT} [

Service Name	SwCluC_BManif_GetNoOfHandleSets_<Resource Entry Group>_<Resource Entry> (draft)	
Syntax	SwCluC_BManif_HandleIndexType SwCluC_BManif_GetNoOfHandleSets_<Resource Entry Group>_<Resource Entry> (void)	
Service ID [hex]	0x14	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	SwCluC_BManif_HandleIndexType	Number of actually used - and thereby connected - handle sets



△

Description	Returns the number of actually used - and thereby connected - handle sets Tags: atp.Status=draft
Available via	SwCluC_BManif.h

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

[SWS_SwCluC_00066]{DRAFT} [The function `SwCluC_BManif_GetNoOfHandleSets` according to [\[SWS_SwCluC_10032\]](#) shall exist for each `SwCluCBManifProvideResourceEntry` in a `SwCluCBManifProvideResourceEntryGroup` where the parameter `SwCluCBManifMultipleNotifierSupport` is set to `MULTIPLE_NOTIFIER_SETS`.]([SRS_SwCluC_00006](#))

8.3.1.7 SwCluC_BManif_GetNoOfHandleSets

[SWS_SwCluC_10033]{DRAFT} [

Service Name	SwCluC_BManif_GetNoOfHandleSets_<Resource Entry Group> (draft)	
Syntax	SwCluC_BManif_HandleIndexType SwCluC_BManif_GetNoOfHandleSets_<Resource Entry Group> (SwCluC_BManif_TableIndexType resIndex)	
Service ID [hex]	0x15	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	resIndex	Index of the resource entry in a Resource Entry Group in the range 0.. SWCLUC_BMANIF_NO_OF_ENTRIES_<Resource Entry Group> -1.
Parameters (inout)	None	
Parameters (out)	None	
Return value	SwCluC_BManif_HandleIndexType	Number of actually used - and thereby connected - handle sets
Description	Returns the number of actually used - and thereby connected - handle sets. Tags: atp.Status=draft	
Available via	SwCluC_BManif.h	

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

[SWS_SwCluC_00067]{DRAFT} [The function `SwCluC_BManif_GetNoOfHandleSets` according to [\[SWS_SwCluC_10033\]](#) shall exist for each `SwCluCBManifProvideResourceEntryGroup` where the parameter `SwCluCBManifMultipleNotifierSupport` is set to `MULTIPLE_NOTIFIER_SETS`.]([SRS_SwCluC_00006](#))

8.3.1.8 SwCluC_BManif_GetValidityMarker

[SWS_SwCluC_10034]{DRAFT} [

Service Name	SwCluC_BManif_GetValidityMarker (draft)	
Syntax	uint32 SwCluC_BManif_GetValidityMarker (void)	
Service ID [hex]	0x16	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint32	Value of the Subscribed Interface Validity Marker
Description	Returns the value of the Subscribed Interface Validity Marker. Tags: atp.Status=draft	
Available via	SwCluC_BManif.h	

]([SRS_SwCluC_00006](#), [SRS_BSW_00310](#))

8.3.1.9 SWCLUC_BMANIF_NO_OF_ENTRIES

[SWS_SwCluC_01000]{DRAFT} [

Symbolic name: SWCLUC_BMANIF_NO_OF_ENTRIES_<Resource Entry Group>

Value: Number of [Resource Entrys](#) in the according [Resource Entry Group](#)

Comments: n.a.]([SRS_SwCluC_00006](#), [SRS_BSW_00441](#))

[SWS_SwCluC_00068]{DRAFT} [The definition SWCLUC_BMANIF_NO_OF_ENTRIES according to [[SWS_SwCluC_01000](#)] shall exist for each [SwCluC_BManifProvideResourceEntryGroup](#).]([SRS_SwCluC_00006](#))

8.3.1.10 SWCLUC_BMANIF_NO_OF_NOTIFIER_SETS

[SWS_SwCluC_01002]{DRAFT} [

Symbolic name: SWCLUC_BMANIF_MAX_NO_OF_NOTIFIER_SETS_

<Resource Entry Group>_<Resource Entry>

Value: Maximum number of notifier sets for the according [Provide Resource Entry](#)

Comments: n.a.]([SRS_SwCluC_00006](#), [SRS_BSW_00441](#))

[SWS_SwCluC_00069]{DRAFT} [The definition SWCLUC_BMANIF_NO_OF_NOTIFIER_SETS according to [[SWS_SwCluC_01002](#)] shall exist for each [SwCluC_BManifProvideResourceEntry](#) in a [SwCluC_BManifProvideResourceEntryGroup](#) where

the parameter `SwCluC_BManifMultipleNotifierSupport` is set to `MULTIPLE_NOTIFIER_SETS.`] ([SRS_SwCluC_00006](#))

8.3.2 Cross Cluster Communication

In addition to the interfaces listed in this section, the Cross Cluster Connection implements the interfaces of a [Cross Software Cluster Communication Plug-In](#) as specified in document [9].

8.3.2.1 SwCluC_Xcc_Init1

[SWS_SwCluC_11000]{DRAFT} [

Service Name	SwCluC_Xcc_Init1 (draft)
Syntax	<pre>void SwCluC_Xcc_Init1 (void)</pre>
Service ID [hex]	0x20
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	<p>SwCluC_Xcc_Init is intended to allocate and initialize system resources used by the Software Cluster Connection. It initializes interface memory which might be needed to be read by other Software Clusters but it is not allowed to do any read access to memory of any another Software Cluster.</p> <p>Tags: atp.Status=draft</p>
Available via	SwCluC.h

] ([SRS_SwCluC_00100](#), [SRS_SwCluC_00108](#), [SRS_BSW_00101](#), [SRS_BSW_00358](#), [SRS_BSW_00310](#))

8.3.2.2 SwCluC_Xcc_Init2

[SWS_SwCluC_11001]{DRAFT} [

Service Name	SwCluC_Xcc_Init2 (draft)
Syntax	<pre>void SwCluC_Xcc_Init2 (void)</pre>





Service ID [hex]	0x21
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	SwCluC_Xcc_Init2 is intended to allocate and initialize system resources used by the Software Cluster Connection. It can execute initializations which require a read access to interface memory of another Software Clusters. Tags: atp.Status=draft
Available via	SwCluC.h

]([SRS_SwCluC_00100](#), [SRS_SwCluC_00108](#), [SRS_BSW_00101](#), [SRS_BSW_00358](#), [SRS_BSW_00310](#))

8.3.3 Proxy Modules

The [Proxy Modules](#) partly implement the interfaces of the according AUTOSAR BSW modules. Those are listed in appendix D.

Following functions are provided in addition:

8.3.3.1 SwCluC_OsProxy_Init

[SWS_SwCluC_12000]{DRAFT} [

Service Name	SwCluC_OsProxy_Init_<SwCluCOsProxyOsBaseSocket> (draft)
Syntax	<pre>void SwCluC_OsProxy_Init_<SwCluCOsProxyOsBaseSocket> (void)</pre>
Service ID [hex]	0x30
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	SwCluC_OsProxy_Init is intended to allocate and initialize system resources used by the Os Proxy linked to this specific socketed Tags: atp.Status=draft
Available via	SwCluC.h

]([SRS_BSW_00101](#), [SRS_BSW_00358](#), [SRS_BSW_00310](#))

The name part <SwCluCnVMProxyOsBaseSocket> is the [shortName](#) of the [SwCluCnVMProxyOsBaseSocket](#).

8.3.3.2 SwCluC_NvMProxy_Init

[SWS_SwCluC_12100]{DRAFT} [

Service Name	SwCluC_NvMProxy_Init_<SwCluCNvMBaseSocket> (draft)
Syntax	void SwCluC_NvMProxy_Init_<SwCluCNvMBaseSocket> (void)
Service ID [hex]	0x31
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	SwCluC_NvMProxy_Init is intended to allocate and initialize system resources used by the NvMProxy linked to this specific socket. Tags: atp.Status=draft
Available via	SwCluC.h

]([SRS_BSW_00101](#), [SRS_BSW_00358](#), [SRS_BSW_00310](#))

The name part <SwCluCNvMBaseSocket> is the [shortName](#) of the [SwCluCNvMBaseSocket](#).

8.3.3.3 SwCluC_LdComProxy_Init

[SWS_SwCluC_13000]{DRAFT} [

Service Name	SwCluC_LdComProxy_Init_<SwCluCLdComBaseSocket> (draft)
Syntax	void SwCluC_LdComProxy_Init_<SwCluCLdComBaseSocket> (void)
Service ID [hex]	0x32
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None





Description	SwCluC_LdComProxy_Init is intended to allocate and initialize system resources used by the LdComProxy linked to this specific socket. Tags: atp.Status=draft
Available via	SwCluC.h

]([SRS_BSW_00101](#), [SRS_BSW_00358](#), [SRS_BSW_00310](#))

The name part <SwCluC_LdComProxyBaseSocket> is the [shortName](#) of the [SwCluC_LdComProxyBaseSocket](#).

8.3.3.4 SwCluC_ComProxy_Init

[SWS_SwCluC_13001]{DRAFT} [

Service Name	SwCluC_ComProxy_Init_<SwCluCComBaseSocket> (draft)
Syntax	void SwCluC_ComProxy_Init_<SwCluCComBaseSocket> (void)
Service ID [hex]	0x33
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	SwCluC_ComProxy_Init is intended to allocate and initialize system resources used by the ComProxy linked to this specific socket. Tags: atp.Status=draft
Available via	SwCluC.h

]([SRS_BSW_00101](#), [SRS_BSW_00358](#), [SRS_BSW_00310](#))

The name part <SwCluCComProxyBaseSocket> is the [shortName](#) of the [SwCluC-ComProxyBaseSocket](#).

8.3.3.5 SwCluC_DcmProxy_Init

[SWS_SwCluC_12200]{DRAFT} [

Service Name	SwCluC_DcmProxy_Init_<SwCluCDcmBaseSocket> (draft)
Syntax	void SwCluC_DcmProxy_Init_<SwCluCDcmBaseSocket> (void)





Service ID [hex]	0x34
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	SwCluC_DcmProxy_Init is intended to allocate and initialize system resources used by the Dcm Proxy linked to this specific socked. Tags: atp.Status=draft
Available via	SwCluC.h

|(SRS_BSW_00101, SRS_BSW_00358, SRS_BSW_00310)

The name part <SwCluC_DcmProxyBaseSocket> is the [shortName](#) of the [SwCluC_DcmProxyBaseSocket](#).

8.3.3.6 SwCluC_DemProxy_Init

[SWS_SwCluC_12300]{DRAFT} [

Service Name	SwCluC_DemProxy_Init_<SwCluCDemBaseSocket> (draft)
Syntax	<pre>void SwCluC_DemProxy_Init_<SwCluCDemBaseSocket> (void)</pre>
Service ID [hex]	0x35
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	SwCluC_DemProxy_Init is intended to allocate and initialize system resources used by the Dem Proxy linked to this specific socked. Tags: atp.Status=draft
Available via	SwCluC.h

|(SRS_BSW_00101, SRS_BSW_00358, SRS_BSW_00310)

The name part <SwCluCDemProxyBaseSocket> is the [shortName](#) of the [SwCluCDemProxyBaseSocket](#).

8.3.3.7 SwCluC_FiMProxy_Init

[SWS_SwCluC_12400]{DRAFT} [

Service Name	SwCluC_FiMProxy_Init_<SwCluCFiMBaseSocket> (draft)
Syntax	<pre>void SwCluC_FiMProxy_Init_<SwCluCFiMBaseSocket> (void)</pre>
Service ID [hex]	0x36
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	<p>SwCluC_FiMProxy_Init is intended to allocate and initialize system resources used by the FiM Proxy linked to this specific socket.</p> <p>Tags: atp.Status=draft</p>
Available via	SwCluC.h

|(SRS_BSW_00101, SRS_BSW_00358, SRS_BSW_00310)

The name part <SwCluCFiMProxyBaseSocket> is the [shortName](#) of the [SwCluC-FiMProxyBaseSocket](#).

8.4 Callback notifications

This is a list of functions provided for other modules.

8.4.1 Binary Manifest

The [Binary Manifest](#) has no callback notifications.

8.4.2 Cross Cluster Connection

The [Cross Cluster Connection](#) implements the interfaces of a [Cross Software Cluster Communication Plug-In](#) as specified in document [9].

8.4.3 Proxy Modules

The [Proxy Modules](#) partly implement the callback notifications of the according AUTOSAR BSW modules. Those are listed in appendix [D](#).

8.4.3.0.1 Com Proxy

8.4.3.0.1.1 SwCluC_ComProxy_CbkRxAck

[SWS_SwCluC_13002]{DRAFT} [

Service Name	SwCluC_ComProxy_<BS>_CbkRxAck (draft)	
Syntax	<pre>void SwCluC_ComProxy_<BS>_CbkRxAck (CbkHandleIdType ComUserCbkJHandleId)</pre>	
Service ID [hex]	0x37	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ComUserCbkJHandleId	Com user callback handle Id
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	<p>This callback function indicates that the signal of the primitive data item/event is ready for reception. It is an implementation of a <ComUser_CbkRxAck> callback.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_ComProxy_Cbk.h	

]()

8.4.3.0.1.2 SwCluC_ComProxy_COMCbkJInv

[SWS_SwCluC_13003]{DRAFT} [

Service Name	SwCluC_ComProxy_<BS>_COMCbkJInv (draft)	
Syntax	<pre>void SwCluC_ComProxy_<BS>_COMCbkJInv (CbkHandleIdType ComUserCbkJHandleId)</pre>	
Service ID [hex]	0x38	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ComUserCbkJHandleId	Com user callback handle Id
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	<p>This callback function indicates that COM has received a signal and parsed it as "invalid". It is an implementation of a <ComUser_CbkJInv> callback.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_ComProxy_Cbk.h	

]()

8.4.3.0.1.3 SwCluC_ComProxy_COMCbKRxTOut

[SWS_SwCluC_13004]{DRAFT} [

Service Name	SwCluC_ComProxy_<BS>_COMCbKRxTOut (draft)	
Syntax	<pre>void SwCluC_ComProxy_<BS>_COMCbKRxTOut (CbkHandleIdType ComUserCbKHandleId)</pre>	
Service ID [hex]	0x39	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ComUserCbKHandleId	Com user callback handle Id
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	<p>This callback function indicates that the aliveTimeout after the last successful reception of the signal of the data item/event has expired (data element outdated). It is an implementation of a <ComUser_CbKRxTOut> callback.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_ComProxy_Cbk.h	

]()

8.4.3.0.1.4 SwCluC_ComProxy_COMCbKTack

[SWS_SwCluC_13005]{DRAFT} [

Service Name	SwCluC_ComProxy_<BS>_COMCbKTack (draft)	
Syntax	<pre>void SwCluC_ComProxy_<BS>_COMCbKTack (CbkHandleIdType ComUserCbKHandleId)</pre>	
Service ID [hex]	0x3a	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ComUserCbKHandleId	Com user callback handle Id
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	<p>This callback function indicates that the signal of the data item/event is already handed over by COM to the PDU router. It is an implementation of a <ComUser_CbKTxAck> callback.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_ComProxy_Cbk.h	

]()

8.4.3.0.1.5 SwCluC_ComProxy_COMCbKTErr

[SWS_SwCluC_13006]{DRAFT} [

Service Name	SwCluC_ComProxy_<BS>_COMCbKTErr (draft)	
Syntax	<pre>void SwCluC_ComProxy_<BS>_COMCbKTErr (CbkHandleIdType ComUserCbKHandleId)</pre>	
Service ID [hex]	0x3b	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ComUserCbKHandleId	Com user callback handle Id
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	<p>This callback function indicates that an error occurred when the signal of the data item/event was handed over by COM to the PDU router. It is an implementation of a <ComUser_CbKTxErr> callback.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_ComProxy_Cbk.h	

]()

8.4.3.0.1.6 SwCluC_ComProxy_COMCbKTxTOut

[SWS_SwCluC_13007]{DRAFT} [

Service Name	SwCluC_ComProxy_<BS>_COMCbKTxTOut (draft)	
Syntax	<pre>void SwCluC_ComProxy_<BS>_COMCbKTxTOut (CbkHandleIdType ComUserCbKHandleId)</pre>	
Service ID [hex]	0x3c	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ComUserCbKHandleId	Com user callback handle Id
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	<p>This callback function indicates that the timeout of TransmissionAcknowledgementRequest for sending the signal of the data item/event has expired. It is an implementation of a <ComUser_CbKTxTOut> callback.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_ComProxy_Cbk.h	

]()

8.4.3.0.2 LdCom Proxy

8.4.3.0.2.1 SwCluC_LdComProxy_LdComCbkJCopyRxData

[SWS_SwCluC_13008]{DRAFT} [

Service Name	SwCluC_LdComProxy_<BS>_LdComCbkJCopyRxData (draft)	
Syntax	<pre>BufReq_ReturnType SwCluC_LdComProxy_<BS>_LdComCbkJCopyRxData (CbkHandleIdType LdComUserCbkJHandleId, const PduInfoType* info, PduLengthType* bufferSizePtr)</pre>	
Service ID [hex]	0x3d	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same LdComUserCbkJHandleId, otherwise Reentrant	
Parameters (in)	LdComUserCbkJHandleId	LdCom user callback handle Id corresponding to the received I-PDU
	info	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). An SduLength of 0 can be used to query the current amount of available buffer in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
Parameters (inout)	None	
Parameters (out)	bufferSizePtr	Available receive buffer after data has been copied.
Return value	BufReq_ReturnType	BUFREQ_OK: Data copied successfully BUFREQ_E_NOT_OK: Data was not copied because an error occurred.
Description	<p>This function is called to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this function provides the next part of the I-PDU data. The size of the remaining data is written to the position indicated by bufferSizePtr.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_LdComProxy_Cbk.h	

]()

8.4.3.0.2.2 SwCluC_LdComProxy_LdComCbkJCopyTxData

[SWS_SwCluC_13009]{DRAFT} [

Service Name	SwCluC_LdComProxy_<BS>_LdComCbkJCopyTxData (draft)	
Syntax	<pre>BufReq_ReturnType SwCluC_LdComProxy_<BS>_LdComCbkJCopyTxData (CbkHandleIdType LdComUserCbkJHandleId, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)</pre>	
Service ID [hex]	0x3e	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same LdComUserCbkJHandleId, otherwise Reentrant	





Parameters (in)	LdComUserCbkJHandleId	LdCom user callback handle Id corresponding to the transmitted I-PDU.
	info	Provides the destination buffer (SduDataPtr) and the number of bytes to be copied (SduLength). If not enough transmit data is available, no data is copied by the upper layer module and BUFREQ_E_BUSY is returned. The lower layer module may retry the call. An SduLength of 0 can be used to indicate state changes in the retry parameter or to query the current amount of available data in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
	retry	Will not be handled by LdCom and its upper layer.
Parameters (inout)	None	
Parameters (out)	availableDataPtr	Indicates the remaining number of bytes that are available in the upper layer module's Tx buffer. availableDataPtr can be used by TP modules that support dynamic payload lengths (e.g. FrlsoTp) to determine the size of the following CFS.
Return value	BufReq_ReturnType	BUFREQ_OK: Data has been copied to the transmit buffer completely as requested. BUFREQ_E_BUSY: Request could not be fulfilled, because the required amount of Tx data is not available. The lower layer module may retry this call later on. No data has been copied. BUFREQ_E_NOT_OK: Data has not been copied. Request failed.
Description	This function is called to acquire the transmit data of an I-PDU segment (N-PDU). Each call to this function provides the next part of the I-PDU data unless retry->TpDataState is TP_DATA_RETRY. In this case the function restarts to copy the data beginning at the offset from the current position indicated by retry->TxTpDataCnt. The size of the remaining data is written to the position indicated by availableDataPtr Tags: atp.Status=draft	
Available via	SwCluC_LdComProxy_Cbk.h	

]()

8.4.3.0.2.3 SwCluC_LdComProxy_LdComCbkJRxIndication

[SWS_SwCluC_13010]{DRAFT} [

Service Name	SwCluC_LdComProxy_<BS>_LdComCbkJRxIndication (draft)	
Syntax	<pre>void SwCluC_LdComProxy_<BS>_LdComCbkJRxIndication (CbkHandleIdType LdComUserCbkJHandleId, const PduInfoType* PduInfoPtr)</pre>	
Service ID [hex]	0x3f	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same LdComUserCbkJHandleId, otherwise Reentrant	
Parameters (in)	LdComUserCbkJHandleId	LdCom user callback handle Id corresponding to the received PDU
	PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout)	None	





Parameters (out)	None
Return value	None
Description	Indication of a received PDU from a lower layer communication interface module. Tags: atp.Status=draft
Available via	SwCluC_LdComProxy_Cbk.h

]()

8.4.3.0.2.4 SwCluC_LdComProxy_LdComCbkJStartOfReception

[SWS_SwCluC_13015]{DRAFT} [

Service Name	SwCluC_LdComProxy_<BS>_LdComCbkJStartOfReception (draft)	
Syntax	<pre>BufReq_ReturnType SwCluC_LdComProxy_<BS>_LdComCbkJStartOfReception (CbkHandleIdType LdComUserCbkHandleId, const PduInfoType* info, PduLengthType TpSduLength, PduLengthType* bufferSizePtr)</pre>	
Service ID [hex]	0x40	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same LdComUserCbkHandleId, otherwise Reentrant	
Parameters (in)	LdComUserCbkHandleId	LdCom user callback handle Id corresponding to the I-PDU.
	info	Pointer to a PduInfoType structure containing the payload data (without protocol information) and payload length of the first frame or single frame of a transport protocol I-PDU reception, and the MetaData related to this PDU. If neither first/single frame data nor MetaData are available, this parameter is set to NULL_PTR.
	TpSduLength	Total length of the N-SDU to be received.
Parameters (inout)	None	
Parameters (out)	bufferSizePtr	Available receive buffer in the receiving module. This parameter will be used to compute the Block Size (BS) in the transport protocol module.
Return value	BufReq_ReturnType	<p>BUFREQ_OK: Connection has been accepted. bufferSizePtr indicates the available receive buffer; reception is continued. If no buffer of the requested size is available, a receive buffer size of 0 shall be indicated by bufferSizePtr.</p> <p>BUFREQ_E_NOT_OK: Connection has been rejected; reception is aborted. bufferSizePtr remains unchanged.</p> <p>BUFREQ_E_OVFL: No buffer of the required length can be provided; reception is aborted. bufferSizePtr remains unchanged.</p>
Description	<p>This function is called at the start of receiving an N-SDU. The N-SDU might be fragmented into multiple N-PDUs (FF with one or more following CFs) or might consist of a single N-PDU (SF). The service shall provide the currently available maximum buffer size when invoked with TpSdu Length equal to 0.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_LdComProxy_Cbk.h	

]()

8.4.3.0.2.5 SwCluC_LdComProxy_LdComCbkJpRxIndication

[SWS_SwCluC_13011]{DRAFT} [

Service Name	SwCluC_LdComProxy_<BS>_LdComCbkJpRxIndication (draft)	
Syntax	<pre>void SwCluC_LdComProxy_<BS>_LdComCbkJpRxIndication (CbkHandleIdType LdComUserCbkHandleId, Std_ReturnType result)</pre>	
Service ID [hex]	0x41	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same sn, otherwise Reentrant	
Parameters (in)	LdComUserCbkHandleId	LdCom user callback handle Id corresponding to the received I-PDU
	result	Result of the reception.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	<p>Called after an I-PDU has been received via the TP API, the result indicates whether the transmission was successful or not.</p> <p>Tags: atp.Status=draft</p>	
Available via	SwCluC_LdComProxy_Cbk.h	

]()

8.4.3.0.2.6 SwCluC_LdComProxy_LdComCbkJriggerTransmit

[SWS_SwCluC_13012]{DRAFT} [

Service Name	SwCluC_LdComProxy_<BS>_LdComCbkJriggerTransmit (draft)	
Syntax	<pre>Std_ReturnType SwCluC_LdComProxy_<BS>_LdComCbkJriggerTransmit (CbkHandleIdType LdComUserCbkHandleId, PduInfoType* PduInfoPtr)</pre>	
Service ID [hex]	0x42	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant for same LdComUserCbkHandleId, otherwise Reentrant	
Parameters (in)	LdComUserCbkHandleId	LdCom user callback handle Id corresponding to the ID of the SDU that is requested to be transmitted
	PduInfoPtr	Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLengh. On return, the service will indicate the length of the copied SDU data in SduLength.
Parameters (inout)	None	
Parameters (out)	None	





Return value	Std_ReturnType	E_OK: SDU has been copied and SduLength indicates the number of copied bytes. E_NOT_OK: No SDU data has been copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.
Description	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PduInfoPtr. Tags: atp.Status=draft	
Available via	SwCluC_LdComProxy_Cbk.h	

]()

8.4.3.0.2.7 SwCluC_LdComProxy_LdComCbkJxConfirmation

[SWS_SwCluC_13013]{DRAFT} [

Service Name	SwCluC_LdComProxy_<BS>_LdComCbkJxConfirmation (draft)	
Syntax	<pre>void SwCluC_LdComProxy_<BS>_LdComCbkJxConfirmation (CbkHandleIdType LdComUserCbkHandleId, Std_ReturnType result)</pre>	
Service ID [hex]	0x43	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same LdComUserCbkHandleId, otherwise Reentrant	
Parameters (in)	LdComUserCbkHandleId	LdCom user callback handle Id corresponding to 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. Tags: atp.Status=draft	
Available via	SwCluC_LdComProxy_Cbk.h	

]()

8.4.3.0.2.8 SwCluC_LdComProxy_LdComCbkJpTxConfirmation

[SWS_SwCluC_13014]{DRAFT} [

Service Name	SwCluC_LdComProxy_<BS>_LdComCbkJpTxConfirmation (draft)	
Syntax	<pre>void SwCluC_LdComProxy_<BS>_LdComCbkJpTxConfirmation (CbkHandleIdType LdComUserCbkJandleId, Std_ReturnType result)</pre>	
Service ID [hex]	0x44	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same LdComUserCbkJandleId, otherwise Reentrant	
Parameters (in)	LdComUserCbkJandleId	LdCom user callback handle Id corresponding to the transmitted I-PDU
	result	E_OK - transmission successful E_NOT_OK - transmission not successful
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This function is called after a Signal has been transmitted via the TP-API on its network. Tags: atp.Status=draft	
Available via	SwCluC_LdComProxy_Cbk.h	

}]()

8.5 Scheduled functions

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

8.5.1 Binary Manifest

The [Binary Manifest](#) has no scheduled functions.

8.5.2 Cross Cluster Communication Scheduled functions

The [Cross Cluster Communications](#) has no standardized scheduled functions but is free to implement some if functionally needed.

8.5.3 Proxy Modules Scheduled functions

The [Proxy Modules](#) have no standardized scheduled functions but are free to implement some if functionally needed.

8.6 Expected interfaces

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

8.6.1 Mandatory interfaces

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

8.6.1.1 Binary Manifest

The [Binary Manifest](#) has no mandatory interfaces.

8.6.1.2 Cross Cluster Connection

The [Cross Cluster Connection](#) has no mandatory interfaces.

8.6.1.3 Proxy Modules

The [Proxy Modules](#) partly implement the mandatory interfaces of the according AUTOSAR BSW modules. Those are listed in [appendix D](#).

8.6.2 Optional interfaces

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

8.6.2.1 Binary Manifest

The [Binary Manifest](#) has no optional interfaces.

8.6.2.2 Cross Cluster Connection

The [Cross Cluster Connection](#) has no optional interfaces.

8.6.2.3 Proxy Modules

The [Proxy Modules](#) partly implement the optional interfaces of the according AUTOSAR BSW modules. Those are listed in appendix [D](#).

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 callback function. The names of this kind of interfaces are not fixed because they are configurable.

8.6.3.1 Binary Manifest

The [Binary Manifest](#) has no configurable interfaces.

8.6.3.2 Cross Cluster Connection

8.6.3.3 Proxy Modules

The [Proxy Modules](#) partly implement the configurable interfaces of the according AUTOSAR BSW modules. Those are listed in appendix [D](#).

8.7 Service Interfaces

8.7.1 Binary Manifest

The [Binary Manifest](#) has no Service Interfaces.

8.7.2 Cross Cluster Connection

The `Cross Cluster Connection` has no Service Interfaces.

8.7.3 Proxy Modules

The [Proxy Modules](#) only implement Service Interfaces, which are already defined by the original AUTOSAR Service. Those are listed in appendix [E](#).

9 Sequence diagrams

No sequence diagrams are contained in this section.

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter 10.1 in the specification to guarantee comprehension.

Chapter 10.2 specifies the structure (containers) and the parameters of the module [SwCluC](#).

Chapter 10.3 specifies published information of the module [SwCluC](#).

10.1 How to read this chapter

For details, refer to the chapter 10.1 “Introduction to configuration specification” in [SWS_BSWGeneral](#).

10.2 Containers and configuration parameters

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

10.2.1 Module Configuration

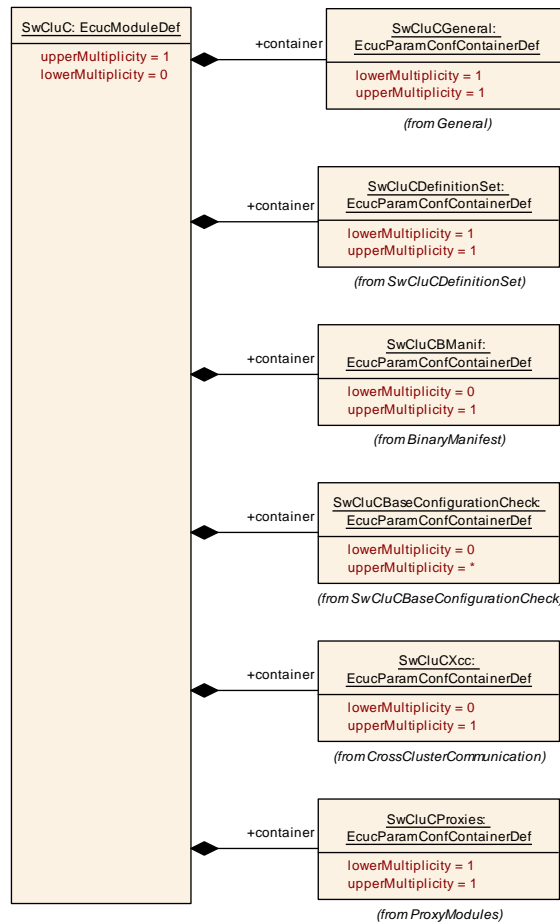


Figure 10.1: SwCluC configuration overview

Module SWS Item	ECUC_SwCluC_00001	
Module Name	SwCluC	
Module Description	Module to collect Software Cluster Connection specific configuration information.	
Post-Build Variant Support	true	
Supported Config Variants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE	
Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCBManif	0..1	Configuration of the Binary Manifest of the Software Cluster Connection Tags: atp.Status=draft

Container Name	Multiplicity	Scope / Dependency
SwCluCBaseConfiguration Check	0..*	Configuration of the base configuration check of Software Cluster Connection. This configuration places a mandatory entry in the Binary Manifest where the guard value is used to proof if the general setup of the Applicative Software Cluster is compatible to the general setup of the Host Software Cluster. For each Applicative Software Cluster an individual base configuration check should be defined. Tags: atp.Status=draft
SwCluCDefinitionSet	1	Definition of a set of Software Clusters Tags: atp.Status=draft
SwCluCGeneral	1	General configuration of the Software Cluster Connection Tags: atp.Status=draft
SwCluCProxies	1	General configuration of the Proxy Modules of Software Cluster Connection Tags: atp.Status=draft
SwCluCXcc	0..1	Configuration of the Binary Manifest of the Software Cluster Connection Tags: atp.Status=draft

10.2.2 General configuration parameters

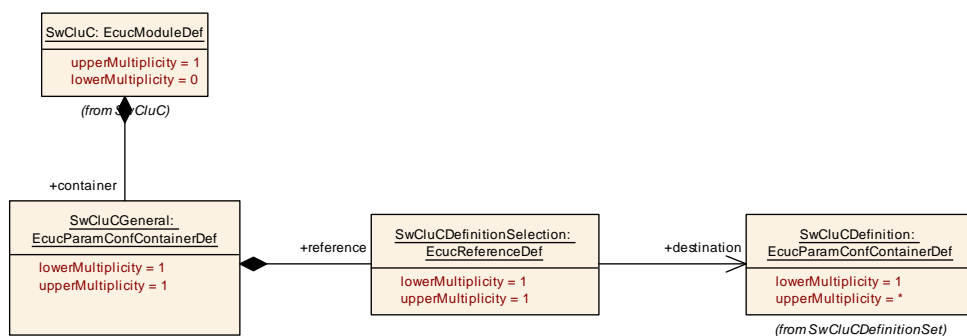


Figure 10.2: SwCluC General configuration parameters

SWS Item	[ECUC_SwCluC_00002]
Container Name	SwCluCGeneral
Parent Container	SwCluC

Description	General configuration of the Software Cluster Connection		
	Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCDefinitionSelection [ECUC_SwCluC_00005]		
Parent Container	SwCluCGeneral		
Description	This reference selects the Software Cluster Definition which is applied for this Software Cluster.		
	Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to SwCluCDefinition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.3 Software Cluster Definition

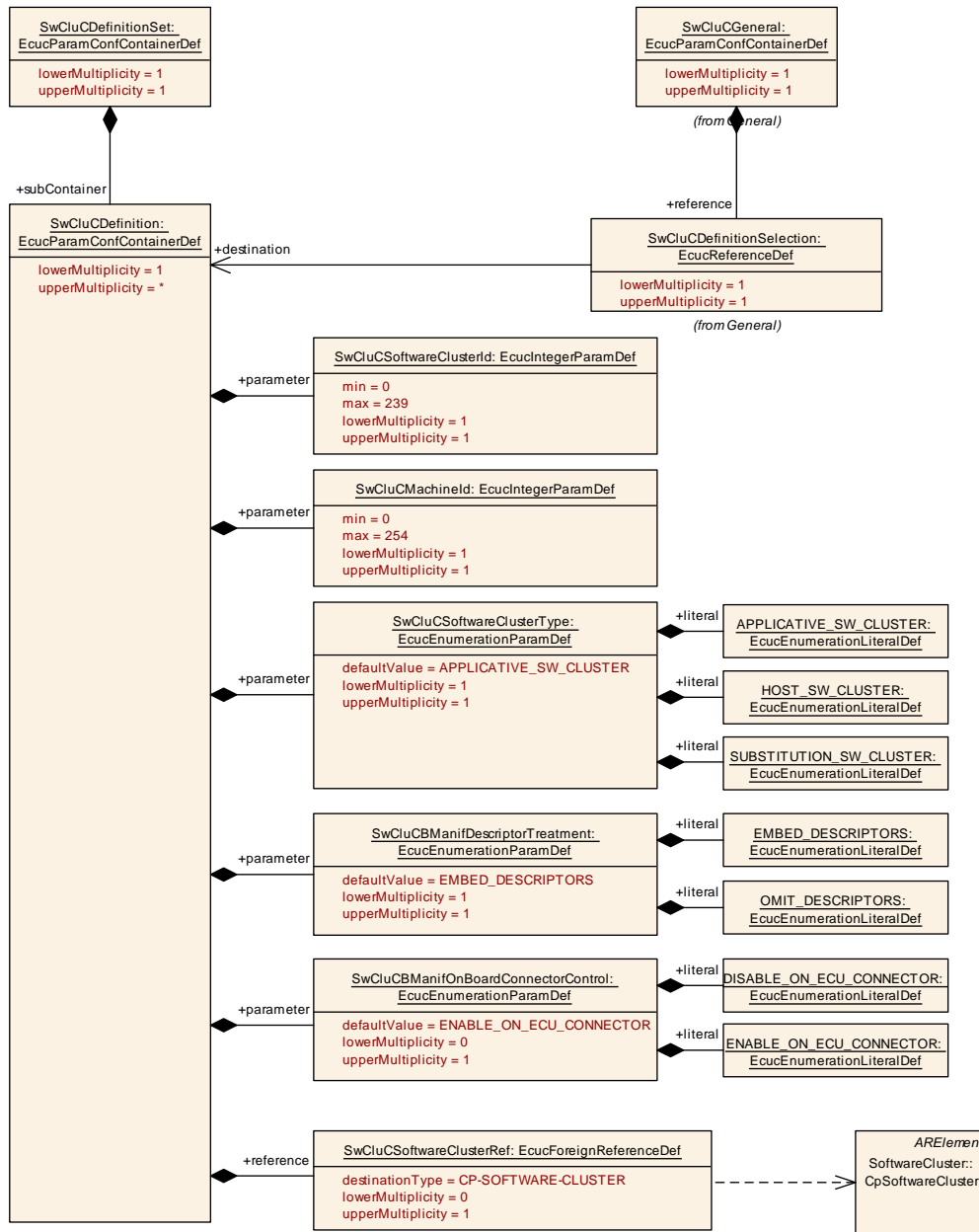


Figure 10.3: SwCluC Software Cluster Definition

SWS Item	[ECUC_SwCluC_00004]
Container Name	SwCluCDefinitionSet
Parent Container	SwCluC
Description	Definition of a set of Software Clusters
	Tags: atp.Status=draft
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCDefinition	1..*	<p>Definition of the general properties of the Software Cluster. The definitions are separated in a set of definitions in order to enable an holistic view about all existing Software Cluster Definitions.</p> <p>Tags: atp.Status=draft</p>

SWS Item	[ECUC_SwCluC_00006]		
Container Name	SwCluCDefinition		
Parent Container	SwCluCDefinitionSet		
Description	<p>Definition of the general properties of the Software Cluster. The definitions are separated in a set of definitions in order to enable an holistic view about all existing Software Cluster Definitions.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCBManifDescriptorTreatment [ECUC_SwCluC_00010]	
Parent Container	SwCluCDefinition	
Description	<p>Configures the existence of the Interface Descriptor Table in the Binary Object of the Software Cluster.</p> <p>Tags: atp.Status=draft</p>	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	EMBED_DESCRIPTOR	<p>The Interface Descriptor Table of the Binary Manifest is embed into the Binary Object.</p> <p>Tags: atp.Status=draft</p>
	OMIT_DESCRIPTOR	<p>The Interface Descriptor Table of the Binary Manifest is omitted from the Binary Object.</p> <p>The information needs to be delivered as CpSoftwareClusterBinaryManifest-Descriptor.</p> <p>Tags: atp.Status=draft</p>
Default Value	EMBED_DESCRIPTOR	

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCManifestOnBoardConnectorControl [ECUC_SwCluC_00011]		
Parent Container	SwCluCDefinition		
Description	SwCluCManifestOnBoardConnectorControl enables or disables the onECU execution of the Software Cluster Connector. This setting is only applicable for the Host Software Cluster. Tags: atp.Status=draft		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	DISABLE_ON_ECU_CONNECTOR		The connection of Software Clusters OnECU is disabled for this machine. Tags: atp.Status=draft
	ENABLE_ON_ECU_CONNECTOR		The connection of Software Clusters OnECU is enabled for this machine. Tags: atp.Status=draft
Default Value	ENABLE_ON_ECU_CONNECTOR		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCMachineId [ECUC_SwCluC_00008]		
Parent Container	SwCluCDefinition		
Description	<p>Unique number of the (virtual or physical) machine to which the Software Cluster belongs.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 254		
Default Value			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCSoftwareClusterId [ECUC_SwCluC_00007]		
Parent Container	SwCluCDefinition		
Description	<p>Unique number of the Software Cluster</p> <p>Numbers >= 0xFE are reserved to indicate special values or the unconnected state.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 239		
Default Value			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCSoftwareClusterType [ECUC_SwCluC_00009]		
Parent Container	SwCluCDefinition		
Description	The type of the Software Cluster Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	APPLICATIVE_SW_CLUSTER		This Software Cluster is an Applicative Software Cluster. Tags: atp.Status=draft
	HOST_SW_CLUSTER		This Software Cluster is a Host Software Cluster. Tags: atp.Status=draft
	SUBSTITUTION_SW_CLUSTER		This Software Cluster is a Substitution Software Cluster. Tags: atp.Status=draft
Default Value	APPLICATIVE_SW_CLUSTER		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCSoftwareClusterRef [ECUC_SwCluC_00034]		
Parent Container	SwCluCDefinition		
Description	Reference to the CpSoftwareCluster to define the link to the Software Cluster description in the System. Tags: atp.Status=draft		
Multiplicity	0..1		
Type	Foreign reference to CP-SOFTWARE-CLUSTER		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.4 Software Cluster Base Configuration Check

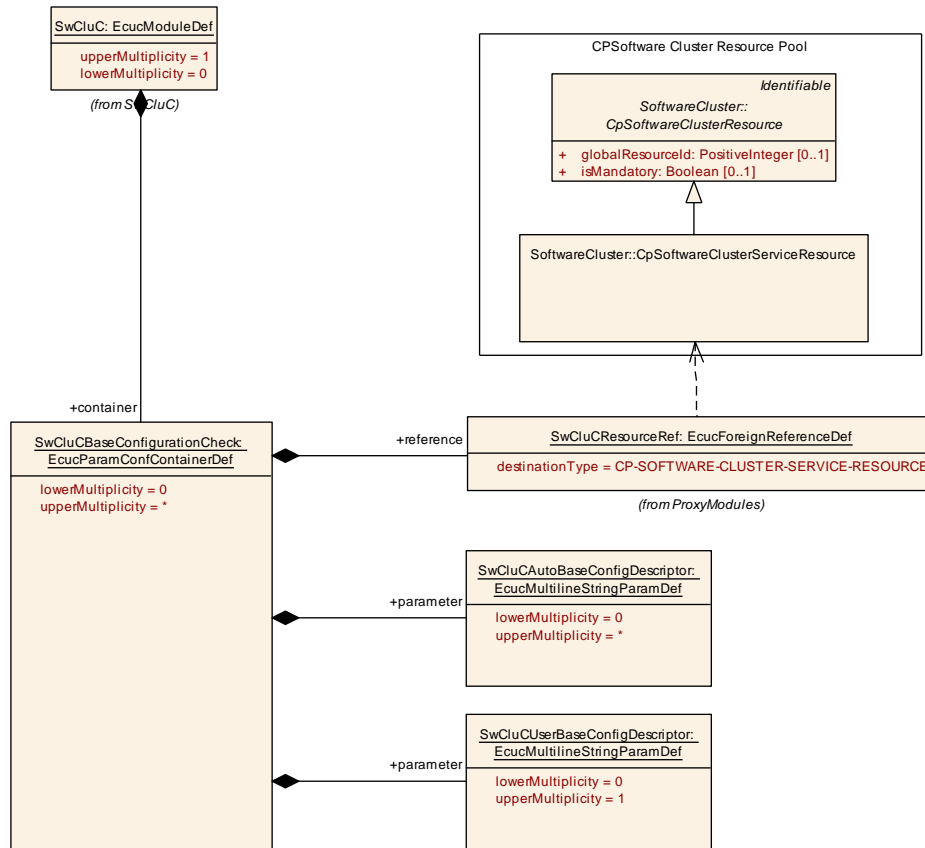


Figure 10.4: Software Cluster Base Configuration Check

SWS Item	[ECUC_SwCluC_00079]		
Container Name	SwCluCBaseConfigurationCheck		
Parent Container	SwCluC		
Description	<p>Configuration of the base configuration check of Software Cluster Connection. This configuration places a mandatory entry in the Binary Manifest where the guard value is used to proof if the general setup of the Applicative Software Cluster is compatible to the general setup of the Host Software Cluster. For each Applicative Software Cluster an individual base configuration check should be defined.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCAutoBaseConfigDescriptor [ECUC_SwCluC_00080]		
Parent Container	SwCluCBaseConfigurationCheck		
Description	<p>A machine determined string which represent the basic configuration assumption. Multiple SwCluCAutoBaseConfigDescriptor values can be added to support that different tools contribute to the checksum. For instance Software Cluster Connection Tool, Compiler, Link, or Locate tooling. isAuto shall be set to true since tool chain adds its collected values automatically.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..*		
Type	EcucMultilineStringParamDef		
Default Value			
Regular Expression			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCUserBaseConfigDescriptor [ECUC_SwCluC_00081]		
Parent Container	SwCluCBaseConfigurationCheck		
Description	<p>A user determined string which represent the basic configuration assumption.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucMultilineStringParamDef		
Default Value			
Regular Expression			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00093]		
Parent Container	SwCluCBaseConfigurationCheck		

Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_00078]{DRAFT} [SwCluCBaseConfigurationCheck](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_BASE_CNF](#) [The [SwCluCBaseConfigurationCheck.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_BASE_CNF](#).]()

10.2.5 Binary Manifest

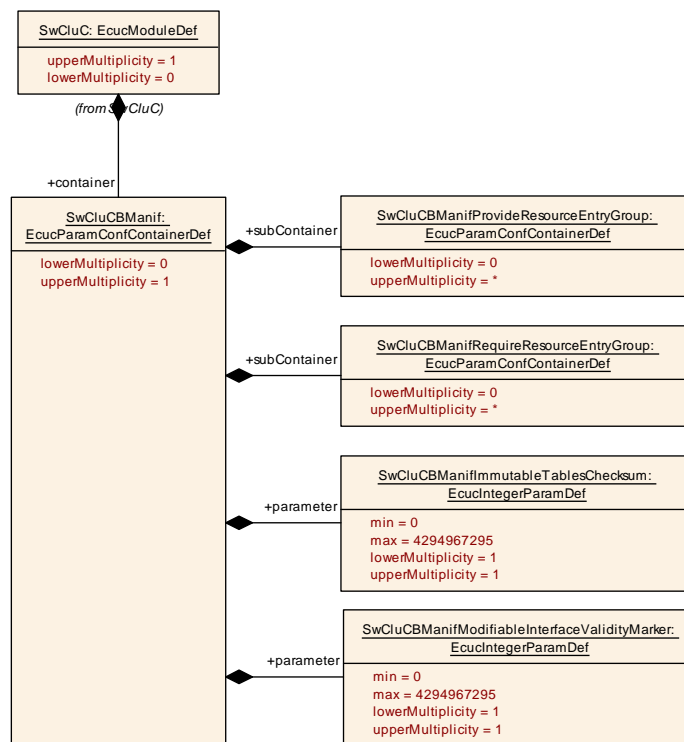


Figure 10.5: SwCluC Binary Manifest

SWS Item	[ECUC_SwCluC_00003]
Container Name	SwCluCBManif
Parent Container	SwCluC

Description	Configuration of the Binary Manifest of the Software Cluster Connection Tags: atp.Status=draft
Configuration Parameters	

Name	SwCluCBManifImmutableTablesChecksum [ECUC_SwCluC_00020]		
Parent Container	SwCluCBManif		
Description	This parameter defines the initialization value of the Immutable Tables Checksum. If the Binary Manifest code generator already calculates the checksum, the value can be stored here (withAuto == true). Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCBManifModifiableInterfaceValidityMarker [ECUC_SwCluC_00022]		
Parent Container	SwCluCBManif		
Description	This parameter defines the initialization value of the Modifiable Interface Validity Marker. The init value should be set to the erased value of the flash memory. Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCBManifProvideResourceEntryGroup	0..*	<p>Describes a group of provided resources in the Binary Manifest. The belonging handles are put into the Binary Manifest's tables into an consecutive order which supports an array based access.</p> <p>Tags: atp.Status=draft</p>
SwCluCBManifRequireResourceEntryGroup	0..*	<p>Describes a group of required resources in the Binary Manifest. The belonging handles are put into the Binary Manifest's tables into an consecutive order which supports an array based access.</p> <p>Tags: atp.Status=draft</p>
SwCluCBManifResourceType	0..*	<p>This container defines the structure of a resource type in the Binary Manifest.</p> <p>Tags: atp.Status=draft</p>

10.2.5.1 Provide Resource Entry Group

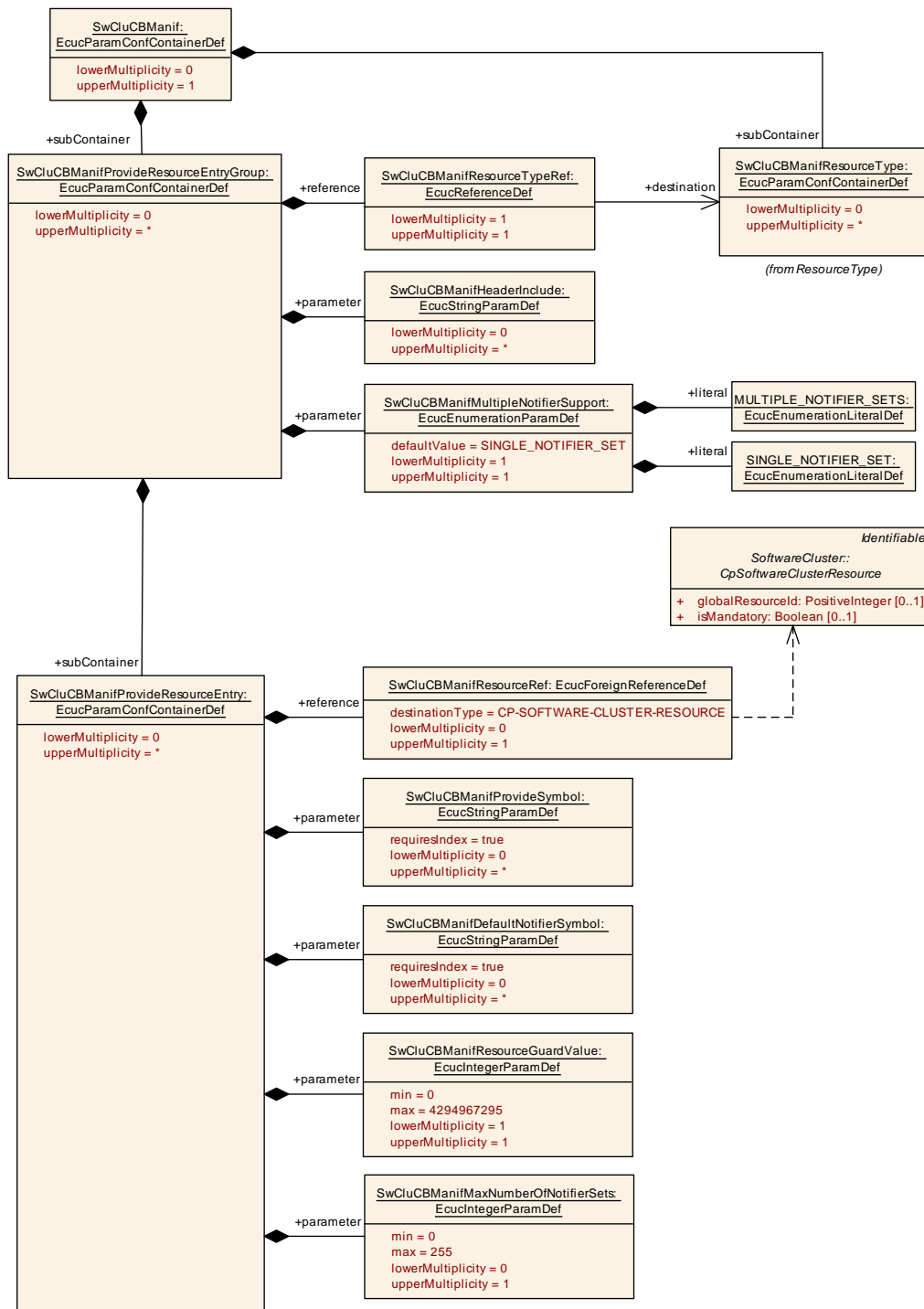


Figure 10.6: SwCluC Provide Resource Entry Group

SWS Item	[ECUC_SwCluC_00018]
Container Name	SwCluCBManifProvideResourceEntryGroup
Parent Container	SwCluCBManif

Description	<p>Describes a group of provided resources in the Binary Manifest. The belonging handles are put into the Binary Manifest's tables into an consecutive order which supports an array based access.</p> <p>Tags: atp.Status=draft</p>
Configuration Parameters	

Name	SwCluCBManifHeaderInclude [ECUC_SwCluC_00024]		
Parent Container	SwCluCBManifProvideResourceEntryGroup		
Description	<p>Defines the header file(s) where the owner of the SwCluCBManifProvideResourceEntryGroup / SwCluCBManifRequireResourceEntryGroup has the declarations of the symbols for the handle initialization.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..*		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	SwCluCBManifMultipleNotifierSupport [ECUC_SwCluC_00016]	
Parent Container	SwCluCBManifProvideResourceEntryGroup	
Description	<p>SwCluCBManifMultipleNotifierSupport defines whether multiple Software Cluster can register on the notifier handles. For each possible connection a set of notifier handles is allocated.</p> <p>Tags: atp.Status=draft</p>	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	MULTIPLE_NOTIFIER_S ETS	<p>The provided resource supports the co-resident notifier connections to multiple required resources.</p> <p>Tags: atp.Status=draft</p>

Default Value	SINGLE_NOTIFIER_SET	The provided resource supports at most one connection of notifiers to a required resource.	
	SINGLE_NOTIFIER_SET		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	SwCluCBManifResourceTypeRef [ECUC_SwCluC_00023]
Parent Container	SwCluCBManifProvideResourceEntryGroup
Description	Defines the resource type for all SwCluCBManifProvideResourceEntry / SwCluCBManifRequireResourceEntry Tags: atp.Status=draft
Multiplicity	1
Type	Reference to SwCluCBManifResourceType
Scope / Dependency	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCBManifProvideResourceEntry	0..*	Describes a single entry for a provided resource in the Binary Manifest. Tags: atp.Status=draft

10.2.5.2 Provided Resource Entry

SWS Item	[ECUC_SwclC_00009]
Container Name	SwCluCBManifProvideResourceEntry
Parent Container	SwCluCBManifProvideResourceEntryGroup
Description	Describes a single entry for a provided resource in the Binary Manifest. Tags: atp.Status=draft
Configuration Parameters	

Name	SwCluCBManifDefaultNotifierSymbol [ECUC_SwCluC_00027]
Parent Container	SwCluCBManifProvideResourceEntry
Description	<p>SwCluCBManifDefaultNotifierSymbol set the default value of a Notifier Handle put into the Modifiable Interface. The number and order of SwCluCBManifDefaultNotifierSymbol needs to match the number and order of SwCluCBManifNotifierHandle in the referenced SwCluCBManifResourceType.</p> <p>The parameter is defined as string in order to support the usage of symbols as initialize. Values have to be set as string in a C supported number format, e.g. 0xC001CAFE. In case the provide handle holds a pointer the address operator is part of the string, e.g. &myExampleFunction</p> <p>Tags: atp.Status=draft</p> <p>Attributes: requiresIndex=true</p>
Multiplicity	0..*
Type	EcucStringParamDef
Default Value	
Regular Expression	
Scope / Dependency	

Name	SwCluCBManifMaxNumberOfNotifierSets [ECUC_SwCluC_00029]		
Parent Container	SwCluCBManifProvideResourceEntry		
Description	<p>SwCluCBManifMaxNumberOfNotifierSets defines the maximum number of possible required resources using notifier handles. It is required in case SwCluCBManifMultipleNotifierSupport is set to MULTIPLE_NOTIFIER_SETS.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	SwCluCBManifProvideSymbol [ECUC_SwCluC_00026]
Parent Container	SwCluCBManifProvideResourceEntry
Description	<p>SwCluCBManifProvideSymbol set the value of a Provide Handle put into the Immutable Interface. The number and order of SwCluCBManifProvideSymbols needs to match the number and order of SwCluCBManifProvideHandle in the referenced SwCluCBManifResourceType.</p> <p>The parameter is defined as string in order to support the usage of symbols as initialize. Values have to be set as string in a C supported number format, e.g. 0xC001CAFE. In case the provide handle holds a pointer the address operator is part of the string, e.g. &myExampleFunction</p> <p>Tags: atp.Status=draft</p> <p>Attributes: requiresIndex=true</p>
Multiplicity	0..*
Type	EcucStringParamDef
Default Value	
Regular Expression	
Scope / Dependency	

Name	SwCluCBManifResourceGuardValue [ECUC_SwCluC_00028]
Parent Container	SwCluCBManifProvideResourceEntry
Description	<p>SwCluCBManifResourceGuardValue provides guarding information checked by the Software Cluster connector. A provided resources and required resources of Software Clusters can only be connected, if the resource guarding values of provider and requester are equal.</p> <p>Tags: atp.Status=draft</p>
Multiplicity	1
Type	EcucIntegerParamDef
Range	0 .. 4294967295
Default Value	
Scope / Dependency	

Name	SwCluCBManifResourceRef [ECUC_SwCluC_00025]
Parent Container	SwCluCBManifProvideResourceEntry
Description	<p>Reference to the CpSoftwareClusterResource determining the global resource Id. If the reference is not set, the global resource Id in the Binary Manifest will be set to 0 for this resource entry in order to indicate an invalid resource.</p> <p>Tags: atp.Status=draft</p>
Multiplicity	0..1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-RESOURCE
Scope / Dependency	

No Included Containers

10.2.5.3 Require Resource Entry Group

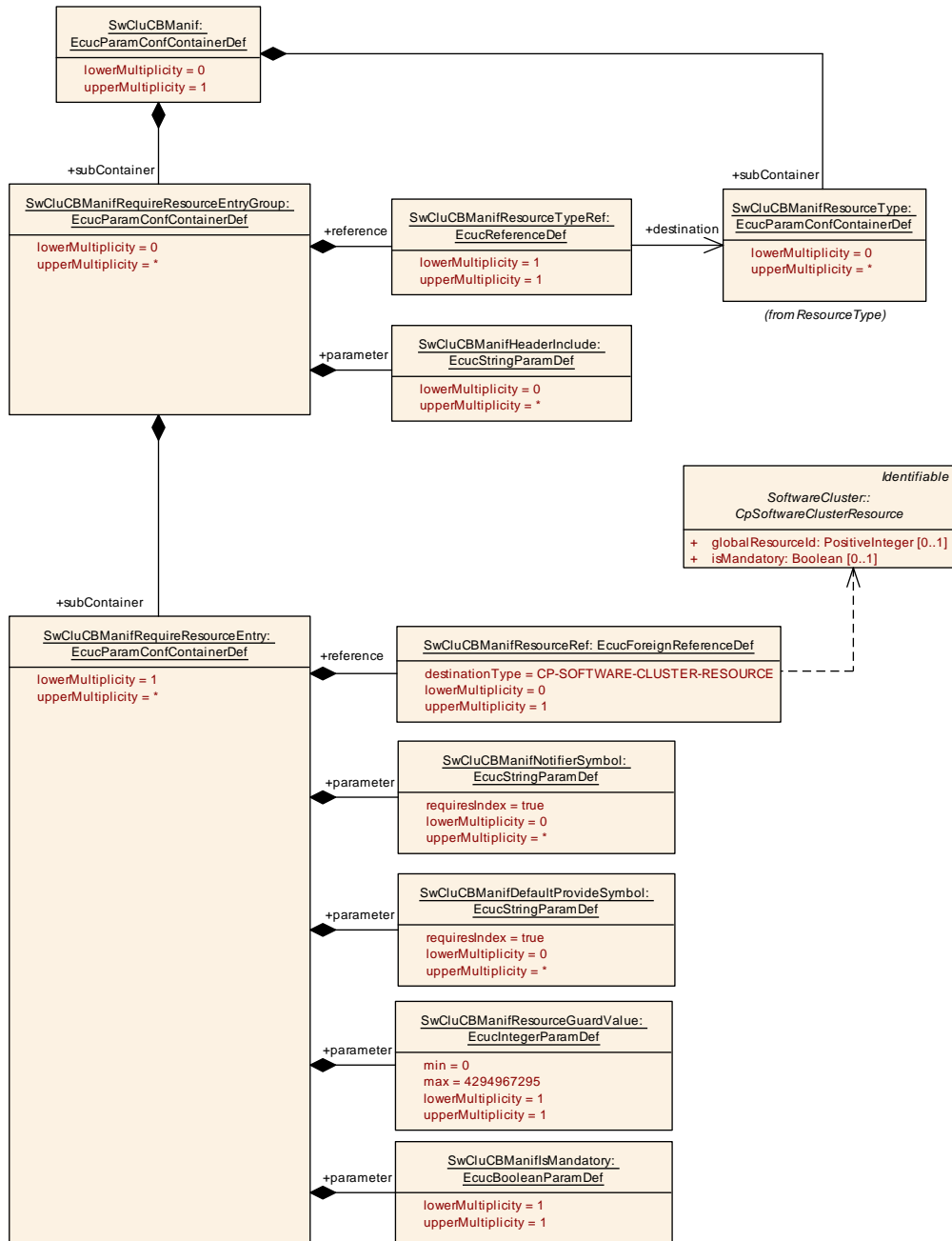


Figure 10.7: SwCluC Require Resource Entry Group

SWS Item	[ECUC_SwCluC_00019]
Container Name	SwCluCBManifRequireResourceEntryGroup
Parent Container	SwCluCBManif

Description	<p>Describes a group of required resources in the Binary Manifest. The belonging handles are put into the Binary Manifest's tables into an consecutive order which supports an array based access.</p> <p>Tags: atp.Status=draft</p>
Configuration Parameters	

Name	SwCluCBManifHeaderInclude [ECUC_SwCluC_00024]		
Parent Container	SwCluCBManifRequireResourceEntryGroup		
Description	<p>Defines the header file(s) where the owner of the SwCluCBManifProvideResourceEntryGroup / SwCluCBManifRequireResourceEntryGroup has the declarations of the symbols for the handle initialization.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..*		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	SwCluCBManifResourceTypeRef [ECUC_SwCluC_00023]		
Parent Container	SwCluCBManifRequireResourceEntryGroup		
Description	<p>Defines the resource type for all SwCluCBManifProvideResourceEntry / SwCluCBManifRequireResourceEntry</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	Reference to SwCluCBManifResourceType		
Scope / Dependency			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCBManifRequireResourceEntry	1..*	Describes a single entry for a required resource in the Binary Manifest. Tags: atp.Status=draft

10.2.5.4 Require Resource Entry

SWS Item	[ECUC_SwclC_00021]
Container Name	SwCluCBManifRequireResourceEntry
Parent Container	SwCluCBManifRequireResourceEntryGroup
Description	Describes a single entry for a required resource in the Binary Manifest. Tags: atp.Status=draft
Configuration Parameters	

Name	SwCluCBManifDefaultProvideSymbol [ECUC_SwCluC_00031]
Parent Container	SwCluCBManifRequireResourceEntry
Description	SwCluCBManifDefaultProvideSymbol set the default value of a Provide Handle put into the Modifiable Interface. The number and order of SwCluCBManifDefaultProvideSymbols needs to match the number and order of SwclCBManifProvideHandles in the referenced SwCluCBManifResourceType. The parameter is defined as string in order to support the usage of symbols as initialize. Values have to be set as string in a C supported number format, e.g. 0xC001CAFE. In case the provide handle holds a pointer the address operator is part of the string, e.g. &myExampleFunction Tags: atp.Status=draft Attributes: requiresIndex=true
Multiplicity	0..*
Type	EcucStringParamDef
Default Value	
Regular Expression	
Scope / Dependency	

Name	SwCluCBManifIsMandatory [ECUC_SwCluC_00032]
Parent Container	SwCluCBManifRequireResourceEntry
Description	<p>SwCluCBManifIsMandatory indicates, that the resource is mandatory to operate this Software Cluster. If the resource is not provided on the machine the connection process for this Software Cluster requiring this resource gets aborted.</p> <p>Tags: atp.Status=draft</p>
Multiplicity	1
Type	EcucBooleanParamDef
Default Value	
Scope / Dependency	

Name	SwCluCBManifNotifierSymbol [ECUC_SwCluC_00030]
Parent Container	SwCluCBManifRequireResourceEntry
Description	<p>SwCluCBManifNotifierSymbol set the value of a Notifier Handle put into the Immutable Interface. The number and order of SwCluCBManifNotifierSymbol needs to match the number and order of SwCluCBManifNotifierHandles in the referenced SwCluCBManifResourceType.</p> <p>The parameter is defined as string in order to support the usage of symbols as initialize. Values have to be set as string in a C supported number format, e.g. 0xC001CAFE. In case the provide handle holds a pointer the address operator is part of the string, e.g. &myExampleFunction</p> <p>Tags: atp.Status=draft</p> <p>Attributes: requiresIndex=true</p>
Multiplicity	0..*
Type	EcucStringParamDef
Default Value	
Regular Expression	
Scope / Dependency	

Name	SwCluCBManifResourceGuardValue [ECUC_SwCluC_00028]
Parent Container	SwCluCBManifRequireResourceEntry
Description	<p>SwCluCBManifResourceGuardValue provides guarding information checked by the Software Cluster connector. A provided resources and required resources of Software Clusters can only be connected, if the resource guarding values of provider and requester are equal.</p> <p>Tags: atp.Status=draft</p>
Multiplicity	1
Type	EcucIntegerParamDef
Range	0 .. 4294967295
Default Value	

Scope / Dependency	
---------------------------	--

Name	SwCluCBManifResourceRef [ECUC_SwCluC_00025]
Parent Container	SwCluCBManifRequireResourceEntry
Description	<p>Reference to the CpSoftwareClusterResource determining the global resource Id. If the reference is not set, the global resource Id in the Binary Manifest will be set to 0 for this resource entry in order to indicate an invalid resource.</p> <p>Tags: atp.Status=draft</p>
Multiplicity	0..1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-RESOURCE
Scope / Dependency	

No Included Containers

10.2.5.5 Resource Type

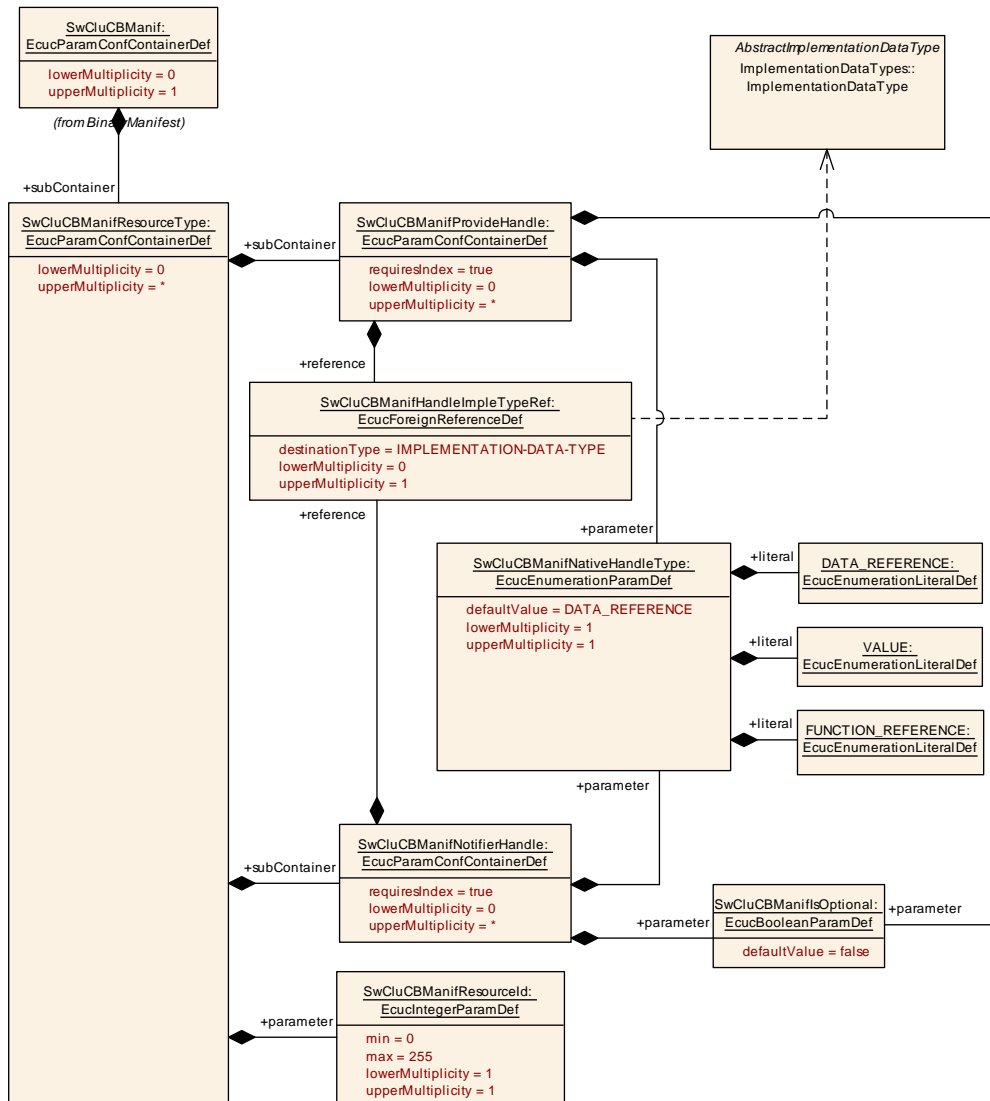


Figure 10.8: SwCluC resource type

SWS Item	[ECUC_SwCluC_00012]		
Container Name	SwCluCBManifResourceType		
Parent Container	SwCluCBManif		
Description	This container defines the structure of a resource type in the Binary Manifest. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCBManifResourceId [ECUC_SwCluC_00017]		
Parent Container	SwCluCBManifResourceType		
Description	Unique number of the resource type. Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCBManifNotifier Handle	0..*	Defines a Notifier Handle and its properties for this resource type. Tags: atp.Status=draft
SwCluCBManifProvide Handle	0..*	Defines a Provide Handle and its properties for this resource type. Tags: atp.Status=draft

SWS Item	[ECUC_SwCluC_00013]
Container Name	SwCluCBManifProvideHandle
Parent Container	SwCluCBManifResourceType
Description	Defines a Provide Handle and its properties for this resource type. Tags: atp.Status=draft Attributes: requiresIndex=true
Configuration Parameters	

Name	SwCluCBManifIsOptional [ECUC_SwCluC_00099]		
Parent Container	SwCluCBManifProvideHandle		
Description	Indicates that this handle is optional and may not always exist for each Provide Resource Entry or Require Resource Entry.		
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			

Name	SwCluCBManifNativeHandleType [ECUC_SwCluC_00033]		
Parent Container	SwCluCBManifProvideHandle		
Description	<p>This paramter determines the underlying native type of the handle which can be</p> <ul style="list-style-type: none"> • pointer to variable • value • pointer to function <p>This information is required to correctly initialize the handle in the Immutable Interface or Modifiable Interface.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DATA_REFERENCE	Handle is pointer to variable	
	FUNCTION_REFERENCE	Handle is pointer to function	
	VALUE	Handle is a value	
Default Value	DATA_REFERENCE		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	SwCluCBManifHandleImpleTypeRef [ECUC_SwCluC_00015]		
Parent Container	SwCluCBManifProvideHandle		
Description	Reference to the ImplementationDataType of the handle. This type can be additionally configured to get a correct casting of the native handle type to the returned handel type. Tags: atp.Status=draft		
Multiplicity	0..1		
Type	Foreign reference to IMPLEMENTATION-DATA-TYPE		
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

SWS Item	[ECUC_SwCluC_00014]		
Container Name	SwCluCBManifNotifierHandle		
Parent Container	SwCluCBManifResourceType		
Description	Defines a Notifier Handle and its properties for this resource type. Tags: atp.Status=draft Attributes: requiresIndex=true		
Configuration Parameters			

Name	SwCluCBManifIsOptional [ECUC_SwCluC_00099]		
Parent Container	SwCluCBManifNotifierHandle		
Description	Indicates that this handle is optional and may not always exist for each Provide Resource Entry or Require Resource Entry.		
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			

Name	SwCluCBManifNativeHandleType [ECUC_SwCluC_00033]		
Parent Container	SwCluCBManifNotifierHandle		
Description	<p>This paramter determines the underlying native type of the handle which can be</p> <ul style="list-style-type: none"> • pointer to variable • value • pointer to function <p>This information is required to correctly initialize the handle in the Immutable Interface or Modifiable Interface.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DATA_REFERENCE	Handle is pointer to variable	
	FUNCTION_REFERENCE	Handle is pointer to function	
	VALUE	Handle is a value	
Default Value	DATA_REFERENCE		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	SwCluCBManifHandleImpleTypeRef [ECUC_SwCluC_00015]		
Parent Container	SwCluCBManifNotifierHandle		
Description	<p>Reference to the ImplementationDataType of the handle. This type can be additionally configured to get a correct casting of the native handle type to the returned handel type.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	Foreign reference to IMPLEMENTATION-DATA-TYPE		
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

10.2.6 Cross Cluster Communication

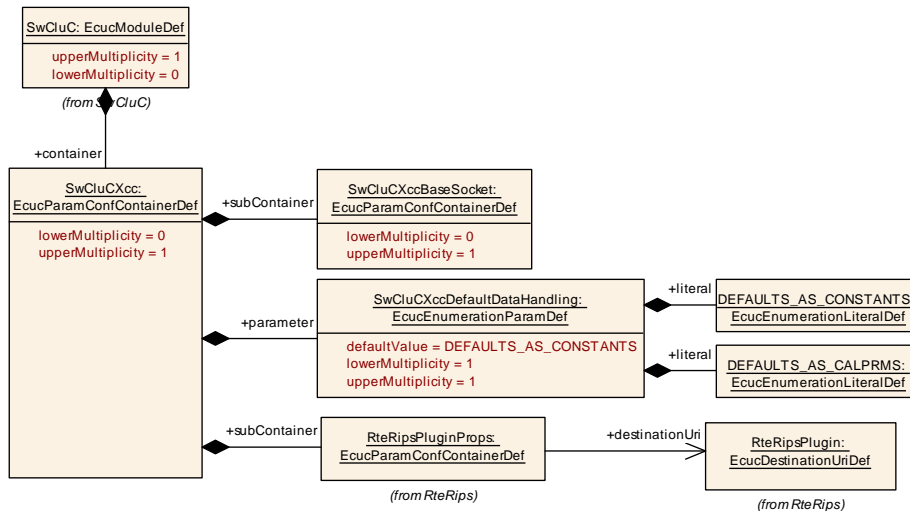


Figure 10.9: General Cross Cluster Communication Parameters

SWS Item	[ECUC_SwCluC_00075]
Container Name	SwCluCXcc
Parent Container	SwCluC
Description	Configuration of the Binary Manifest of the Software Cluster Connection Tags: atp.Status=draft
Configuration Parameters	

Name	SwCluCXccDefaultDataHandling [ECUC_SwCluC_00084]	
Parent Container	SwCluCXcc	
Description	SwCluCXccDefaultDataHandling defines whether the default data for unconnected RPortPrototypes of a Software Cluster are instantiated as fixed constants or as calibration parameter. Tags: atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEFAULTS_AS_CALPRMS	Default data are implemented as calibration parameter. Tags: atp.Status=draft

Default Value	DEFAULTS_AS_CONSTANTS	Default data are implemented as fixed constants. Tags: atp.Status=draft	
	DEFAULTS_AS_CONSTANTS		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
RteRipsPluginProps	1	This container defines the identity of the Rte Implementation Plug-in and provides the RTE relevant parameters of the Rte Implementation Plug-in. The shortName of the container defines the name of the Rte Implementation Plug-in used for the API infixes.
SwCluCXccBaseSocket	0..1	Maps a XccBaseSocket to the CpSoftwareClusterServiceResource describing the XccBaseSocket resource in the clustered system. Maps a XccBaseSocket to a specific EcucPartition on which the XccBaseSocket is working. Tags: atp.Status=draft

10.2.6.1 Cross Cluster Communication Base Socket

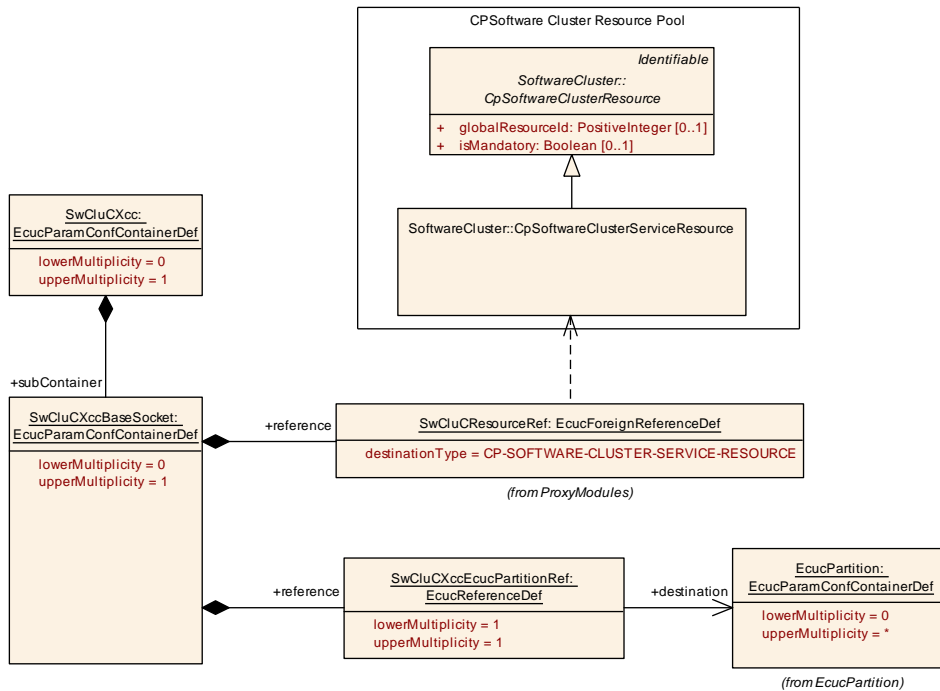


Figure 10.10: Cross Cluster Communication Base Socket

SWS Item	[ECUC_SwCluC_00076]
Container Name	SwCluCXccBaseSocket
Parent Container	SwCluCXcc
Description	Maps a XccBaseSocket to the CpSoftwareClusterServiceResource describing the XccBaseSocket resource in the clustered system. Maps a XccBaseSocket to a specific EcucPartition on which the XccBaseSocket is working. Tags: atp.Status=draft
Configuration Parameters	

Name	SwCluCResourceRef [ECUC_SwCluC_00096]
Parent Container	SwCluCXccBaseSocket
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOUR CE
Scope / Dependency	scope: ECU

Name	SwCluCXccEcucPartitionRef [ECUC_SwCluC_00077]
Parent Container	SwCluCXccBaseSocket

Description	Reference to the EcucPartition on which the XccBaseSocket is available.		
	Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

[SWS_SwCluC_CONSTR_03063]{DRAFT} [SwCluCXccBaseSocket](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_XCC_BASE_SOCKET](#) [The [SwCluCXccBaseSocket.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_XCC_BASE_SOCKET](#).]()

10.2.6.2 Cross Cluster Communication Rte Plug-in Props

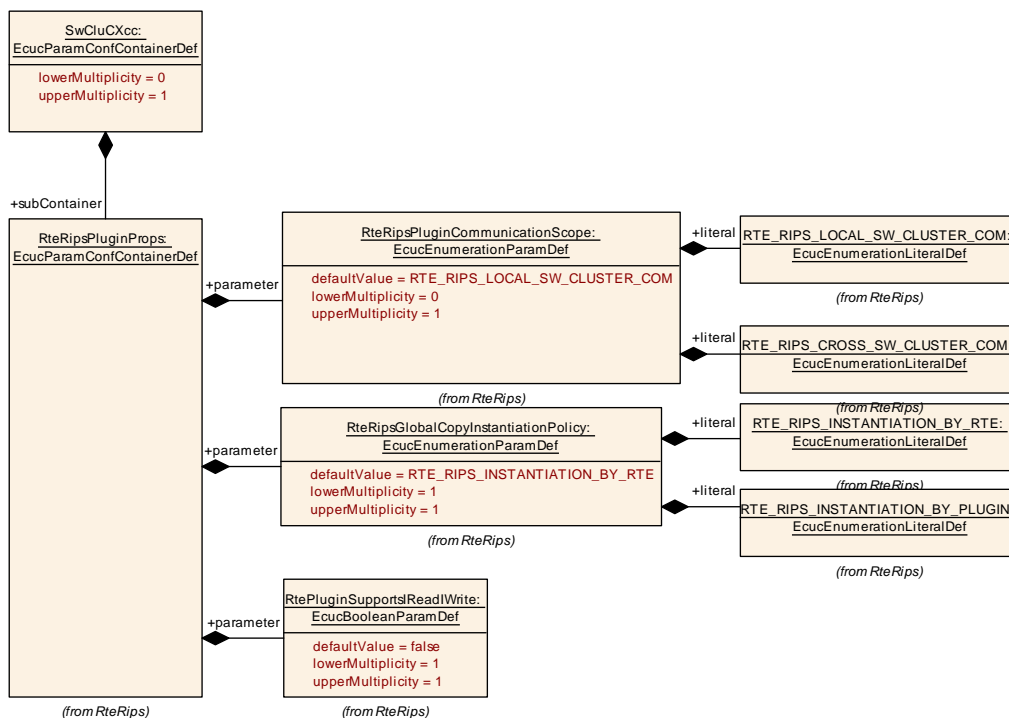


Figure 10.11: Cross Cluster Communication RTE Plug-In Properties

SWS Item	[ECUC_Rte_09173]
Container Name	RteRipsPluginProps
Parent Container	SwCluCXcc
Destination Uri Definition	RteRipsPlugin
Description	This container defines the identity of the Rte Implementation Plug-in and provides the RTE relevant parameters of the Rte Implementation Plug-in. The shortName of the container defines the name of the Rte Implementation Plug-in used for the API infixes.
Configuration Parameters	

Name	RtePluginSupportsIReadIWrite [ECUC_Rte_09169]
Parent Container	RteRipsPluginProps
Description	Denotes if or if not the plug-in supports the Rte_Rips_IRead/IWrite macros for primitive data.
Multiplicity	1
Type	EcucBooleanParamDef
Default Value	false
Post-Build Variant Value	false
Scope / Dependency	scope: local

Name	RteRipsGlobalCopyInstantiationPolicy [ECUC_Rte_09170]	
Parent Container	RteRipsPluginProps	
Description	Globally enables or disables the support for Rte Implementation Plug-Ins (RIPS)	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	RTE_RIPS_INSTANTIATION_BY_PLUGIN	The Rte Implementation Plug-In shall provide the global copy(s) for each Communication Graph.
	RTE_RIPS_INSTANTIATION_BY_RTE	The RTE shall provide an individual global copy for each Communication Graph.
Default Value	RTE_RIPS_INSTANTIATION_BY_RTE	
Post-Build Variant Value	false	
Scope / Dependency	scope: local	

Name	RteRipsPluginCommunicationScope [ECUC_Rte_09171]	
Parent Container	RteRipsPluginProps	
Description	Defines the communication scope for which the Rte Implementation Plug-Ins (RIPS) serves. If this parameter is not set, the default behavior RTE_RIPS_LOCAL_SW_CLUSTER_COM applies.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	RTE_RIPS_CROSS_SW_CLUSTER_COM	The Rte Implementation Plug-In handles the Cross Software Cluster Communication.

Default Value	RTE_RIPS_LOCAL_SW_CLUSTER_COM	The Rte Implementation Plug-In handles the Local Software Cluster Communication.
	RTE_RIPS_LOCAL_SW_CLUSTER_COM	
Post-Build Variant Value	false	
Scope / Dependency	scope: local	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
RteRipsPlugin CommunicationType Support	0..1	This container defines the different elements of communication graphs that the plugin is responsible for. Tags: atp.Status=draft

10.2.7 Proxy Modules

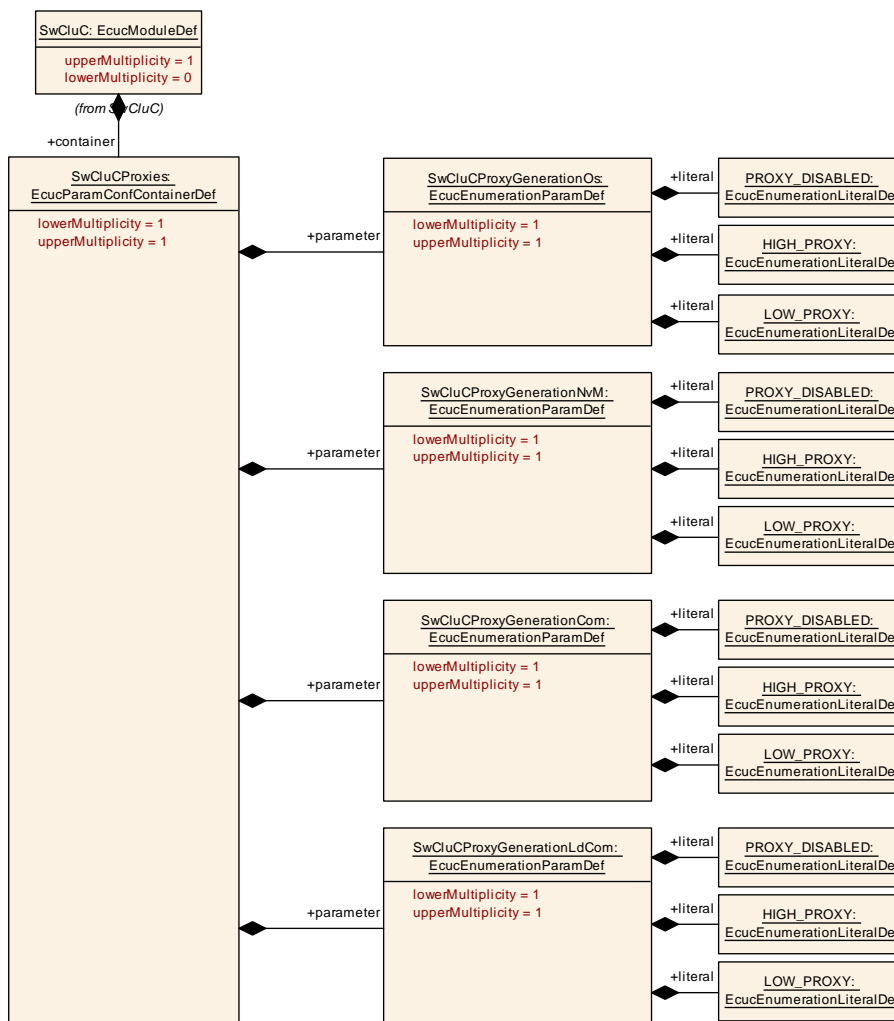


Figure 10.12: General Proxies Parameters

SWS Item	[ECUC_SwCluC_00036]		
Container Name	SwCluCProxies		
Parent Container	SwCluC		
Description	General configuration of the Proxy Modules of Software Cluster Connection Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCProxyGenerationCom [ECUC_SwCluC_00100]		
Parent Container	SwCluCProxies		
Description	Defines whether the Com Proxy code and models are generated. Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	HIGH_PROXY	Enables the according High Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft	
	LOW_PROXY	Enables the according Low Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft	
	PROXY_DISABLED	Disables the Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft	
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyGenerationDcm [ECUC_SwCluC_00102]	
Parent Container	SwCluCProxies	
Description	Defines whether the Dcm Proxy code and models are generated. Tags: atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	HIGH_PROXY	Enables the according High Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft
	LOW_PROXY	Enables the according Low Proxy Module code and AUTOSAR model generation.
	PROXY_DISABLED	Disables the Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft
	Pre-compile time	X All Variants
	Link time Post-build time	–
Multiplicity Configuration Class	Pre-compile time	X All Variants
	Link time Post-build time	–
	Value Configuration Class	–
Value Configuration Class	Pre-compile time	X All Variants
	Link time Post-build time	–
	Scope / Dependency	scope: ECU

Name	SwCluCProxyGenerationDem [ECUC_SwCluC_00103]	
Parent Container	SwCluCProxies	
Description	Defines whether the Dem Proxy code and models are generated. Tags: atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	HIGH_PROXY	Enables the according High Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft
	LOW_PROXY	Enables the according Low Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft

Multiplicity Configuration Class	PROXY_DISABLED	Disables the Proxy Module code and AUTOSAR model generation.	
	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyGenerationFiM [ECUC_SwCluC_00104]		
Parent Container	SwCluCProxies		
Description	Defines whether the FiM Proxy code and models are generated.		
	Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	HIGH_PROXY	Enables the according High Proxy Module code and AUTOSAR model generation.	
		Tags: atp.Status=draft	
	LOW_PROXY	Enables the according Low Proxy Module code and AUTOSAR model generation.	
	Tags: atp.Status=draft		
Multiplicity Configuration Class	PROXY_DISABLED	Disables the Proxy Module code and AUTOSAR model generation.	
		Tags: atp.Status=draft	
	Pre-compile time	X	All Variants
	Link time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyGenerationLdCom [ECUC_SwCluC_00101]	
Parent Container	SwCluCProxies	
Description	Defines whether the LdCom Proxy code and models are generated. Tags: atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	HIGH_PROXY	Enables the according High Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft
	LOW_PROXY	Enables the according Low Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft
	PROXY_DISABLED	Disables the Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft
	Pre-compile time	X All Variants
	Link time	–
	Post-build time	–
Multiplicity Configuration Class	Pre-compile time	X All Variants
	Link time	–
	Post-build time	–
Value Configuration Class	Pre-compile time	X All Variants
	Link time	–
	Post-build time	–
Scope / Dependency	scope: ECU	

Name	SwCluCProxyGenerationNvM [ECUC_SwCluC_00038]	
Parent Container	SwCluCProxies	
Description	Defines whether the NvM Proxy code and models are generated. Tags: atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	HIGH_PROXY	Enables the according High Proxy Module code and AUTOSAR model generation.
	LOW_PROXY	Enables the according Low Proxy Module code and AUTOSAR model generation.
	PROXY_DISABLED	Disables the Proxy Module code and AUTOSAR model generation.

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyGenerationOs [ECUC_SwCluC_00037]		
Parent Container	SwCluCProxies		
Description	Defines whether the Os Proxy code and models are generated. Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	HIGH_PROXY	Enables the according High Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft	
	LOW_PROXY	Enables the according Low Proxy Module code and AUTOSAR model generation. Tags: atp.Status=draft	
	PROXY_DISABLED	Disables the Proxy Module code and AUTOSAR model generation Tags: atp.Status=draft	
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCComProxy	0..1	LdCom Proxy specific parameters. Tags: atp.Status=draft

SwCluCdcmProxy	0..1	Dcm Proxy specific parameters. Tags: atp.Status=draft
SwCluCDemProxy	0..1	DemProxy specific parameters. Tags: atp.Status=draft
SwCluCFiMProxy	0..1	FiM Proxy specific parameters. Tags: atp.Status=draft
SwCluCLdComProxy	0..1	LdCom Proxy specific parameters. Tags: atp.Status=draft
SwCluCNvMProxy	1	NvM Proxy specific parameters. Tags: atp.Status=draft
SwCluCOsProxy	1	Os Proxy specific parameters. Tags: atp.Status=draft

10.2.7.1 NvM Proxy

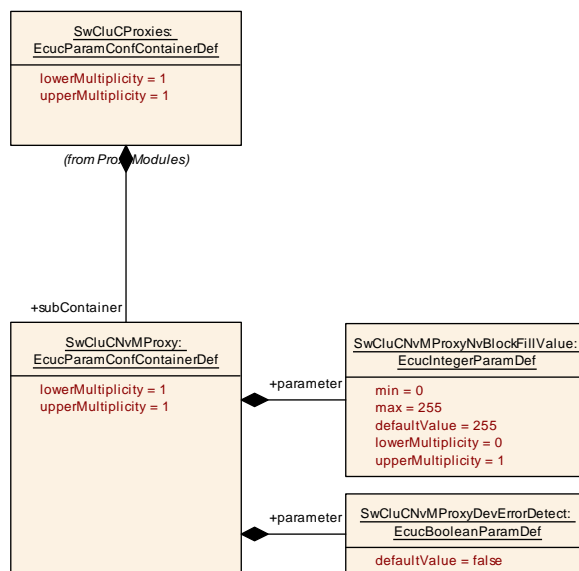


Figure 10.13: NvM Proxy Parameter

SWS Item	[ECUC_SwCluC_00039]
Container Name	SwCluCNvMProxy
Parent Container	SwCluCProxies

Description	NvM Proxy specific parameters.		
	Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCNvMProxyDevErrorDetect [ECUC_SwCluC_00083]		
Parent Container	SwCluCNvMProxy		
Description	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> • true: detection and notification is enabled. • false: detection and notification is disabled. 		
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		

Name	SwCluCNvMProxyNvBlockFillValue [ECUC_SwCluC_00055]		
Parent Container	SwCluCNvMProxy		
Description	Defines the default value used to fill unused bytes in NvBlock space in case NvBlock length in Applicative Software Cluster < NvBlock length configured in NvM of Host Software Cluster. Dependent on NvM Proxy implementation this parameter is only relevant for NvM Low Proxy or NvM High Proxy Tags: atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value	255		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCNvMBaseSocket	1..*	<p>This container configures how many EcucPartitions specific API links are required for the NvM Proxy and to which cluster resource the API set belongs.</p> <p>In the NvM Low Proxy all offered API sets needs to be configured. In the NvM High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>Tags: atp.Status=draft</p>
SwCluCNvMProxyNvBlock	1..*	<p>Maps a NvMBlockDescriptor to the CpSoftwareClusterServiceResource describing the NvBlock resource in the clustered system.</p> <p>Tags: atp.Status=draft</p>

10.2.7.1.1 NvM Base Socket

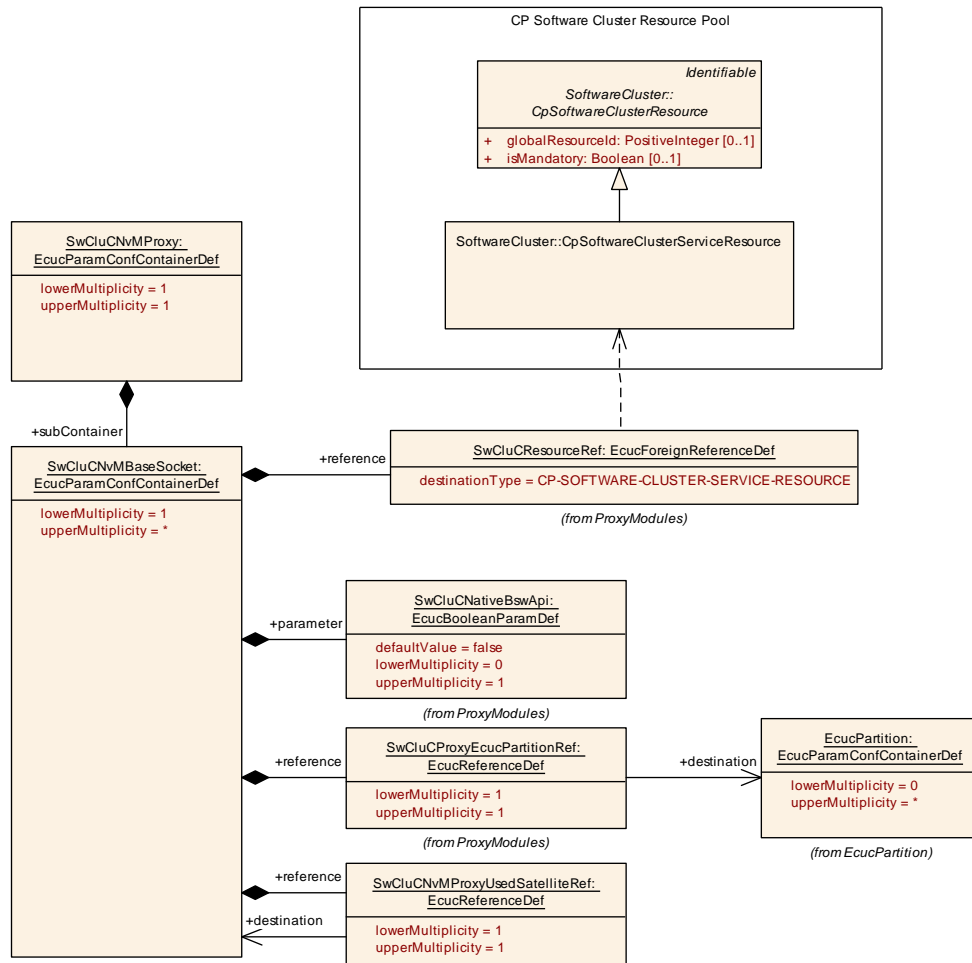


Figure 10.14: NvM Proxy Base Socket

SWS Item	[ECUC_SwCluC_00056]
Container Name	SwCluCNvMBaseSocket
Parent Container	SwCluCNvMProxy
Description	<p>This container configures how many EcucPartitions specific API links are required for the NvM Proxy and to which cluster resource the API set belongs.</p> <p>In the NvM Low Proxy all offered API sets needs to be configured. In the NvM High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>Tags: atp.Status=draft</p>

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCNativeBswApi [ECUC_SwCluC_00058]		
Parent Container	SwCluCNvMBaseSocket		
Description	<p>Defines if the native C-API without pre or suffixes is offered in the Applicative Software Cluster on this EcucPartition.</p> <p>This Parameter is only relevant for the NvM High Proxy.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCNvMProxyUsedSatelliteRef [ECUC_SwCluC_00082]		
Parent Container	SwCluCNvMBaseSocket		
Description	<p>Reference to the SwCluCNvMBaseSocket which has access to a satellite of the NvM. The owning SwCluCNvMBaseSocket uses the NvM satellite of the referenced SwCluCNvMBaseSocket.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	Reference to SwCluCNvMBaseSocket		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCNvMBaseSocket		
Description	Reference to the EcucPartition. Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00092]		
Parent Container	SwCluCNvMBaseSocket		
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft		
Multiplicity	1		
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE		
Scope / Dependency	scope: ECU		

No Included Containers

[SWS_SwCluC_CONSTR_02130]{DRAFT} [SwCluCNvMBaseSocket](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_NVM_BASE_SOCKET](#) [The [SwCluCNvMBaseSocket.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_NVM_BASE_SOCKET](#).]()

10.2.7.1.2 NvM Proxy NvBlock configuration

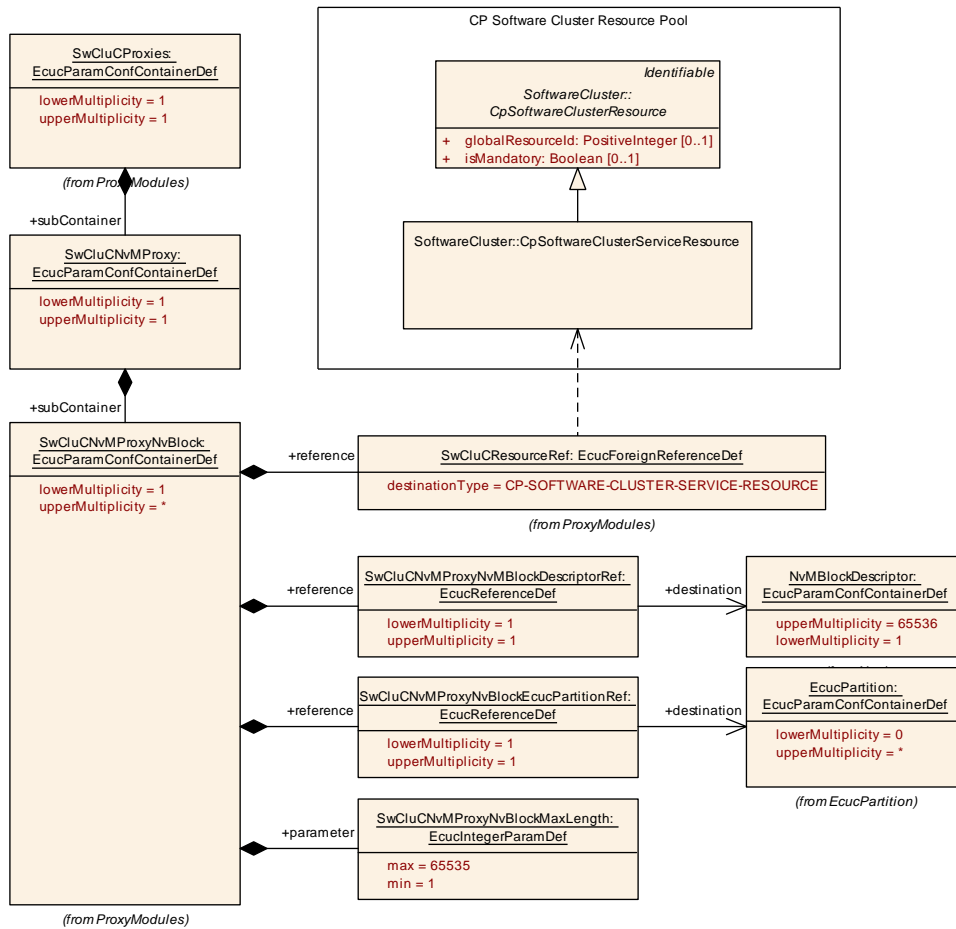


Figure 10.15: NvM Proxy NvBlock

SWS Item	[ECUC_SwCluC_00040]		
Container Name	SwCluCNvMProxyNvBlock		
Parent Container	SwCluCNvMProxy		
Description	Maps a NvMBlockDescriptor to the CpSoftwareClusterServiceResource describing the NvBlock resource in the clustered system. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Configuration Parameters			

Name	SwCluCNvMProxyNvBlockMaxLength [ECUC_SwCluC_00043]		
Parent Container	SwCluCNvMProxyNvBlock		
Description	<p>Defines the maximum NV block data length in bytes.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default Value			
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCNvMProxyNvBlockEcucPartitionRef [ECUC_SwCluC_00060]		
Parent Container	SwCluCNvMProxyNvBlock		
Description	<p>Reference to the EcucPartition on which the NvM Proxy offers the NvBlock access (call of APIfunctions, callback functions and Ports on the Service Software Component)</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCNvMProxyNvMBlockDescriptorRef [ECUC_SwCluC_00042]		
Parent Container	SwCluCNvMProxyNvBlock		
Description	<p>Reference to the NvMBlockDescriptor</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	Reference to NvMBlockDescriptor		

Scope / Dependency	scope: ECU
---------------------------	------------

Name	SwCluCResourceRef [ECUC_SwCluC_00091]
Parent Container	SwCluCNvMProxyNvBlock
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_02131]{DRAFT} [SwCluCNvMProxyNvBlock](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_NV_BLOCK](#) [The [SwCluCNvMProxyNvBlock.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_NV_BLOCK](#).]()

10.2.7.2 Os Proxy

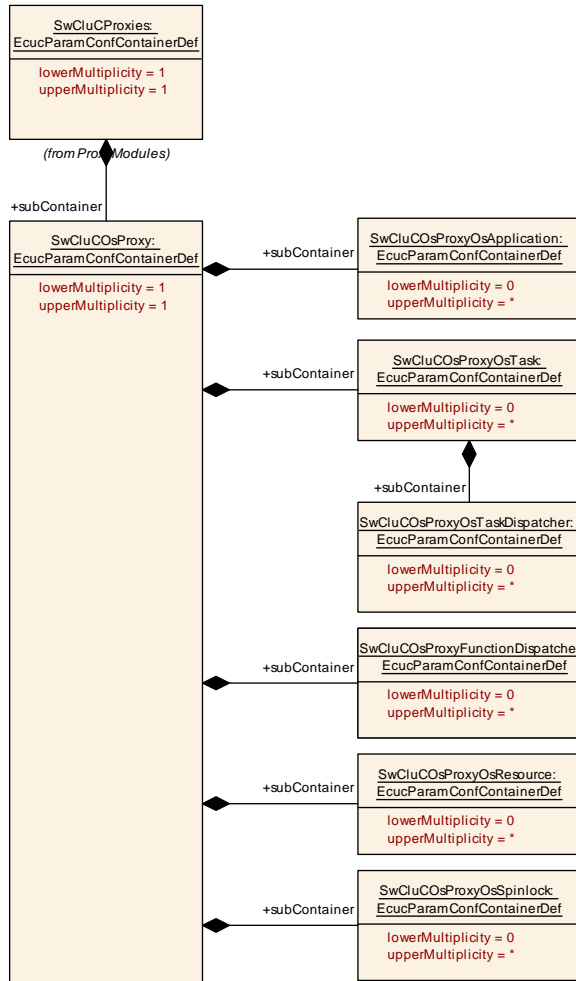


Figure 10.16: Os Proxy Parameter

SWS Item	[ECUC_SwCluC_00085]		
Container Name	SwCluCOsProxy		
Parent Container	SwCluCProxies		
Description	Os Proxy specific parameters. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCOsProxyFunctionDispatcher	0..*	<p>This container configures a function dispatcher. In the Os High Proxy this configuration puts a function as a Dispatch Entry Point in the Binary Manifest which can be called by the Os Low Proxy. In the Os Low Proxy this configuration provides the Dispatcher Function (and according resource Entry in the Binary Manifest) which is able to call the MaxNumberOfCallee Dispatch Entry Points of Os High Proxies.</p> <p>Tags: atp.Status=draft</p>
SwCluCOsProxyOsApplication	0..*	<p>Maps a OsApplication to the CpSoftwareClusterServiceResource describing the OsApplication resource in the clustered system.</p> <p>Tags: atp.Status=draft</p>
SwCluCOsProxyOsBaseSocket	0..*	<p>Maps a OsBaseSocket to the CpSoftwareClusterServiceResource describing the OsBaseSocket resource in the clustered system.</p> <p>Tags: atp.Status=draft</p>
SwCluCOsProxyOsResource	0..*	<p>Maps a OsResource to the CpSoftwareClusterServiceResource describing the OsResource resource in the clustered system.</p> <p>Tags: atp.Status=draft</p>
SwCluCOsProxyOsSpinlock	0..*	<p>Maps a OsSpinlock to the CpSoftwareClusterServiceResource describing the OsSpinlock resource in the clustered system.</p> <p>Tags: atp.Status=draft</p>
SwCluCOsProxyOsTask	0..*	<p>Maps a OsTask to the CpSoftwareClusterServiceResource describing the OsTask resource in the clustered system.</p> <p>Tags: atp.Status=draft</p>

10.2.7.2.1 Os Proxy Base Socket configuration

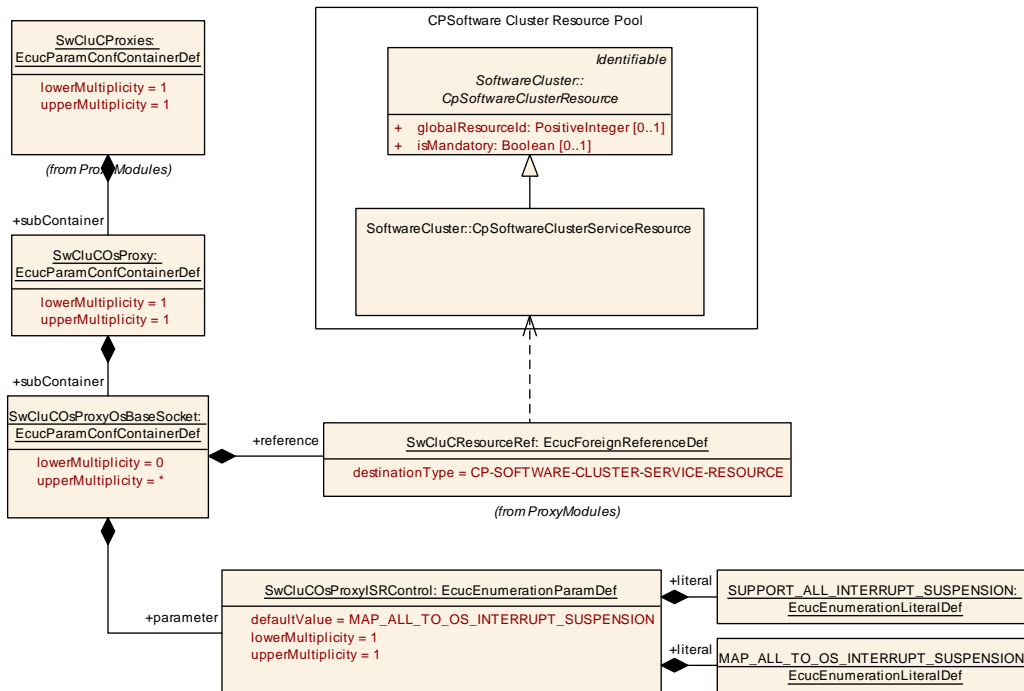


Figure 10.17: Os Proxy Base Socket Parameter

SWS Item	[ECUC_SwCluC_00053]		
Container Name	SwCluCOsProxyOsBaseSocket		
Parent Container	SwCluCOsProxy		
Description	Maps a OsBaseSocket to the CpSoftwareClusterServiceResource describing the OsBaseSocket resource in the clustered system. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Configuration Parameters			

Name	SwCluCOsProxyISRControl [ECUC_SwCluC_00054]		
Parent Container	SwCluCOsProxyOsBaseSocket		
Description	Configures whether the ISR handling in the Os Base Socket maps the Suspend / ResumeAllInterrupt to Suspend / ResumeOsInterrupt Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	MAP_ALL_TO_OS_INTERRUPT_SUSPENSION		ActivateTask is NOT passed to the Host Software Cluster. Tags: atp.Status=draft
	SUPPORT_ALL_INTERRUPT_SUSPENSION		ActivateTask is passed to the Host Software Cluster. Tags: atp.Status=draft
Default Value	MAP_ALL_TO_OS_INTERRUPT_SUSPENSION		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00090]		
Parent Container	SwCluCOsProxyOsBaseSocket		
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft		
Multiplicity	1		
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE		
Scope / Dependency	scope: ECU		

No Included Containers

[SWS_SwCluC_CONSTR_02231]{DRAFT} [SwCluCOsProxyOsBaseSocket](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_OS_BASE_SOCKET](#) [The [SwCluCOsProxyOsBaseSocket.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_OS_BASE_SOCKET](#).]()

10.2.7.2.2 Os Proxy OsApplication configuration

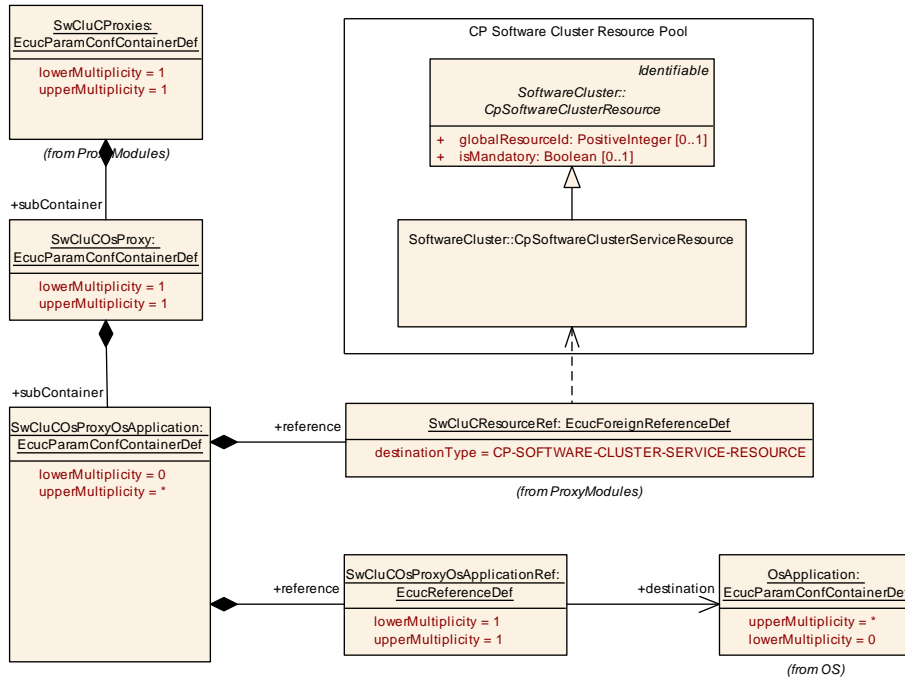


Figure 10.18: Os Proxy OsApplication Parameter

SWS Item	[ECUC_SwCluC_00046]		
Container Name	SwCluCOsProxyOsApplication		
Parent Container	SwCluCOsProxy		
Description	Maps a OsApplication to the CpSoftwareClusterServiceResource describing the OsApplication resource in the clustered system. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Configuration Parameters			

Name	SwCluCOsProxyOsApplicationRef [ECUC_SwCluC_00050]
Parent Container	SwCluCOsProxyOsApplication
Description	Reference to the OsApplication Tags: atp.Status=draft
Multiplicity	1
Type	Reference to OsApplication
Scope / Dependency	scope: ECU

Name	SwCluCResourceRef [ECUC_SwCluC_00088]
-------------	---------------------------------------

Parent Container	SwCluCOsProxyOsApplication
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_02232]{DRAFT} [SwCluCOsProxyOsApplication](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_OS_APPLICATION](#) [The [SwCluCOsProxyOsApplication.SwCluCreateResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_OS_APPLICATION](#).]()

10.2.7.2.3 Os Proxy OsTask configuration

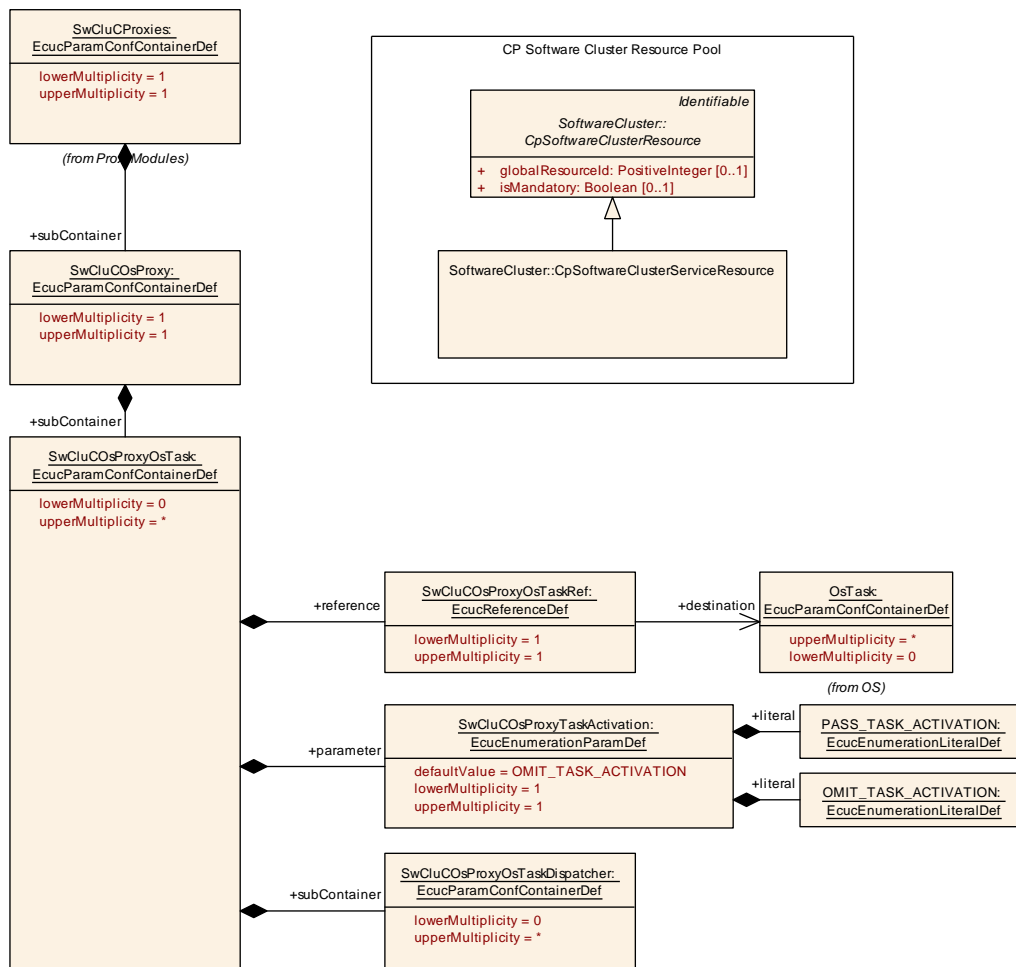


Figure 10.19: Os Proxy OsTask Parameter

SWS Item	[ECUC_SwCluC_00044]		
Container Name	SwCluCOsProxyOsTask		
Parent Container	SwCluCOsProxy		
Description	Maps a OsTask to the CpSoftwareClusterServiceResource describing the OsTask resource in the clustered system. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCOsProxyTaskActivation [ECUC_SwCluC_00048]		
Parent Container	SwCluCOsProxyOsTask		
Description	Configures for this particular OsTask whether ActivateTask is passed to the Host Software Cluster or not. Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	OMIT_TASK_ACTIVATION		ActivateTask is NOT passed to the Host Software Cluster. Tags: atp.Status=draft
	PASS_TASK_ACTIVATION		ActivateTask is passed to the Host Software Cluster. Tags: atp.Status=draft
Default Value	OMIT_TASK_ACTIVATION		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCOsProxyOsTaskRef [ECUC_SwCluC_00045]		
Parent Container	SwCluCOsProxyOsTask		
Description	Reference to the OsTask Tags: atp.Status=draft		

Multiplicity	1
Type	Reference to OsTask
Scope / Dependency	scope: ECU

Name	SwCluCResourceRef [ECUC_SwCluC_00094]
Parent Container	SwCluCOsProxyOsTask
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCOsProxyOsTask Dispatcher	0..*	<p>This container defines a task dispatcher for the owning proxy task. In the Os High Proxy this configuration provides the OS Task body function as a Dispatch Entry Point in the Binary manifest which can be called by the Os Low Proxy. In the HIGH_PROXY configuration at most one SwCluCOsProxyOsTaskDispatcher can be owned by a SwCluCOsProxyOsTask In the Os Low Proxy this configuration provides the Dispatcher Runnable (and according resource Entry in the Binary Manifest) which is able to call the MaxNumberOfCallee Dispatch Entry Points of Os High Proxies. In the LOW_PROXY configuration multiple SwCluCOsProxyOsTaskDispatcher can be owned by a SwCluCOsProxyOsTask.</p> <p>If none of the SwCluCOsProxyDispatcher<xxx>Event containers is configured it's up to the integrator to model the appropriate RTEEvent.</p> <p>Tags: atp.Status=draft</p>

[SWS_SwCluC_CONSTR_02233]{DRAFT} [SwCluCOsProxyOsTask](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_OS_TASK](#) [The [SwCluCOsProxyOsTask.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_OS_TASK](#).]()

10.2.7.2.4 Os Proxy OsResource configuration

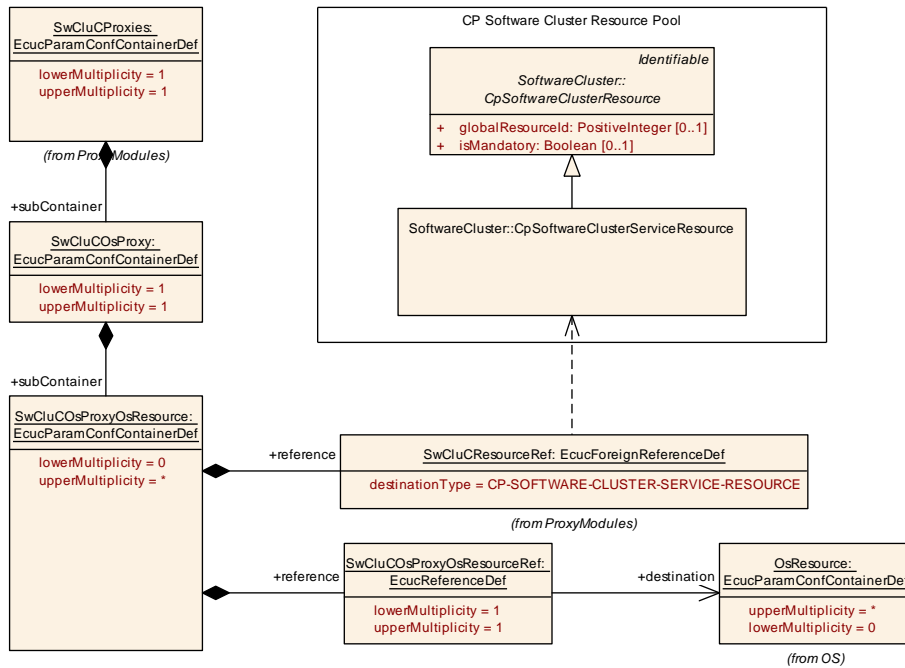


Figure 10.20: Os Proxy OsResource Parameter

SWS Item	[ECUC_SwCluC_00049]		
Container Name	SwCluCOsProxyOsResource		
Parent Container	SwCluCOsProxy		
Description	Maps a OsResource to the CpSoftwareClusterServiceResource describing the OsResource resource in the clustered system. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCOsProxyOsResourceRef [ECUC_SwCluC_00047]
Parent Container	SwCluCOsProxyOsResource
Description	Reference to the OsResource Tags: atp.Status=draft
Multiplicity	1
Type	Reference to OsResource
Scope / Dependency	scope: ECU

Name	SwCluCResourceRef [ECUC_SwCluC_00087]
Parent Container	SwCluCOsProxyOsResource

Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_02234]{DRAFT} **SwCluCOsProxyOsResource** relates only to a **CpSoftwareClusterServiceResource** of category **SWCLUSTER_RES_OS_RESOURCE** [The **SwCluCOsProxyOsResource.SwCluCResourceRef** shall only reference a **CpSoftwareClusterServiceResource** of category **SWCLUSTER_RES_OS_RESOURCE**.]()

10.2.7.2.5 Os Proxy OsSpinlock configuration

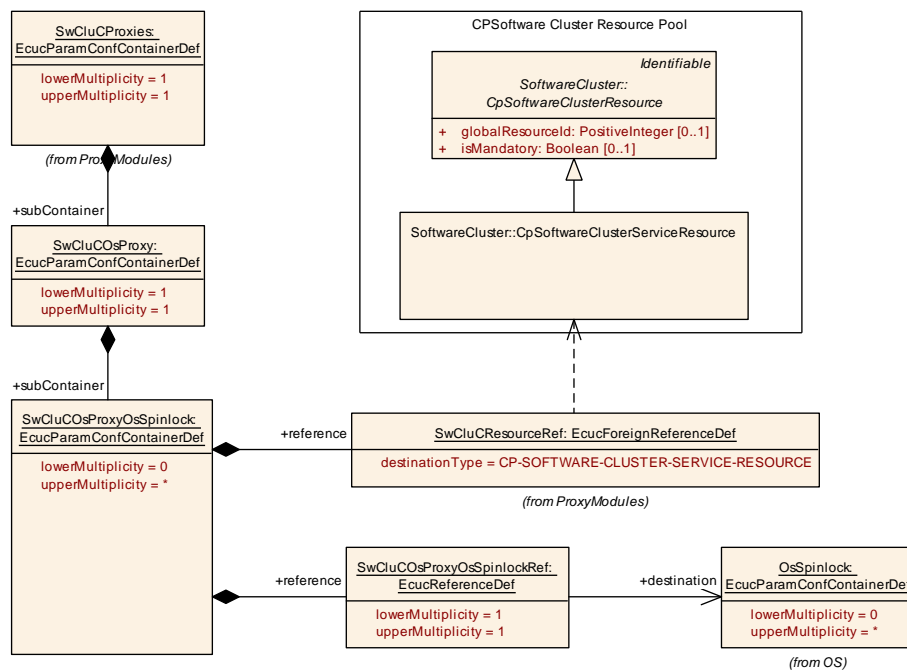


Figure 10.21: Os Proxy OsSpinlock Parameter

SWS Item	[ECUC_SwCluC_00051]
Container Name	SwCluCOsProxyOsSpinlock
Parent Container	SwCluCOsProxy
Description	Maps a OsSpinlock to the CpSoftwareClusterServiceResource describing the OsSpinlock resource in the clustered system. Tags: atp.Status=draft

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCOsProxyOsSpinlockRef [ECUC_SwCluC_00052]
Parent Container	SwCluCOsProxyOsSpinlock
Description	Reference to the OsSpinlock Tags: atp.Status=draft
Multiplicity	1
Type	Reference to OsSpinlock
Scope / Dependency	scope: ECU

Name	SwCluCResourceRef [ECUC_SwCluC_00089]
Parent Container	SwCluCOsProxyOsSpinlock
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_02235]{DRAFT} [SwCluCOsProxyOsSpinlock](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_OS_SPINLOCK](#) [The [SwCluCOsProxyOsSpinlock.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_OS_SPINLOCK](#).]()

10.2.7.2.6 Os Proxy Task Dispatcher configuration

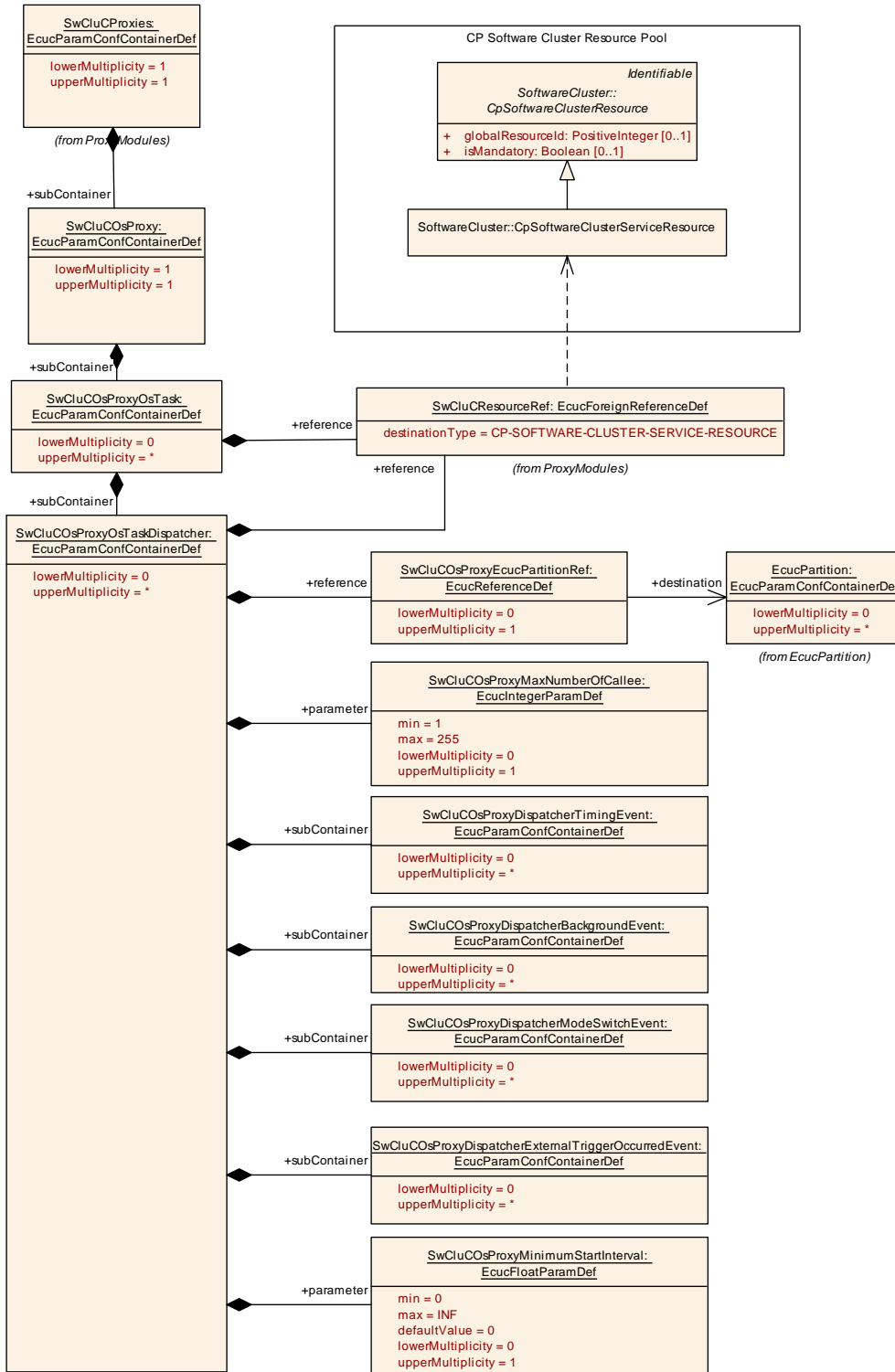


Figure 10.22: Os Proxy Task Dispatcher Parameter

SWS Item	[ECUC_SwCluC_00061]
Container Name	SwCluCOsProxyOsTaskDispatcher
Parent Container	SwCluCOsProxyOsTask

Description	<p>This container defines a task dispatcher for the owning proxy task. In the Os High Proxy this configuration provides the OS Task body function as a Dispatch Entry Point in the Binary manifest which can be called by the Os Low Proxy. In the HIGH_PROXY configuration at most one SwCluCOsProxyOsTaskDispatcher can be owned by a SwCluCOsProxyOsTask In the Os Low Proxy this configuration provides the Dispatcher Runnable (and according resource Entry in the Binary Manifest) which is able to call the MaxNumberOfCallee Dispatch Entry Points of Os High Proxies. In the LOW_PROXY configuration multiple SwCluCOsProxyOsTaskDispatcher can be owned by a SwCluCOsProxyOsTask.</p> <p>If none of the SwCluCOsProxyDispatcher<xxx>Event containers is configured it's up to the integrator to model the appropriate RTEEvent.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Configuration Parameters			

Name	SwCluCOsProxyMaxNumberOfCallee [ECUC_SwCluC_00062]		
Parent Container	SwCluCOsProxyOsTaskDispatcher		
Description	<p>Defines how many Dispatch Entry Points the dispatcher of the Os Low Proxy is able to call at most. In the Os Low Proxy this configuration parameter is mandatory.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Scope / Dependency	scope: ECU		

Name	SwCluCOsProxyMinimumStartInterval [ECUC_SwCluC_00070]		
Parent Container	SwCluCOsProxyOsTaskDispatcher		
Description	<p>Specifies the time in seconds by which two consecutive starts of an dispatcher Runnable are guaranteed to be separated.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. INF]		
Default Value	0		
Scope / Dependency	scope: ECU		

Name	SwCluCOsProxyEcucPartitionRef [ECUC_SwCluC_00078]		
Parent Container	SwCluCOsProxyOsTaskDispatcher		
Description	<p>If the reference is not provided, the task dispatcher is provided on the EcuPartition to which the OsApplication of the OsTask related to the owning SwCluCOsProxyOsTask belongs.</p> <p>If the reference is provided, the task dispatcher is provided on the referenced EcuPartition. In this case the task dispatcher is scheduled in the OsTask related to the owning SwCluCOsProxyOsTask but is changing its partition before the task dispatcher schedules proxy tasks in the Applicative Software Cluster.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00095]		
Parent Container	SwCluCOsProxyOsTaskDispatcher		
Description	Reference to the CpSoftwareClusterServiceResource. <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE		
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCOsProxyDispatcherBackgroundEvent	0..*	Configures a BackgroundEvent starting the owing Dispatcher Runnable. Tags: atp.Status=draft
SwCluCOsProxyDispatcherExternalTriggerOccurredEvent	0..*	Configures a ExternalTriggerOccurredEvent starting the owing Dispatcher Runnable. Tags: atp.Status=draft

SwCluCOsProxyDispatcherModeSwitchEvent	0..*	Configures a ModeSwitchEvent starting the owing Dispatcher Runnable. Tags: atp.Status=draft
SwCluCOsProxyDispatcherTimingEvent	0..*	Configures a TimingEvent starting the owing Dispatcher Runnable. Tags: atp.Status=draft

[SWS_SwCluC_CONSTR_02236]{DRAFT} **SwCluCOsProxyOsTaskDispatcher** relates only to a **CpSoftwareClusterServiceResource** of category **SWCLUSTER_RES_OS_TASK_DISPATCHER** [The **SwCluCOsProxyOsTaskDispatcher.SwCluCResourceRef** shall only reference a **CpSoftwareClusterServiceResource** of category **SWCLUSTER_RES_OS_TASK_DISPATCHER**.]()

10.2.7.2.6.1 Os Proxy Task Dispatcher Timing Event configuration

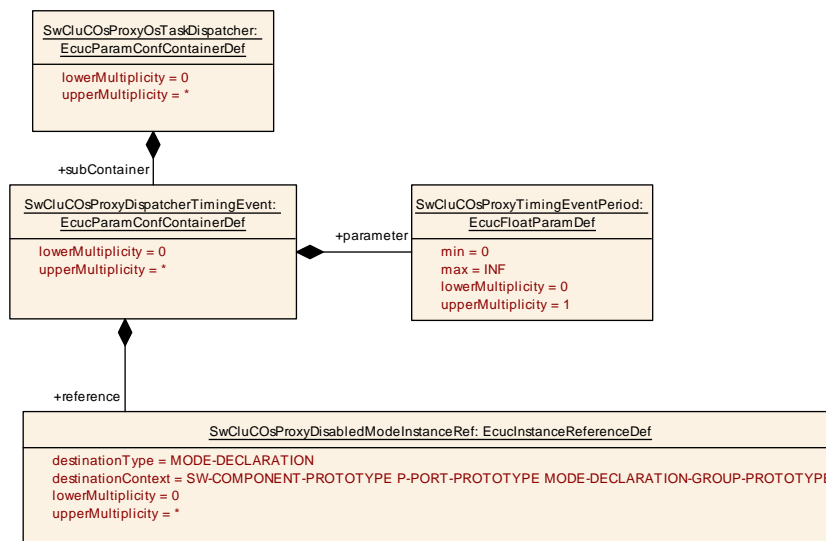


Figure 10.23: Os Proxy Task Dispatcher Timing Event Parameter

SWS Item	[ECUC_SwCluC_00066]		
Container Name	SwCluCOsProxyDispatcherTimingEvent		
Parent Container	SwCluCOsProxyOsTaskDispatcher		
Description	Configures a TimingEvent starting the owing Dispatcher Runnable. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	

Configuration Parameters

Name	SwCluCOsProxyTimingEventPeriod [ECUC_SwCluC_00065]	
Parent Container	SwCluCOsProxyDispatcherTimingEvent	
Description	Period of timing event in seconds. The value of this attribute shall be greater than zero. Tags: atp.Status=draft	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	[0 .. INF]	
Default Value		
Scope / Dependency	scope: ECU	

Name	SwCluCOsProxyDisabledModeInstanceRef [ECUC_SwCluC_00071]	
Parent Container	SwCluCOsProxyDispatcherTimingEvent	
Description	Reference to the Mode instance in a PPortPrototype that disable the Event. Tags: atp.Status=draft	
Multiplicity	0..*	
Type	Instance reference to MODE-DECLARATION context: SW-COMPONENT-PROTOTYPE P-PORT-PROTOTYPE MODE-DECLARATION-GROUP-PROTOTYPE	
Scope / Dependency	scope: ECU	

No Included Containers

10.2.7.2.6.2 Os Proxy Task Dispatcher Background Event configuration

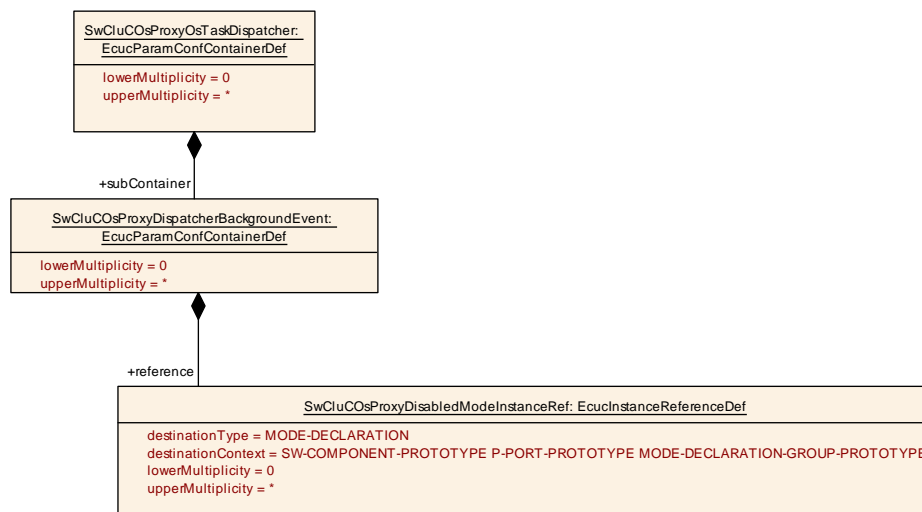


Figure 10.24: Os Proxy Task Dispatcher Background Event Parameter

SWS Item	[ECUC_SwCluC_00067]		
Container Name	SwCluCOsProxyDispatcherBackgroundEvent		
Parent Container	SwCluCOsProxyOsTaskDispatcher		
Description	Configures a BackgroundEvent starting the owing Dispatcher Runnable. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCOsProxyDisabledModeInstanceRef [ECUC_SwCluC_00071]		
Parent Container	SwCluCOsProxyDispatcherBackgroundEvent		
Description	Reference to the Mode instance in a PPortPrototype that disable the Event. Tags: atp.Status=draft		
Multiplicity	0..*		
Type	Instance reference to MODE-DECLARATION context: SW-COMPONENT-PROTOTYPE P-PORT-PROTOTYPE MODE-DECLARATION-GROUP-PROTOTYPE		
Scope / Dependency	scope: ECU		

No Included Containers

10.2.7.2.6.3 Os Proxy Task Dispatcher Mode Switch Event configuration

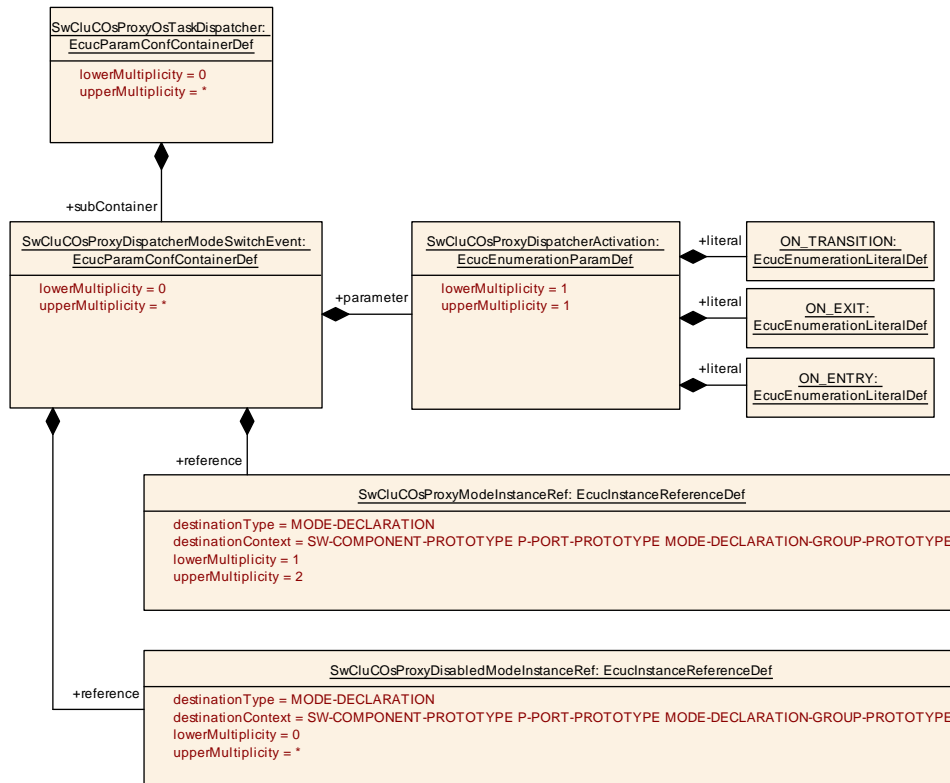


Figure 10.25: Os Proxy Task Dispatcher Mode Switch Event Parameter

SWS Item	[ECUC_SwCluC_00068]		
Container Name	SwCluCOsProxyDispatcherModeSwitchEvent		
Parent Container	SwCluCOsProxyOsTaskDispatcher		
Description	Configures a ModeSwitchEvent starting the owing Dispatcher Runnable. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCOsProxyDispatcherActivation [ECUC_SwCluC_00072]		
Parent Container	SwCluCOsProxyDispatcherModeSwitchEvent		
Description	Specifies if the event is activated on entering or exiting the referenced Mode instance. Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ON_ENTRY	On entering the referred mode.	
	ON_EXIT	On exiting the referred mode.	
	ON_TRANSITION	On transition of the 1st referred mode to the 2nd referred mode.	
	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: ECU		

Name	SwCluCOsProxyDisabledModeInstanceRef [ECUC_SwCluC_00071]		
Parent Container	SwCluCOsProxyDispatcherModeSwitchEvent		
Description	Reference to the Mode instance in a PPortPrototype that disable the Event. Tags: atp.Status=draft		
Multiplicity	0..*		
Type	Instance reference to MODE-DECLARATION context: SW-COMPONENT-PROTOTYPE P-PORT-PROTOTYPE MODE-DECLARATION-GROUP-PROTOTYPE		
Scope / Dependency	scope: ECU		

Name	SwCluCOsProxyModeInstanceRef [ECUC_SwCluC_00073]		
Parent Container	SwCluCOsProxyDispatcherModeSwitchEvent		
Description	Reference to one or two Mode instances in a PPortPrototype that that initiate the mode switch. Tags: atp.Status=draft		
Multiplicity	1..2		
Type	Instance reference to MODE-DECLARATION context: SW-COMPONENT-PROTOTYPE P-PORT-PROTOTYPE MODE-DECLARATION-GROUP-PROTOTYPE		
Scope / Dependency	scope: ECU		

No Included Containers

10.2.7.2.6.4 Os Proxy Task Dispatcher External Trigger Occurred Event configuration

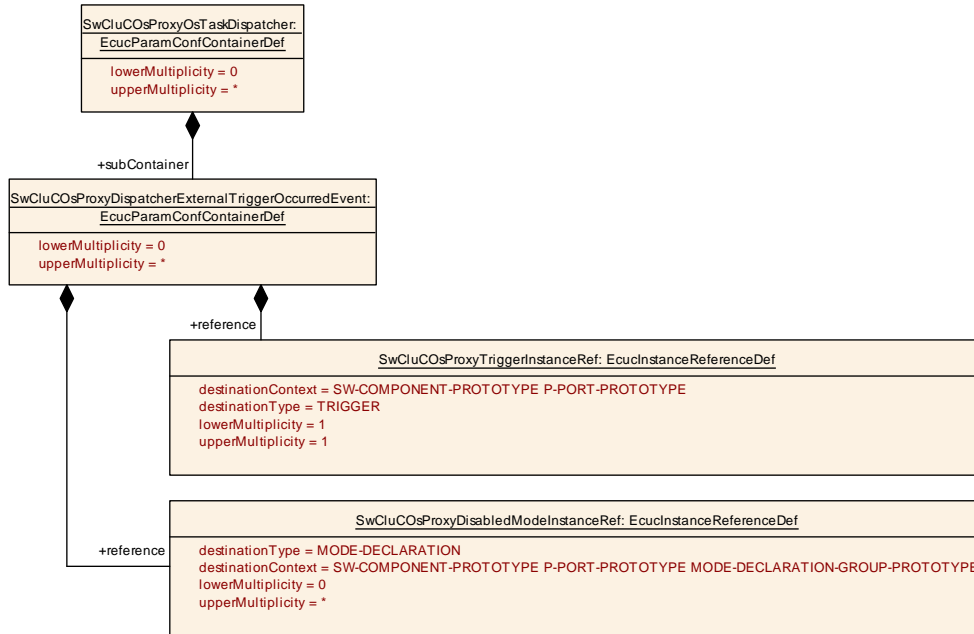


Figure 10.26: Os Proxy Task Dispatcher External Trigger Occurred Event Parameter

SWS Item	[ECUC_SwCluC_00069]		
Container Name	SwCluCOsProxyDispatcherExternalTriggerOccurredEvent		
Parent Container	SwCluCOsProxyOsTaskDispatcher		
Description	Configures a ExternalTriggerOccurredEvent starting the owing Dispatcher Runnable. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Configuration Parameters			

Name	SwCluCOsProxyDisabledModelInstanceRef [ECUC_SwCluC_00071]
Parent Container	SwCluCOsProxyDispatcherExternalTriggerOccurredEvent
Description	Reference to the Mode instance in a PPortPrototype that disable the Event. Tags: atp.Status=draft
Multiplicity	0..*
Type	Instance reference to MODE-DECLARATION context: SW-COMPONENT-PROTOTYPE P-PORT-PROTOTYPE MODE-DECLARATION-GROUP-PROTOTYPE
Scope / Dependency	scope: ECU

Name	SwCluCOsProxyTriggerInstanceRef [ECUC_SwCluC_00074]
Parent Container	SwCluCOsProxyDispatcherExternalTriggerOccurredEvent
Description	Reference to the Trigger instances in a PPortPrototype that raise the trigger. Tags: atp.Status=draft
Multiplicity	1
Type	Instance reference to TRIGGER context: SW-COMPONENT-PROTOTYPE P-PORT-PROTOTYPE
Scope / Dependency	scope: ECU

No Included Containers

10.2.7.2.7 Os Proxy Function Dispatcher configuration

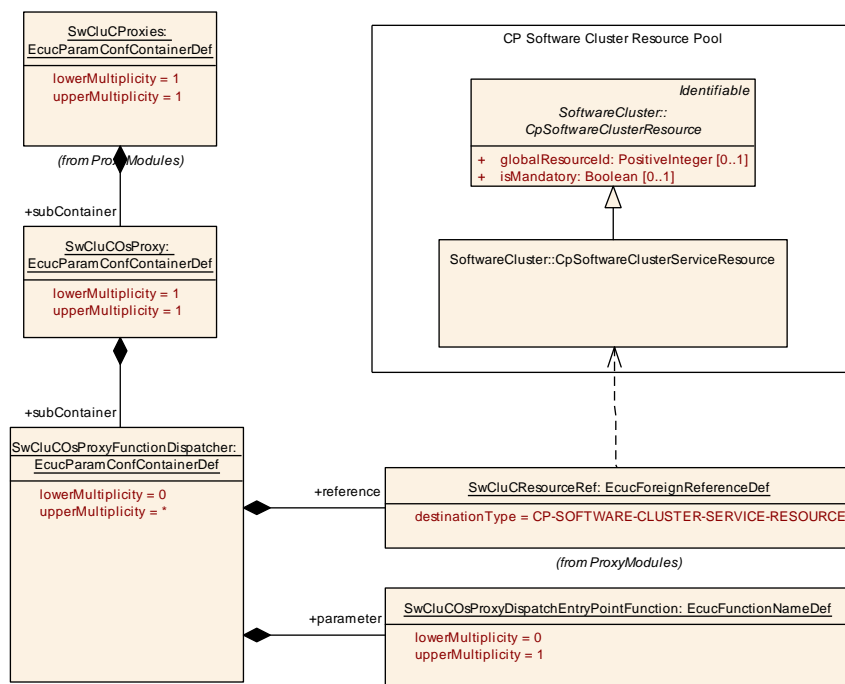


Figure 10.27: Os Proxy Function Dispatcher Parameter

SWS Item	[ECUC_SwCluC_00063]
Container Name	SwCluCOsProxyFunctionDispatcher
Parent Container	SwCluCOsProxy

Description	<p>This container configures a function dispatcher. In the Os High Proxy this configuration puts a function as a Dispatch Entry Point in the Binary Manifest which can be called by the Os Low Proxy. In the Os Low Proxy this configuration provides the Dispatcher Function (and according resource Entry in the Binary Manifest) which is able to call the MaxNumberOfCallee Dispatch Entry Points of Os High Proxies.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCOsProxyDispatchEntryPointFunction [ECUC_SwCluC_00064]		
Parent Container	SwCluCOsProxyFunctionDispatcher		
Description	<p>Name of the function which is put into the Binary Manifest as Dispatch Entry Point. In the Os High Proxy this configuration parameter is mandatory</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00097]		
Parent Container	SwCluCOsProxyFunctionDispatcher		
Description	<p>Reference to the CpSoftwareClusterServiceResource.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE		
Scope / Dependency	scope: ECU		

No Included Containers

[SWS_SwCluC_CONSTR_02237]{DRAFT} **SwCluCOsProxyFunctionDispatcher** relates only to a **CpSoftwareClusterServiceResource** of category **SWCLUSTER_RES_OS_FNC_DISPATCHER** [The **SwCluCOsProxyFunctionDispatcher.SwCluCResourceRef** shall only reference a **CpSoftwareClusterServiceResource** of category **SWCLUSTER_RES_OS_FNC_DISPATCHER**.]()

10.2.7.3 Com Proxy

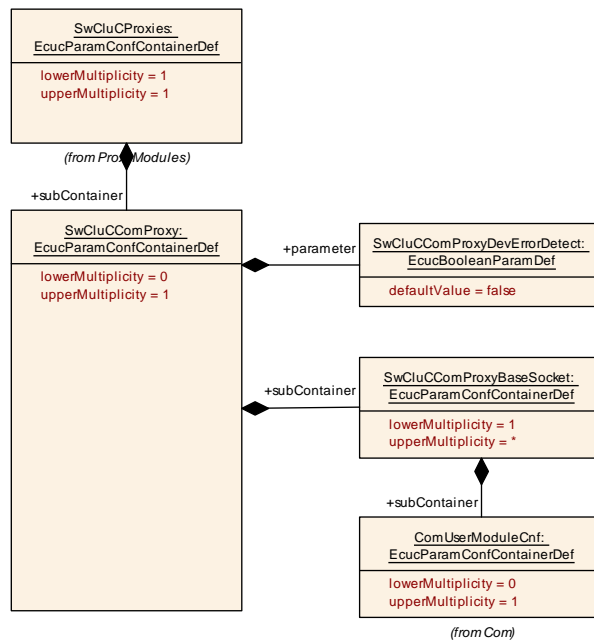


Figure 10.28: Com Proxy Parameter

SWS Item	[ECUC_SwCluC_00127]		
Container Name	SwCluCComProxy		
Parent Container	SwCluCProxies		
Description	LdCom Proxy specific parameters. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCComProxyDevErrorDetect [ECUC_SwCluC_00128]		
Parent Container	SwCluCComProxy		
Description	<p>Switches the development error detection and notification on or off.</p> <ul style="list-style-type: none"> • true: detection and notification is enabled. • false: detection and notification is disabled. <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCComProxyBase Socket	1..*	<p>This container configures how many EcucPartitions specific API links are required for the Com Proxy and to which cluster resource the API set belongs.</p> <p>In the Com Low Proxy all offered API sets (incl callbacks) needs to be configured. In the Com High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>The ComUserModuleCnf is only applicable in the Host Software Cluster and shall not be configured in the scope of a Applicative Software Cluster. One ComUserModuleCnf instance can either server for ComSignals and ComSignalGroups on multiple EcucPartitions or exactly one EcucPartition. How many ComUser instances an Rte in an related Applicative Software Cluster requires is vendor specific.</p> <p>Tags: atp.Status=draft</p>

10.2.7.3.1 Com Proxy Base Socket

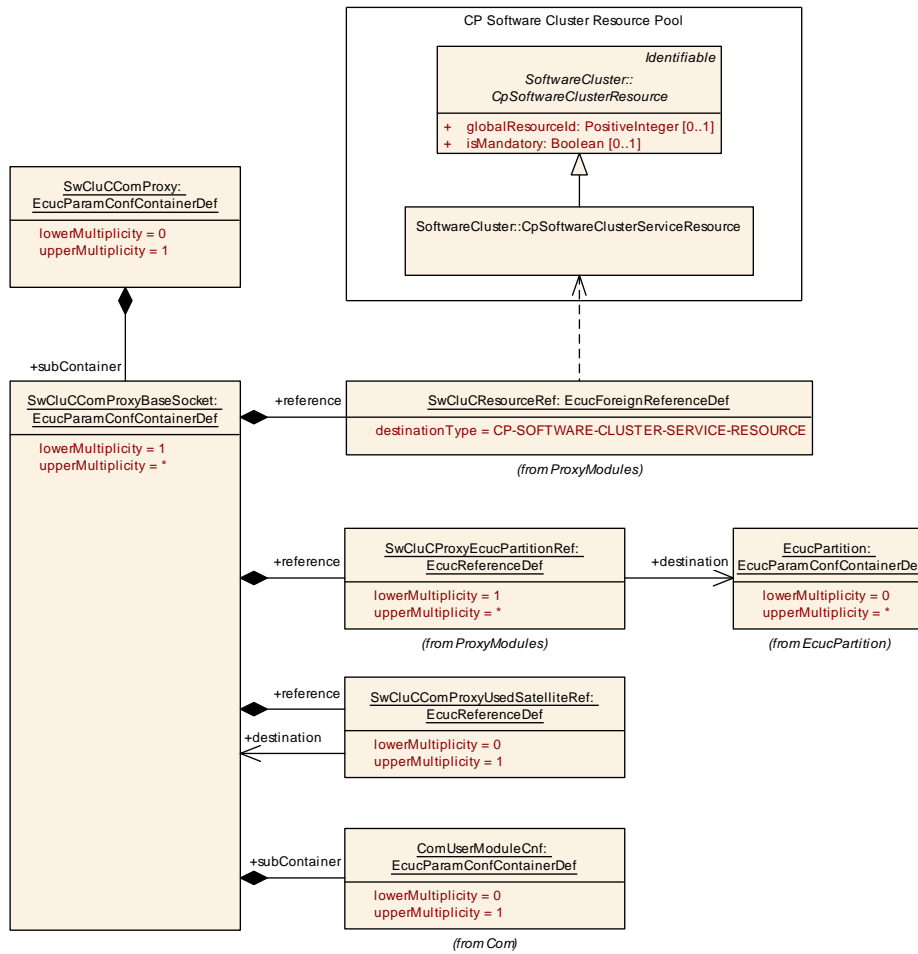


Figure 10.29: Com Proxy Base Socket

SWS Item	[ECUC_SwCluC_00131]
Container Name	SwCluCComProxyBaseSocket
Parent Container	SwCluCComProxy

Description	<p>This container configures how many EcucPartitions specific API links are required for the Com Proxy and to which cluster resource the API set belongs.</p> <p>In the Com Low Proxy all offered API sets (incl callbacks) needs to be configured. In the Com High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>The ComUserModuleCnf is only applicable in the Host Software Cluster and shall not be configured in the scope of a Applicative Software Cluster. One ComUserModuleCnf instance can either server for ComSignals and ComSignalGroups on multiple EcucPartitions or exactly one EcucPartition. How many ComUser instances an Rte in an related Applicative Software Cluster requires is vendor specific.</p> <p>Tags: atp.Status=draft</p>			
	Multiplicity Configuration Class	Pre-compile time	X	All Variants
		Link time	–	
		Post-build time	–	
Configuration Parameters				

Name	SwCluCComProxyUsedSatelliteRef [ECUC_SwCluC_00132]		
Parent Container	SwCluCComProxyBaseSocket		
Description	<p>Reference to the SwCluCComBaseSocket which has access to a satellite of the Com. The owning SwCluCComBaseSocket uses the Com satellite of the referenced SwCluCComBaseSocket.</p> <p>Tags: atp.Status=draft</p>		
	Multiplicity	0..1	
Type	Reference to SwCluCComProxyBaseSocket		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00147]		
Parent Container	SwCluCComProxyBaseSocket		

Description	Reference to the EcucPartition. Tags: atp.Status=draft		
Multiplicity	1..*		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef
Parent Container	SwCluCComProxyBaseSocket
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComUserModuleCnf	0..1	Contains the configuration parameters of the Com user module. Tags: atp.Status=draft

[SWS_SwCluC_CONSTR_02664]{DRAFT} [SwCluCComProxyBaseSocket](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_COM_BASE_SOCKET](#) [The [SwCluCComProxyBaseSocket.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_COM_BASE_SOCKET](#).]()

10.2.7.3.2 Com User Module Configuration

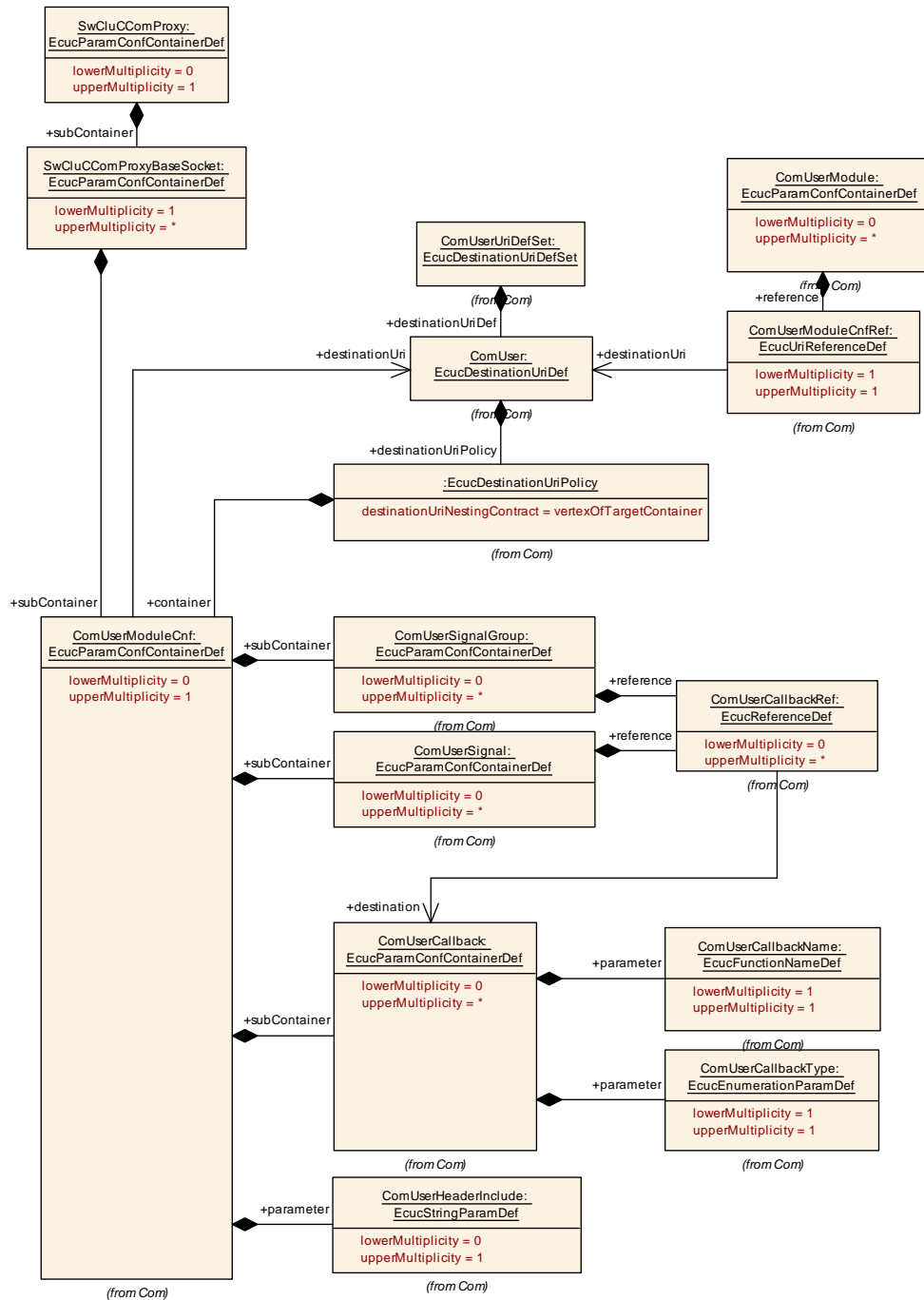


Figure 10.30: Com user module configuration

SWS Item	[ECUC_Com_10030]
Container Name	ComUserModuleCnf
Parent Container	RteComUser, SwCluCComProxyBaseSocket
Destination Uri Definition	ComUser

Description	Contains the configuration parameters of the Com user module. Tags: atp.Status=draft
Post-Build Variant Multiplicity	false
Configuration Parameters	

Name	ComUserHeaderInclude [ECUC_Com_10032]
Parent Container	ComUserModuleCnf
Description	Defines the header file where the Com user provides the function declarations for configured callbacks. Tags: atp.Status=draft
Multiplicity	0..1
Type	EcucStringParamDef
Default Value	
Regular Expression	
Post-Build Variant Value	false
Scope / Dependency	scope: local

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComUserCallback	0..*	This container defines a Com callback function for signals and signal groups. Tags: atp.Status=draft
ComUserSignal	0..*	Contains the configuration parameters of a signal inside a Com user module. Please note that it is valid to define ComUserSignal without any callback function. Tags: atp.Status=draft
ComUserSignalGroup	0..*	Contains the configuration parameters of a signal group inside a Com user module. Please note that it is valid to define ComUserSignalGroup without any callback function. Tags: atp.Status=draft

10.2.7.3.3 Com User Signals and Signal Groups

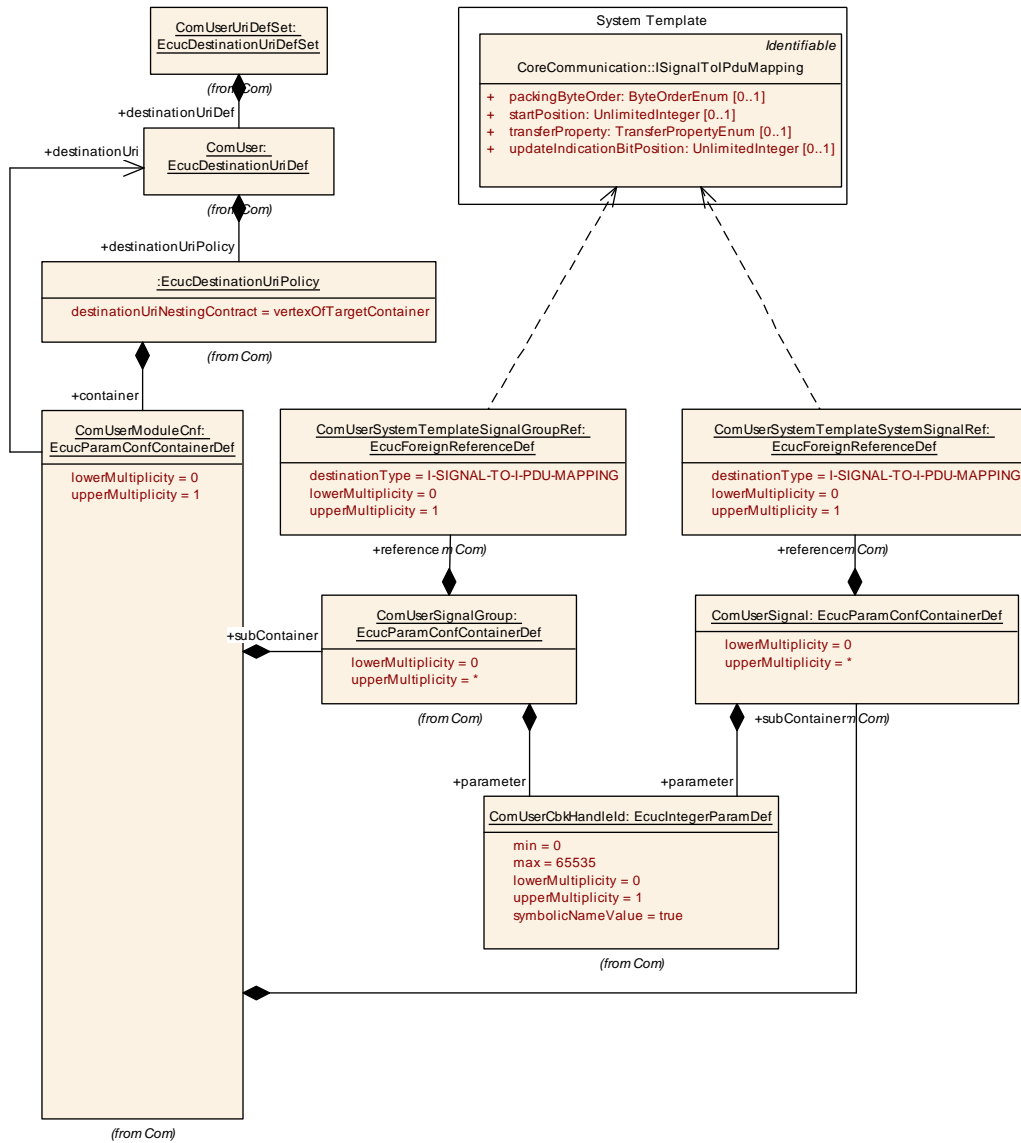


Figure 10.31: Com user signals and signal groups

10.2.7.3.3.1 Com User Signals

SWS Item	[ECUC_Com_10028]
Container Name	ComUserSignal
Parent Container	ComUserModuleCnf
Description	Contains the configuration parameters of a signal inside a Com user module. Please note that it is valid to define ComUserSignal without any callback function. Tags: atp.Status=draft

Post-Build Variant Multiplicity	true
Configuration Parameters	

Name	ComUserCbkJandleId [ECUC_Com_10033]	
Parent Container	ComUserSignal	
Description	<p>The numerical value used as the Com user callback handle Id.</p> <p>This ID identifies signals and signal groups in the COM callbacks using ComUserCbkJandleId parameter respectively.</p> <p>Tags: atp.Status=draft</p>	
Multiplicity	0..1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default Value		
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	
Scope / Dependency	scope: ECU	

Name	ComUserCallbackRef [ECUC_Com_10035]	
Parent Container	ComUserSignal	
Description	<p>Reference(s) to all callback(s) of this signal or signal group.</p> <p>Tags: atp.Status=draft</p>	
Multiplicity	0..*	
Type	Reference to ComUserCallback	
Post-Build Variant Multiplicity	true	
Post-Build Variant Value	true	
Scope / Dependency	scope: ECU	

Name	ComUserSystemTemplateSystemSignalRef [ECUC_Com_10025]	
Parent Container	ComUserSignal	
Description	<p>Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComUserSignal (or ComUserGroupSignal) represents.</p> <p>Tags: atp.Status=draft</p>	
Multiplicity	0..1	
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING	
Post-Build Variant Multiplicity	true	

Post-Build Variant Value	true
Scope / Dependency	scope: ECU

No Included Containers

10.2.7.3.3.2 Com User Signal Groups

SWS Item	[ECUC_Com_10027]
Container Name	ComUserSignalGroup
Parent Container	ComUserModuleCnf
Description	<p>Contains the configuration parameters of a signal group inside a Com user module. Please note that it is valid to define ComUserSignalGroup without any callback function.</p> <p>Tags: atp.Status=draft</p>
Post-Build Variant Multiplicity	true
Configuration Parameters	

Name	ComUserCbkJd [ECUC_Com_10033]	
Parent Container	ComUserSignalGroup	
Description	<p>The numerical value used as the Com user callback handle Id.</p> <p>This ID identifies signals and signal groups in the COM callbacks using ComUserCbkJd parameter respectively.</p> <p>Tags: atp.Status=draft</p>	
Multiplicity	0..1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default Value		
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	
Scope / Dependency	scope: ECU	

Name	ComUserCallbackRef [ECUC_Com_10035]
Parent Container	ComUserSignalGroup
Description	Reference(s) to all callback(s) of this signal or signal group. Tags: atp.Status=draft
Multiplicity	0..*
Type	Reference to ComUserCallback
Post-Build Variant Multiplicity	true
Post-Build Variant Value	true
Scope / Dependency	scope: ECU

Name	ComUserSystemTemplateSignalGroupRef [ECUC_Com_10026]
Parent Container	ComUserSignalGroup
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this ComUserSignalGroup represents. Tags: atp.Status=draft
Multiplicity	0..1
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING
Post-Build Variant Multiplicity	true
Post-Build Variant Value	true
Scope / Dependency	scope: ECU

No Included Containers

10.2.7.3.4 Com User callbacks

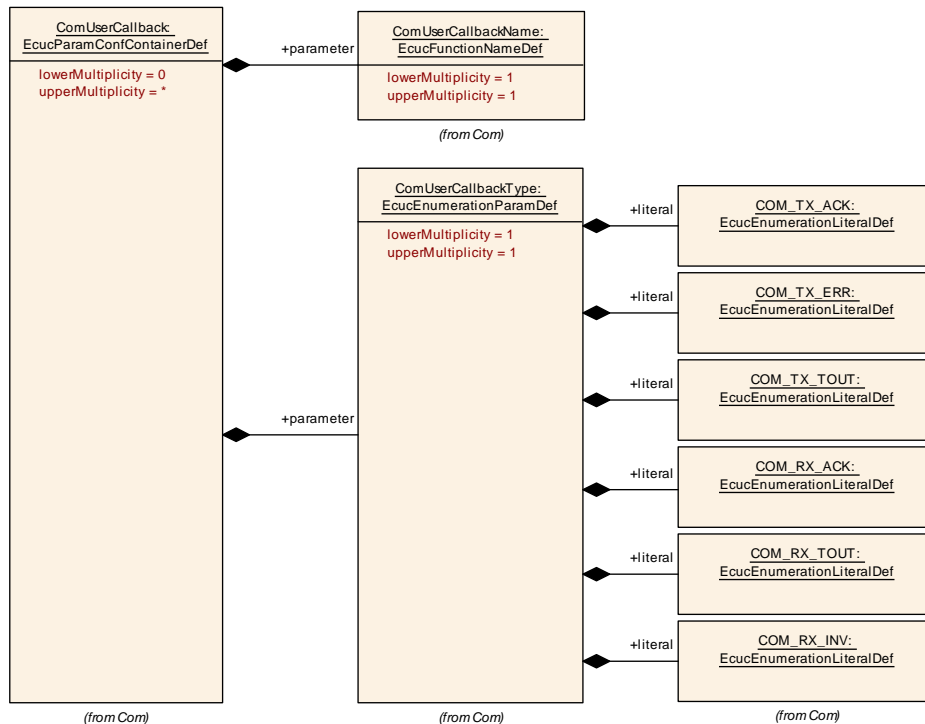


Figure 10.32: Com User callbacks

SWS Item	[ECUC_Com_10038]
Container Name	ComUserCallback
Parent Container	ComUserModuleCnf
Description	This container defines a Com callback function for signals and signal groups. Tags: atp.Status=draft
Post-Build Variant Multiplicity	true
Configuration Parameters	

Name	ComUserCallbackName [ECUC_Com_10036]
Parent Container	ComUserCallback
Description	The name of the callback function to be called. Tags: atp.Status=draft
Multiplicity	1
Type	EcucFunctionNameDef
Default Value	
Regular Expression	
Post-Build Variant Multiplicity	false

Post-Build Variant Value	false
Scope / Dependency	scope: ECU

Name	ComUserCallbackType [ECUC_Com_10034]	
Parent Container	ComUserCallback	
Description	The type of the Com callback Tags: atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	COM_RX_ACK	Com_CbkRxAck callback which is called immediately after the message has been stored in the receiving message object. This type of callback represents a ComNotification. Tags: atp.Status=draft
	COM_RX_INV	Com_CbkInv callback which is called after reception of an invalid signal or signal group respectively. This type of callback represents a ComInvalidNotification. Tags: atp.Status=draft
	COM_RX_TOUT	Com_CbkRxTOut callback which is called immediately after a message reception error has been detected by the deadline monitoring mechanism. This type of callback represents a ComTimeoutNotification. Tags: atp.Status=draft
	COM_TX_ACK	Com_CbkTxAck callback which is called immediately after successful transmission of the I-PDU containing the message. This type of callback represents a ComNotification. Tags: atp.Status=draft

	COM_TX_ERR	Com_CbkTxErr callback which is called in case the transmission is not possible because the corresponding I-PDU group is stopped. This type of callback represents a ComErrorNotification. Tags: atp.Status=draft
	COM_TX_TOUT	Com_CbkTxTOut callback which is called immediately after a message transmission error has been detected by the deadline monitoring mechanism. This type of callback represents a ComTimeoutNotification. Tags: atp.Status=draft
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	
Scope / Dependency	scope: ECU	

No Included Containers

10.2.7.4 LdCom Proxy



Figure 10.33: LdCom Proxy Parameter

SWS Item	[ECUC_SwCluC_00125]
Container Name	SwCluCLdComProxy

Parent Container	SwCluCProxies		
Description	LdCom Proxy specific parameters. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCLdComProxyDevErrorDetect [ECUC_SwCluC_00126]		
Parent Container	SwCluCLdComProxy		
Description	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> • true: detection and notification is enabled. • false: detection and notification is disabled. Tags: atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCLdComProxy BaseSocket	1..*	<p>This container configures how many EcucPartitions specific API links are required for the LdCom Proxy and to which cluster resource the API set belongs.</p> <p>In the LdCom Low Proxy all offered API sets (incl. callbacks) needs to be configured. In the LdCom High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>The LdComUserModuleCnf is only applicable in the Host Software Cluster and shall not be configured in the scope of a Applicative Software Cluster. One LdComUserModuleCnf instance can either server for LdComIPdus on multiple EcucPartitons or exactly one EcucPartiton. How many LdComUser instances an Rte in an related Applicative Software Cluster requires is vendor specific.</p> <p>Tags: atp.Status=draft</p>

10.2.7.4.1 LdCom Base Socket

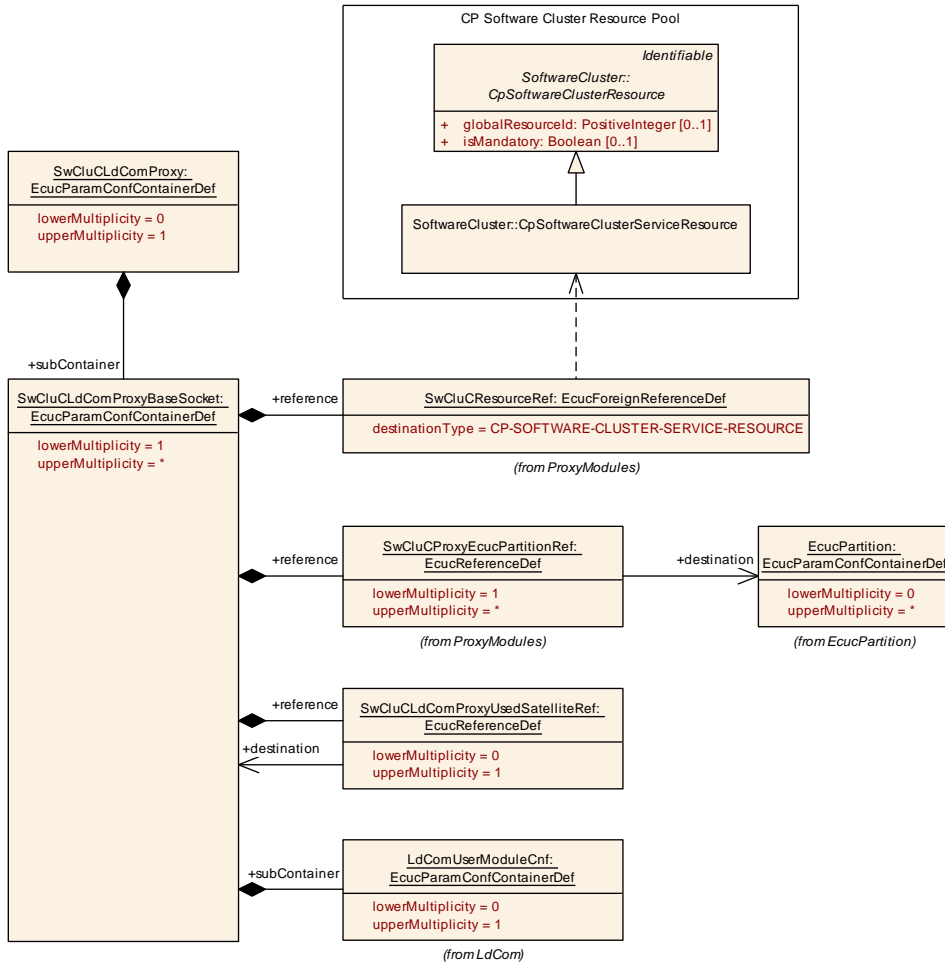


Figure 10.34: LdCom Proxy Base Socket

SWS Item	[ECUC_SwCluC_00129]
Container Name	SwCluCLdComProxyBaseSocket
Parent Container	SwCluCLdComProxy

Description	This container configures how many EcucPartitions specific API links are required for the LdCom Proxy and to which cluster resource the API set belongs.		
	In the LdCom Low Proxy all offered API sets (incl. callbacks) needs to be configured. In the LdCom High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.		
	The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.		
	The LdComUserModuleCnf is only applicable in the Host Software Cluster and shall not be configured in the scope of a Applicative Software Cluster. One LdComUserModuleCnf instance can either server for LdComIPdus on multiple EcucPartitons or exactly one EcucPartiton. How many LdComUser instances an Rte in an related Applicative Software Cluster requires is vendor specific.		
Tags: atp.Status=draft			
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCLdComProxyUsedSatelliteRef [ECUC_SwCluC_00130]		
Parent Container	SwCluCLdComProxyBaseSocket		
Description	Reference to the SwCluCLdComBaseSocket which has access to a satellite of the LdCom. The owning SwCluCLdComBaseSocket uses the LdCom satellite of the referenced SwCluCLdComBaseSocket.		
	Tags: atp.Status=draft		
Multiplicity	0..1		
Type	Reference to SwCluCLdComProxyBaseSocket		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00147]		
Parent Container	SwCluCLdComProxyBaseSocket		

Description	Reference to the EcucPartition. Tags: atp.Status=draft		
Multiplicity	1..*		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef
Parent Container	SwCluCLdComProxyBaseSocket
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

Included Containers		
Container Name	Multiplicity	Scope / Dependency
LdComUserModuleCnf	0..1	Contains the configuration parameters of the LdCom user module. Tags: atp.Status=draft

[SWS_SwCluC_CONSTR_02553]{DRAFT} [SwCluCLdComProxyBaseSocket](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_LDCOM_BASE_SOCKET](#) [The [SwCluCLdComProxyBaseSocket.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_LDCOM_BASE_SOCKET](#).]()

10.2.7.4.2 LdCom User module configuration

SWS Item	[ECUC_LdCom_00030]
Container Name	LdComUserModuleCnf
Parent Container	RteLdComUser, SwCluCLdComProxyBaseSocket

Destination Uri Definition	LdComUser
Description	Contains the configuration parameters of the LdCom user module. Tags: atp.Status=draft
Post-Build Variant Multiplicity	false
Configuration Parameters	

Name	LdComUserHeaderInclude [ECUC_LdCom_00027]
Parent Container	LdComUserModuleCnf
Description	Defines the header file where the LdCom user provides the function declarations for configured callbacks. Tags: atp.Status=draft
Multiplicity	0..1
Type	EcucStringParamDef
Default Value	
Regular Expression	
Post-Build Variant Value	false
Scope / Dependency	scope: local

Included Containers		
Container Name	Multiplicity	Scope / Dependency
LdComUserCallback	0..*	This container defines a LdCom callback function for a LdComIPdu. Tags: atp.Status=draft
LdComUserIPdu	0..*	Contains the configuration parameters for the LdCom's signal (LdComIPdu) inside a LdCom user module. Tags: atp.Status=draft

10.2.7.4.3 LdCom User IPdus

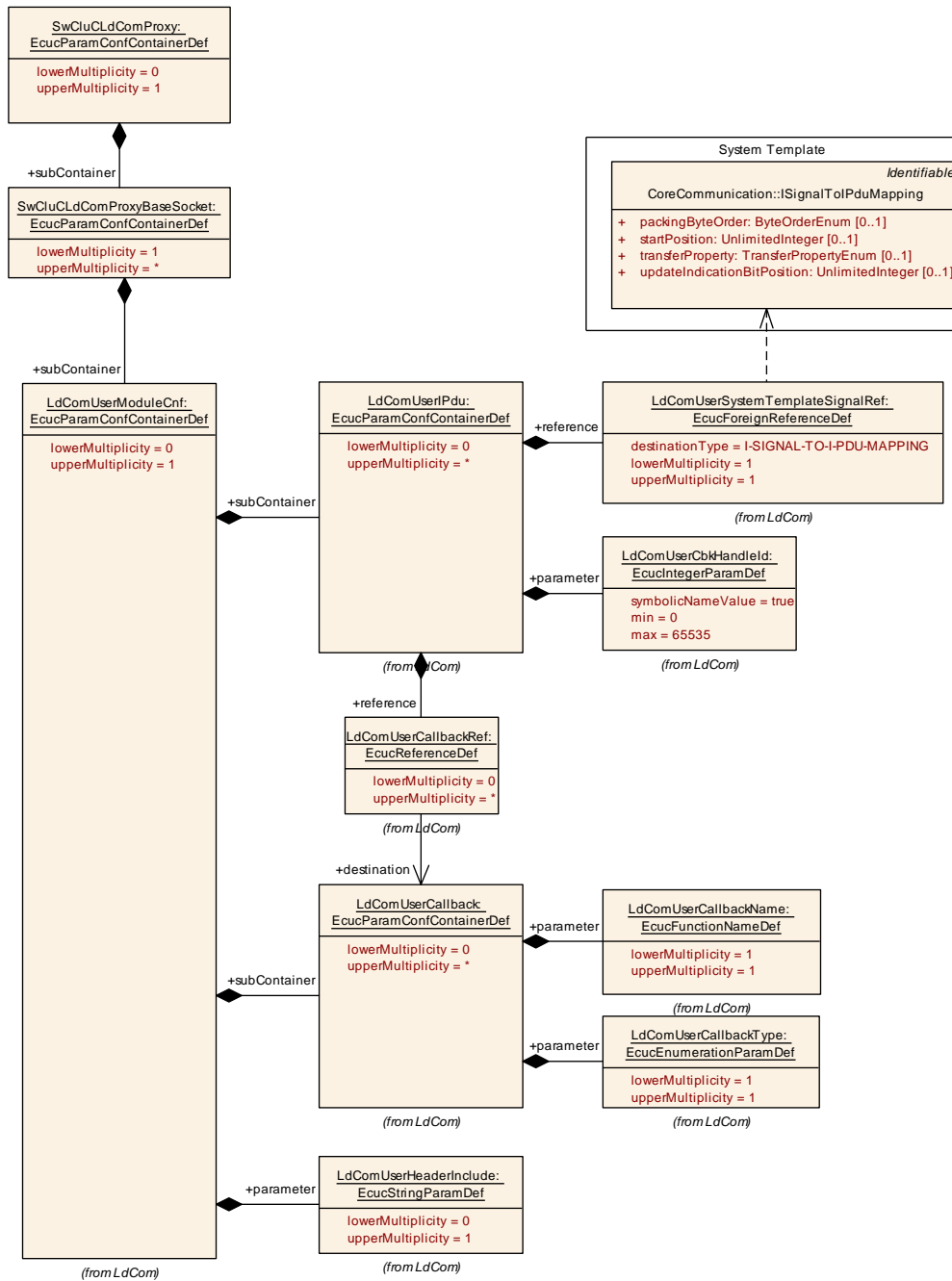


Figure 10.35: LdCom User module configuration

SWS Item	[ECUC_LdCom_00028]
Container Name	LdComUserIPdu
Parent Container	LdComUserModuleCnf
Description	Contains the configuration parameters for the LdCom's signal (LdComIPdu) inside a LdCom user module.
	Tags: atp.Status=draft

Post-Build Variant Multiplicity	true
Configuration Parameters	

Name	LdComUserCbkJandleId [ECUC_LdCom_00026]
Parent Container	LdComUserIPdu
Description	<p>The numerical value used as the LdCom user callback handle Id.</p> <p>This is the ID used by LdCom to invoke callbacks of a LdCom user (Rte, ScCluC LdCom Low Proxy or CDDs) using LdComUserCbkJandleId parameter respectively.</p> <p>A corresponding symbolic name reference is created, which may be used for the invocations of the user.</p> <p>Tags: atp.Status=draft</p>
Multiplicity	1
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)
Range	0 .. 65535
Default Value	
Post-Build Variant Value	false
Scope / Dependency	scope: ECU

Name	LdComUserCallbackRef [ECUC_LdCom_00024]
Parent Container	LdComUserIPdu
Description	<p>Reference(s) to all callback(s) of this LdComIPdu.</p> <p>Tags: atp.Status=draft</p>
Multiplicity	0..*
Type	Reference to LdComUserCallback
Post-Build Variant Multiplicity	true
Post-Build Variant Value	true
Scope / Dependency	scope: ECU

Name	LdComUserSystemTemplateSignalRef [ECUC_LdCom_00033]
Parent Container	LdComUserIPdu
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template). Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING
Post-Build Variant Multiplicity	true
Post-Build Variant Value	true
Scope / Dependency	scope: ECU

No Included Containers

10.2.7.4.4 LdCom User callbacks

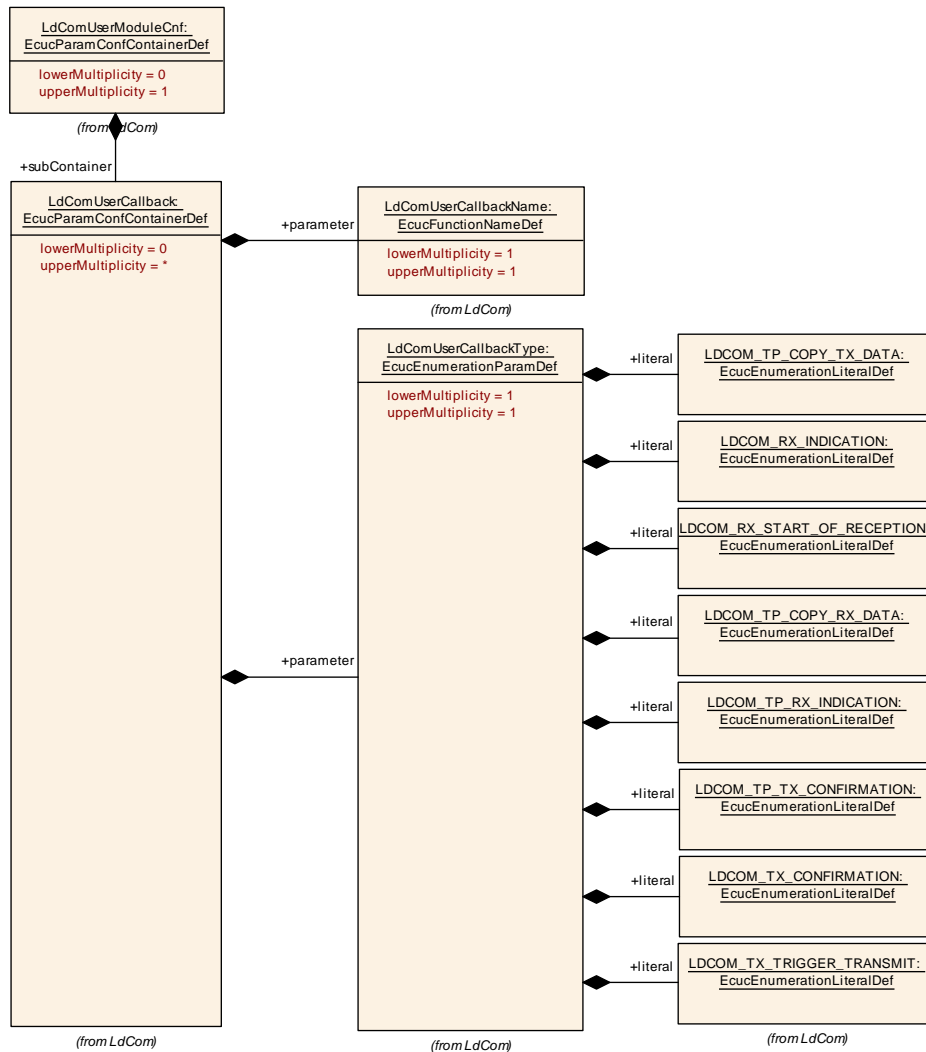


Figure 10.36: LdCom User callbacks

SWS Item	[ECUC_LdCom_00022]
Container Name	LdComUserCallback
Parent Container	LdComUserModuleCnf
Description	This container defines a LdCom callback function for a LdComIPdu. Tags: atp.Status=draft
Post-Build Variant Multiplicity	true
Configuration Parameters	

Name	LdComUserCallbackName [ECUC_LdCom_00023]
Parent Container	LdComUserCallback
Description	The name of the callback function to be called. Tags: atp.Status=draft
Multiplicity	1
Type	EcucFunctionNameDef
Default Value	
Regular Expression	
Post-Build Variant Multiplicity	false
Post-Build Variant Value	false
Scope / Dependency	scope: ECU

Name	LdComUserCallbackType [ECUC_LdCom_00025]	
Parent Container	LdComUserCallback	
Description	The type of the LdCom callback Tags: atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	LDCOM_RX_INDICATION	LdComCbkJxIndication callback indicates a received PDU from a lower layer communication interface module. Tags: atp.Status=draft
	LDCOM_RX_START_OF_RECEPTION	LdComCbkJxStartOfReception callback called at the start of receiving an N-SDU. Tags: atp.Status=draft
	LDCOM_TP_COPY_RX_DATA	LdComCbkJxCopyRxData callback to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Tags: atp.Status=draft
	LDCOM_TP_COPY_TX_DATA	LdComCbkJxCopyTxData callback to acquire the transmit data of an I-PDU segment. Tags: atp.Status=draft

	LDCOM_TP_RX_INDICATION	LdComCbkJpRxIndication callback called after an I-PDU has been received via the TP API Tags: atp.Status=draft
	LDCOM_TP_TX_CONFIRMATION	LdComCbkJpTxConfirmation callback called after a Signal has been transmitted via the TP-API on its network. Tags: atp.Status=draft
	LDCOM_TX_CONFIRMATION	LdComCbkJxConfirmation callback which is called when the lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU. Tags: atp.Status=draft
	LDCOM_TX_TRIGGER_TRANSMIT	LdComCbkJxConfirmation callback which is called when the lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU. Tags: atp.Status=draft
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	
Scope / Dependency	scope: ECU	

No Included Containers

10.2.7.5 Dcm Proxy

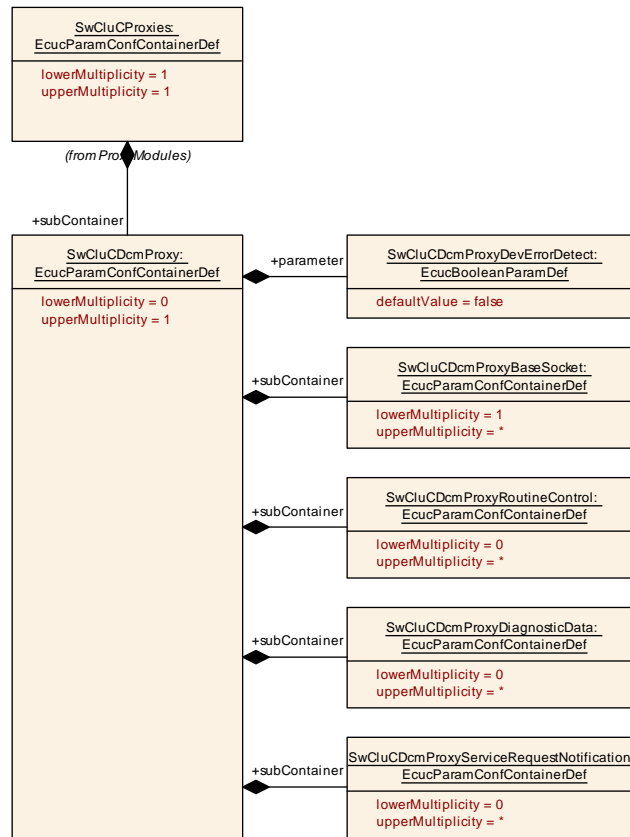


Figure 10.37: Dcm Proxy Parameter

SWS Item	[ECUC_SwCluC_00118]		
Container Name	SwCluCDcmProxy		
Parent Container	SwCluCProxies		
Description	Dcm Proxy specific parameters. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCDcmProxyDevErrorDetect [ECUC_SwCluC_00137]		
Parent Container	SwCluCDcmProxy		
Description	<p>Switches the development error detection and notification on or off.</p> <ul style="list-style-type: none"> • true: detection and notification is enabled. • false: detection and notification is disabled. <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCDcmProxyBaseSocket	1..*	<p>This container configures how many EcucPartitions specific API links are required for the Dcm Proxy and to which cluster resource the API set belongs.</p> <p>In the Dcm Low Proxy all offered API sets needs to be configured. In the Dcm High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>Tags: atp.Status=draft</p>
SwCluCDcmProxyDiagnosticData	0..*	<p>Configures a Diagnostic Data proxy for a DID or PID or IOControl.</p> <p>Tags: atp.Status=draft</p>
SwCluCDcmProxyRoutineControl	0..*	<p>Configures a Routine Control proxy.</p> <p>Tags: atp.Status=draft</p>
SwCluCDcmProxyServiceRequestNotification	0..*	<p>Configures a Service Request Notification proxy.</p> <p>Tags: atp.Status=draft</p>

10.2.7.5.1 Dcm Base Socket

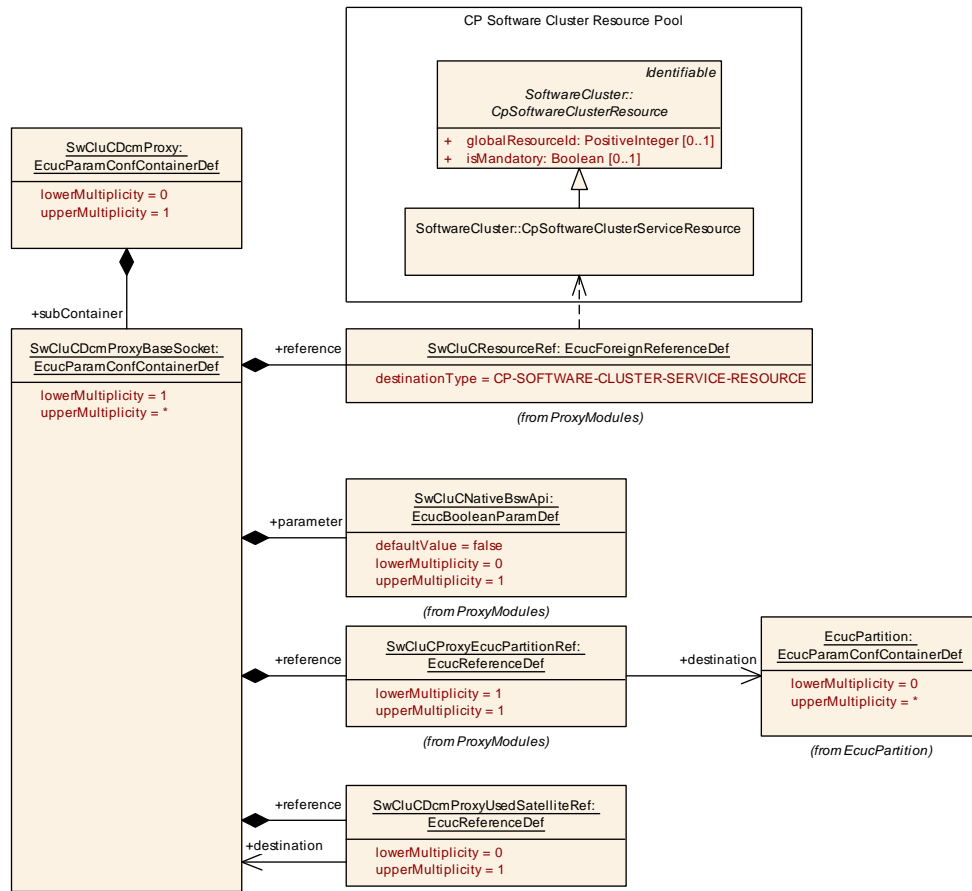


Figure 10.38: Dcm Proxy Base Socket

SWS Item	[ECUC_SwCluC_00120]		
Container Name	SwCluCDcmProxyBaseSocket		
Parent Container	SwCluCDcmProxy		
Description	<p>This container configures how many EcucPartitions specific API links are required for the Dcm Proxy and to which cluster resource the API set belongs.</p> <p>In the Dcm Low Proxy all offered API sets need to be configured. In the Dcm High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Configuration Parameters			

Name	SwCluCNativeBswApi [ECUC_SwCluC_00058]		
Parent Container	SwCluCDcmProxyBaseSocket		
Description	<p>Defines if the native C-API without pre or suffixes is offered in the Applicative Software Cluster on this EcucPartition.</p> <p>This Parameter is only relevant for the NvM High Proxy.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCDcmProxyUsedSatelliteRef [ECUC_SwCluC_00121]		
Parent Container	SwCluCDcmProxyBaseSocket		
Description	<p>Reference to the SwCluCDemBaseSocket which has access to a satellite of the Dcm. The owning SwCluCDemBaseSocket uses the Dcm satellite of the referenced SwCluCDemBaseSocket.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	Reference to SwCluCDcmProxyBaseSocket		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCDcmProxyBaseSocket		
Description	<p>Reference to the EcucPartition.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		

Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef
Parent Container	SwCluCDCmProxyBaseSocket
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_02752]{DRAFT} [SwCluCDCmProxyBaseSocket](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DCM_BASE_SOCKET](#) [The [SwCluCDCmProxyBaseSocket.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DCM_BASE_SOCKET](#).]()

10.2.7.5.2 Dcm Proxy Diagnostic Data

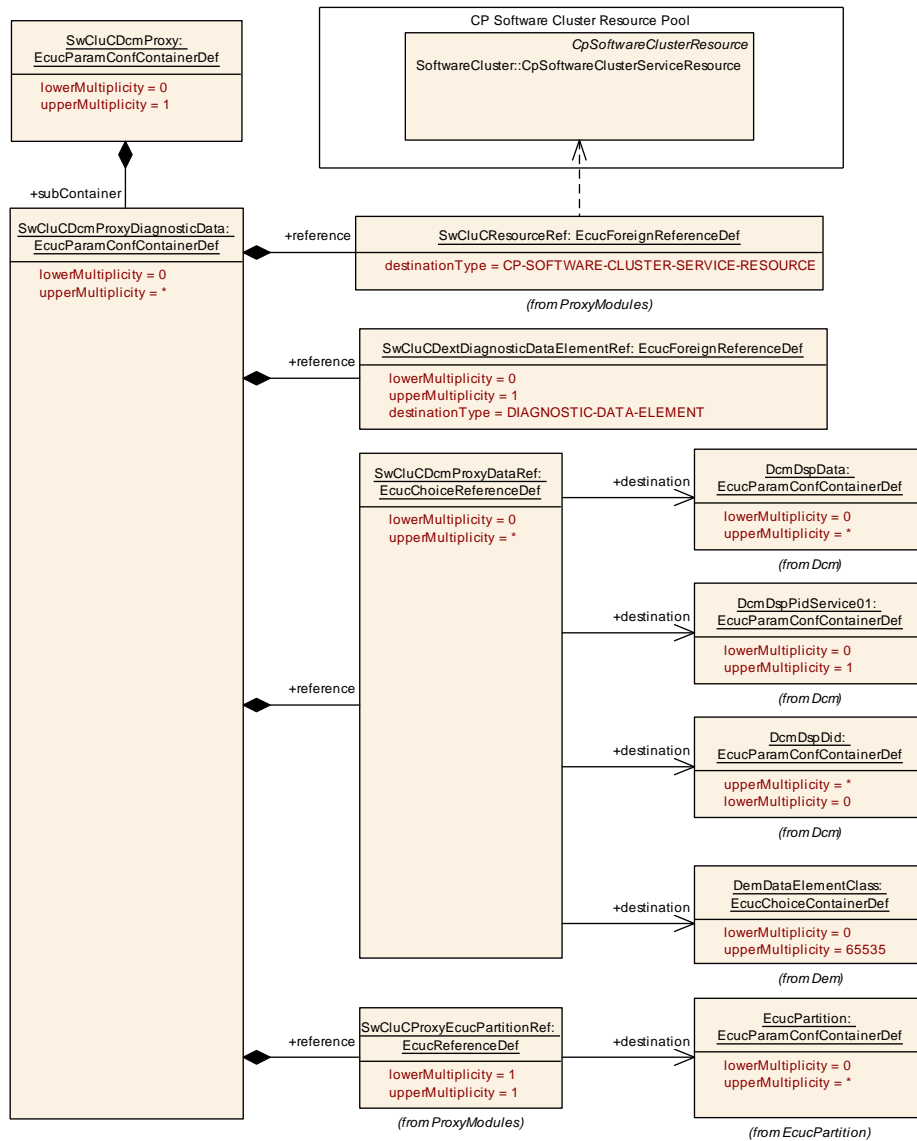


Figure 10.39: Dcm Proxy Diagnostic Data

SWS Item	[ECUC_SwCluC_00122]		
Container Name	SwCluCDcmProxyDiagnosticData		
Parent Container	SwCluCDcmProxy		
Description	Configures a Diagnostic Data proxy for a DID or PID or IOControl. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCdcmProxyDataRef [ECUC_SwCluC_00123]
Parent Container	SwCluCdcmProxyDiagnosticData
Description	<p>Reference to the diagnostic data of Dcm which can either be a DcmDspData, or a DcmDspPidService01, or a DcmDspDid.</p> <p>The DcmDspDid reference is only applicable in the Applicative Sw Cluster if the DID has exactly one Diagnostic Data Element requiring the DID based interface.</p> <p>The reference is mandatory in the Applicative Sw Cluster and can be used in the Host Sw Cluster to identify the related diagnostic data in the Dcm configuration.</p> <p>Tags: atp.Status=draft</p>
Multiplicity	0..*
Type	Choice reference to [DcmDspData, DcmDspDid, DcmDspPidService01, DemDataElementClass]
Scope / Dependency	scope: ECU

Name	SwCluCdextDiagnosticDataElementRef [ECUC_SwCluC_00138]
Parent Container	SwCluCdcmProxyDiagnosticData
Description	<p>Reference to the DiagnosticDataElement in DEXT for back-tracing in case of DEXT based configuration approaches</p> <p>Tags: atp.Status=draft</p>
Multiplicity	0..1
Type	Foreign reference to DIAGNOSTIC-DATA-ELEMENT
Scope / Dependency	scope: ECU

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCdcmProxyDiagnosticData		
Description	<p>Reference to the EcucPartition.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00139]
Parent Container	SwCluCDcmProxyDiagnosticData
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_02427]{DRAFT} [SwCluCDcmProxyDiagnosticData](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DIAGNOSTIC_DATA](#) [The [SwCluCDcmProxyDiagnosticData.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DIAGNOSTIC_DATA](#).]()

10.2.7.5.3 Dcm Proxy Routine Control

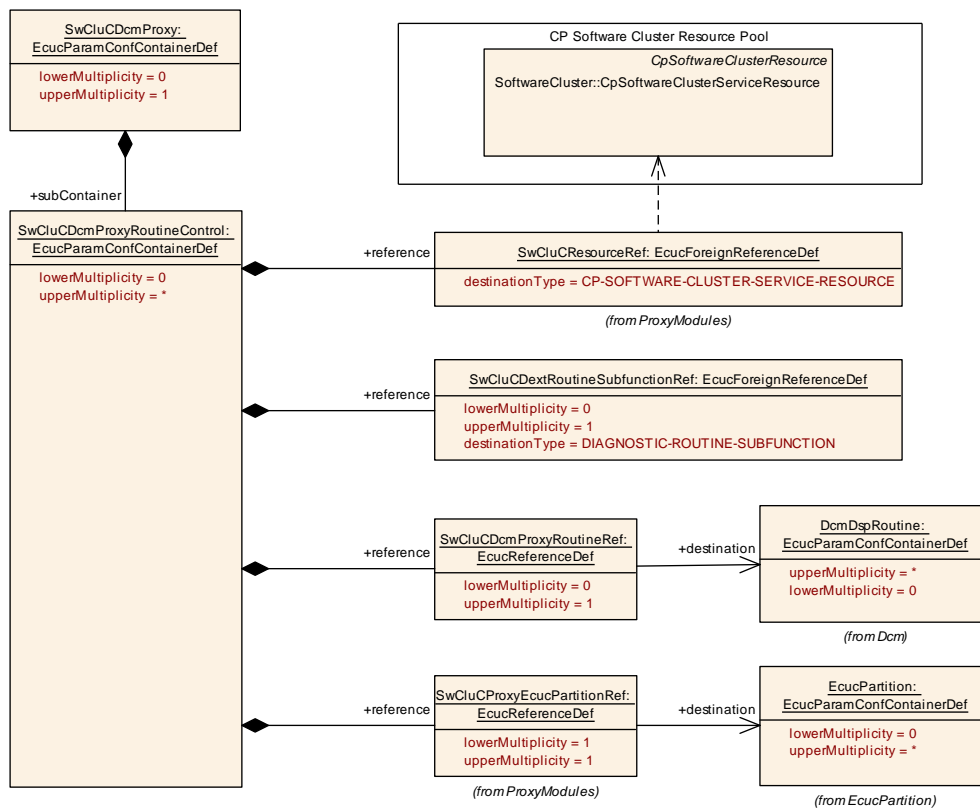


Figure 10.40: Dcm Proxy Routine Control

SWS Item	[ECUC_SwCluC_00133]
Container Name	SwCluCDcmProxyRoutineControl

Parent Container	SwCluCDcmProxy		
Description	Configures a Routine Control proxy. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCDcmProxyRoutineRef [ECUC_SwCluC_00136]		
Parent Container	SwCluCDcmProxyRoutineControl		
Description	Reference to the routine configuration of Dcm. The reference is mandatory in the Applicative Sw Cluster and can be used in the Host Sw Cluster to identify the related routine control in the Dcm configuration. Tags: atp.Status=draft		
Multiplicity	0..1		
Type	Reference to DcmDspRoutine		
Scope / Dependency	scope: ECU		

Name	SwCluCDextRoutineSubfunctionRef [ECUC_SwCluC_00135]		
Parent Container	SwCluCDcmProxyRoutineControl		
Description	Reference to the Routine Subfunction in DEXT for back-tracing in case of DEXT based configuration approaches Tags: atp.Status=draft		
Multiplicity	0..1		
Type	Foreign reference to DIAGNOSTIC-ROUTINE-SUBFUNCTION		
Scope / Dependency	scope: ECU		

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCDcmProxyRoutineControl		
Description	Reference to the EcucPartition. Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00134]
Parent Container	SwCluCDcmProxyRoutineControl
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_02428]{DRAFT} [SwCluCDcmProxyRoutineControl](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DCM_ROUTINE_CONTROL](#) [The [SwCluCDcmProxyRoutineControl.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DCM_ROUTINE_CONTROL](#).]()

10.2.7.5.4 Dcm Proxy Service Request Notification

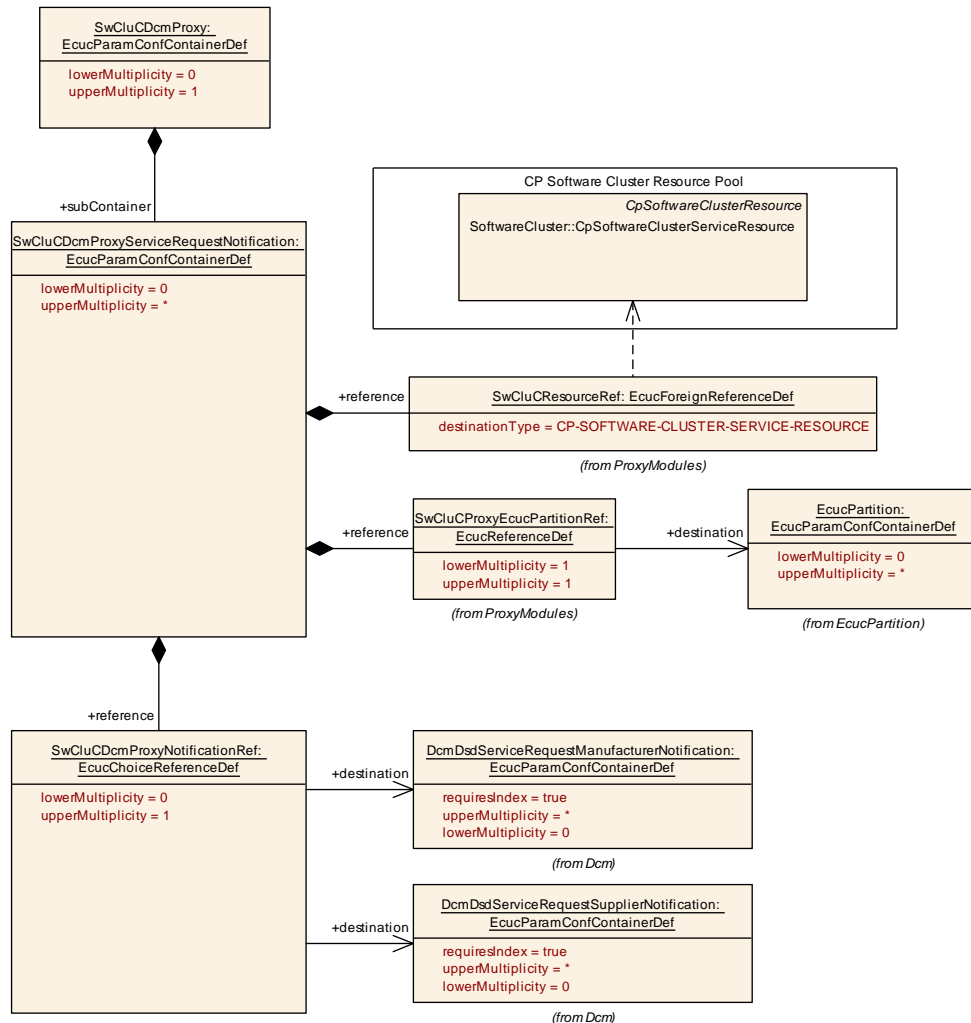


Figure 10.41: Dcm Proxy Service Request Notification

SWS Item	[ECUC_SwCluC_00145]		
Container Name	SwCluCDcmProxyServiceRequestNotification		
Parent Container	SwCluCDcmProxy		
Description	Configures a Service Request Notification proxy. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Configuration Parameters			

Name	SwCluCDcmProxyNotificationRef [ECUC_SwCluC_00146]
Parent Container	SwCluCDcmProxyServiceRequestNotification

Description	Reference to the Service Request Notification configuration of Dcm which can either be a DcmDsdServiceRequestSupplierNotification or DcmDsdServiceRequestManufacturerNotification The reference is mandatory in the Applicative Sw Cluster and can be used in the Host Sw Cluster to identify the related notification in the Dcm configuration. Tags: atp.Status=draft
Multiplicity	0..1
Type	Choice reference to [DcmDsdServiceRequestManufacturerNotification, DcmDsdServiceRequestSupplierNotification]
Scope / Dependency	scope: ECU

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCDcmProxyServiceRequestNotification		
Description	Reference to the EcucPartition. Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00143]		
Parent Container	SwCluCDcmProxyServiceRequestNotification		
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft		
Multiplicity	1		
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE		
Scope / Dependency	scope: ECU		

No Included Containers

[SWS_SwCluC_CONSTR_02429]{DRAFT} [SwCluCDcmProxyServiceRequestNotification](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DCM_SERVICE_REQUEST_NOTIFICATION](#) [The [SwCluCDcmProxyServiceRequestNotification.SwCluCResourceRef](#) shall

only reference a `CpSoftwareClusterServiceResource` of category `SWCLUSTER_RES_DCM_SERVICE_REQUEST_NOTIFICATION`.|()

10.2.7.6 Dem Proxy

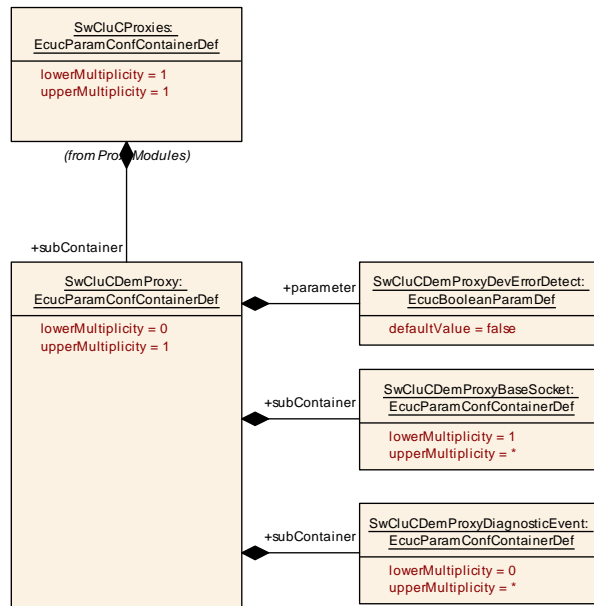


Figure 10.42: Dem Proxy Parameter

SWS Item	[ECUC_SwCluC_00112]		
Container Name	SwCluCDemProxy		
Parent Container	SwCluCProxies		
Description	DemProxy specific parameters. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCDemProxyDevErrorDetect [ECUC_SwCluC_00113]		
Parent Container	SwCluCDemProxy		
Description	<p>Switches the development error detection and notification on or off.</p> <ul style="list-style-type: none"> • true: detection and notification is enabled. • false: detection and notification is disabled. <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCDemProxyBaseSocket	1..*	<p>This container configures how many EcucPartitions specific API links are required for the Dem Proxy and to which cluster resource the API set belongs.</p> <p>In the Dem Low Proxy all offered API sets needs to be configured. In the Dem High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>Tags: atp.Status=draft</p>
SwCluCDemProxyDiagnosticEvent	0..*	<p>Configures a Diagnostic Event proxy.</p> <p>Tags: atp.Status=draft</p>

10.2.7.6.1 Dem Base Socket

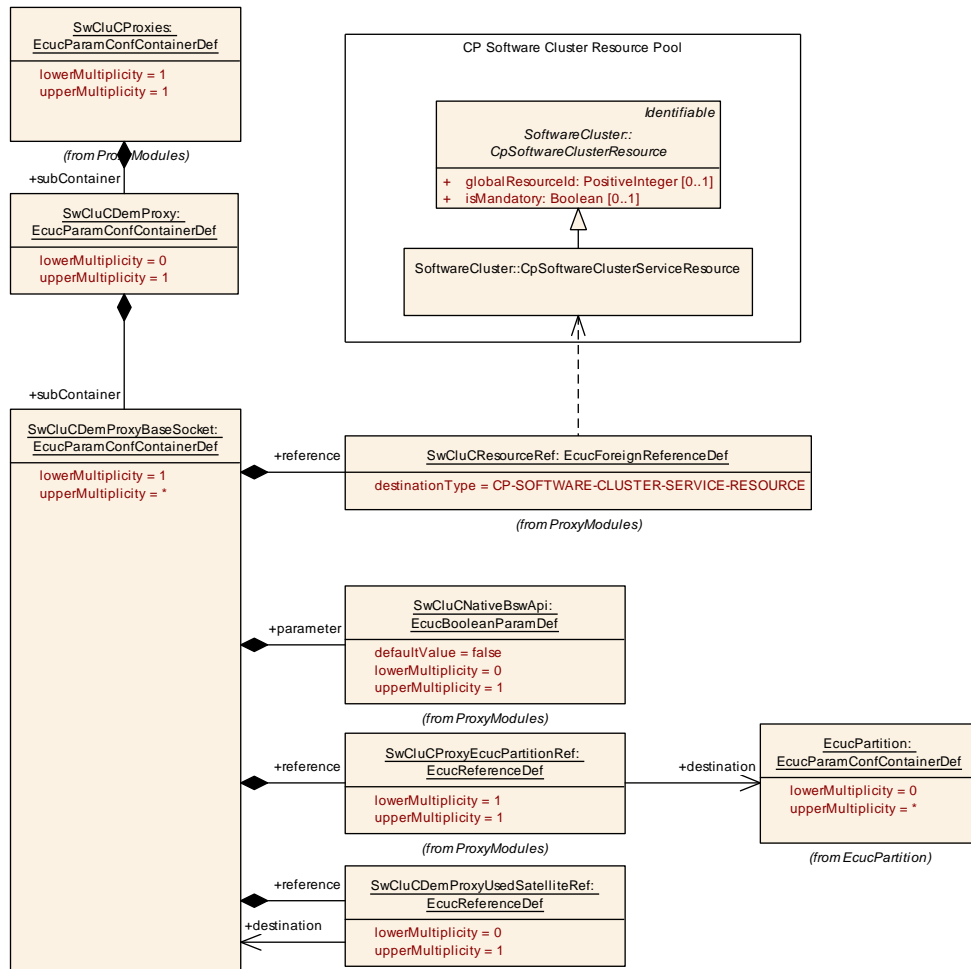


Figure 10.43: Dem Proxy Base Socket

SWS Item	[ECUC_SwCluC_00114]
Container Name	SwCluCDEMProxyBaseSocket
Parent Container	SwCluCDEMProxy
Description	<p>This container configures how many EcucPartitions specific API links are required for the Dem Proxy and to which cluster resource the API set belongs.</p> <p>In the Dem Low Proxy all offered API sets needs to be configured. In the Dem High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>Tags: atp.Status=draft</p>

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCNativeBswApi [ECUC_SwCluC_00058]		
Parent Container	SwCluCDemProxyBaseSocket		
Description	<p>Defines if the native C-API without pre or suffixes is offered in the Applicative Software Cluster on this EcucPartition.</p> <p>This Parameter is only relevant for the NvM High Proxy.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCDemProxyUsedSatelliteRef [ECUC_SwCluC_00115]		
Parent Container	SwCluCDemProxyBaseSocket		
Description	<p>Reference to the SwCluCDemBaseSocket which has access to a satellite of the Dem. The owning SwCluCDemBaseSocket uses the Dem satellite of the referenced SwCluCDemBaseSocket.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	Reference to SwCluCDemProxyBaseSocket		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCDemProxyBaseSocket		
Description	Reference to the EcucPartition. Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef		
Parent Container	SwCluCDemProxyBaseSocket		
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft		
Multiplicity	1		
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE		
Scope / Dependency	scope: ECU		

No Included Containers

[SWS_SwCluC_CONSTR_02720]{DRAFT} [SwCluCDemProxyBaseSocket](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DEM_BASE_SOCKET](#) [The [SwCluCDemProxyBaseSocket.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_DEM_BASE_SOCKET](#).]()

10.2.7.6.2 Dem Proxy Diagnostic Event

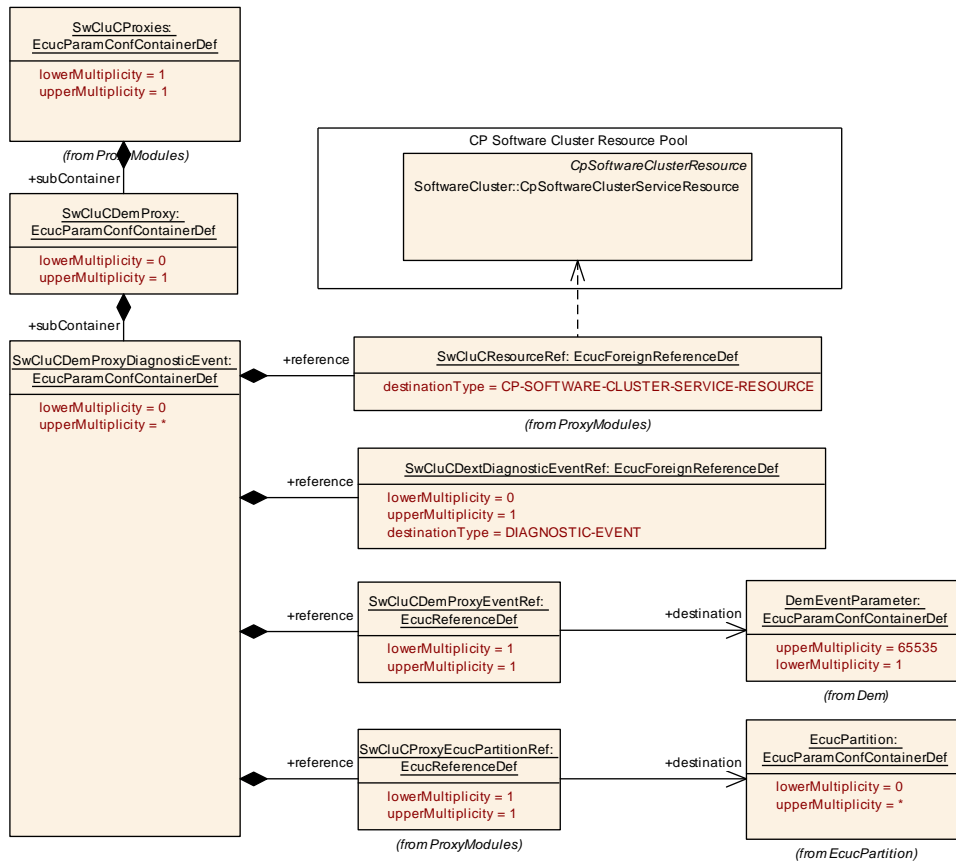


Figure 10.44: Dem Proxy Diagnostic Event

SWS Item	[ECUC_SwCluC_00116]		
Container Name	SwCluCDemProxyDiagnosticEvent		
Parent Container	SwCluCDemProxy		
Description	Configures a Diagnostic Event proxy. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCDemProxyEventRef [ECUC_SwCluC_00117]
Parent Container	SwCluCDemProxyDiagnosticEvent
Description	Reference to the diagnostic event of Dem Tags: atp.Status=draft
Multiplicity	1
Type	Reference to DemEventParameter

Scope / Dependency	scope: ECU
---------------------------	------------

Name	SwCluCDExtDiagnosticEventRef [ECUC_SwCluC_00140]
Parent Container	SwCluCDemProxyDiagnosticEvent
Description	Reference to the Diagnostic Event in DEXT for back-tracing in case of DEXT based configuration approaches Tags: atp.Status=draft
Multiplicity	0..1
Type	Foreign reference to DIAGNOSTIC-EVENT
Scope / Dependency	scope: ECU

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCDemProxyDiagnosticEvent		
Description	Reference to the EcucPartition. Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00141]
Parent Container	SwCluCDemProxyDiagnosticEvent
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_02721]{DRAFT} [SwCluCDemProxyDiagnosticEvent](#) relates only to a [CpSoftwareClusterServiceResource](#) of category **SWCLUSTER_RES_DIAGNOSTIC_EVENT** [The [SwCluCDemProxyDiagnosticEvent.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category **SWCLUSTER_RES_DIAGNOSTIC_EVENT**.]()

10.2.7.7 FiM Proxy

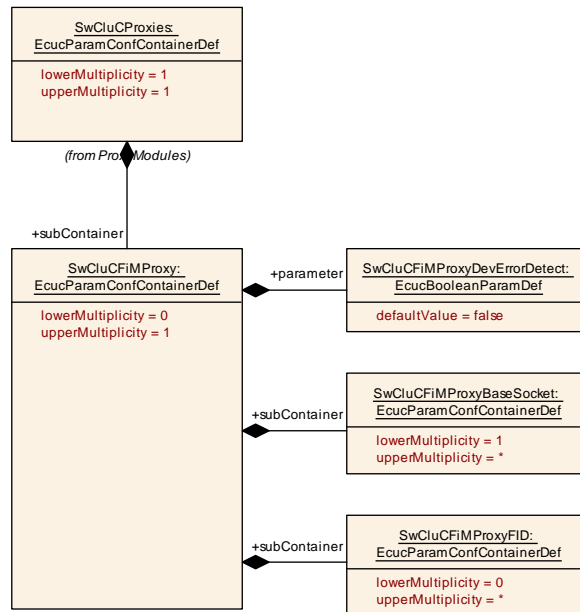


Figure 10.45: FiM Proxy Parameter

SWS Item	[ECUC_SwCluC_00105]		
Container Name	SwCluCFiMProxy		
Parent Container	SwCluCProxies		
Description	FiM Proxy specific parameters. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCFiMProxyDevErrorDetect [ECUC_SwCluC_00106]
Parent Container	SwCluCFiMProxy
Description	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> • true: detection and notification is enabled. • false: detection and notification is disabled. Tags: atp.Status=draft
Multiplicity	1
Type	EcucBooleanParamDef
Default Value	false
Post-Build Variant Value	false

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SwCluCFiMProxyBaseSocket	1..*	<p>This container configures how many EcucPartitions specific API links are required for the FiM Proxy and to which cluster resource the API set belongs.</p> <p>In the FiM Low Proxy all offered API sets needs to be configured. In the FiM High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>Tags: atp.Status=draft</p>
SwCluCFiMProxyFID	0..*	<p>Configures a Function Inhibition proxy.</p> <p>Tags: atp.Status=draft</p>

10.2.7.7.1 FiM Base Socket

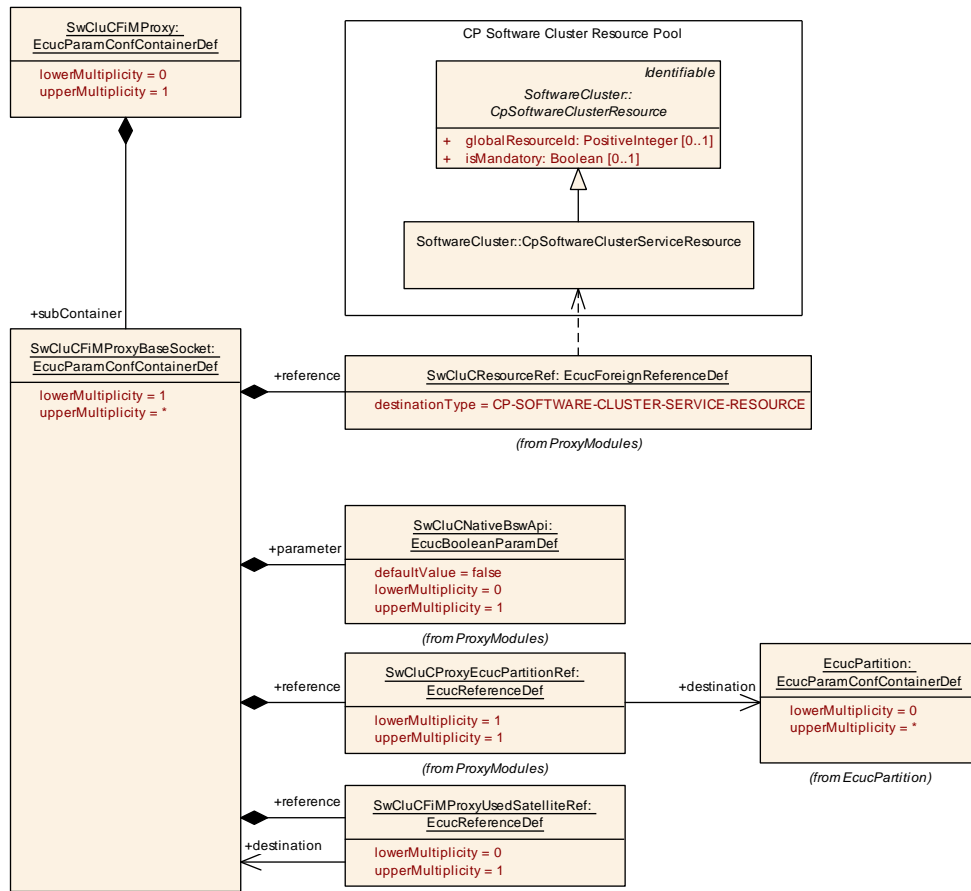


Figure 10.46: FiM Proxy Base Socket

SWS Item	[ECUC_SwCluC_00107]		
Container Name	SwCluCFiMProxyBaseSocket		
Parent Container	SwCluCFiMProxy		
Description	<p>This container configures how many EcucPartitions specific API links are required for the FiM Proxy and to which cluster resource the API set belongs.</p> <p>In the FiM Low Proxy all offered API sets need to be configured. In the FiM High Proxy only the ones are configured which are offered inside this Applicative Software Cluster.</p> <p>The reference SwCluCProxyEcucPartitionRef denotes the EcucPartition on which the API set is provided.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCNativeBswApi [ECUC_SwCluC_00058]		
Parent Container	SwCluCFiMProxyBaseSocket		
Description	<p>Defines if the native C-API without pre or suffixes is offered in the Applicative Software Cluster on this EcucPartition.</p> <p>This Parameter is only relevant for the NvM High Proxy.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCFiMProxyUsedSatelliteRef [ECUC_SwCluC_00108]		
Parent Container	SwCluCFiMProxyBaseSocket		
Description	<p>Reference to the SwCluCFiMBaseSocket which has access to a satellite of the FiM. The owning SwCluCFiMBaseSocket uses the FiM satellite of the referenced SwCluCFiMBaseSocket.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	0..1		
Type	Reference to SwCluCFiMProxyBaseSocket		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCFiMProxyBaseSocket		
Description	<p>Reference to the EcucPartition.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		

Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef
Parent Container	SwCluCFiMProxyBaseSocket
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_03210]{DRAFT} [SwCluCFiMProxyBaseSocket](#) relates only to a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_FIM_BASE_SOCKET](#) [The [SwCluCFiMProxyBaseSocket.SwCluCResourceRef](#) shall only reference a [CpSoftwareClusterServiceResource](#) of category [SWCLUSTER_RES_FIM_BASE_SOCKET](#).]()

10.2.7.7.2 FiM Proxy FID

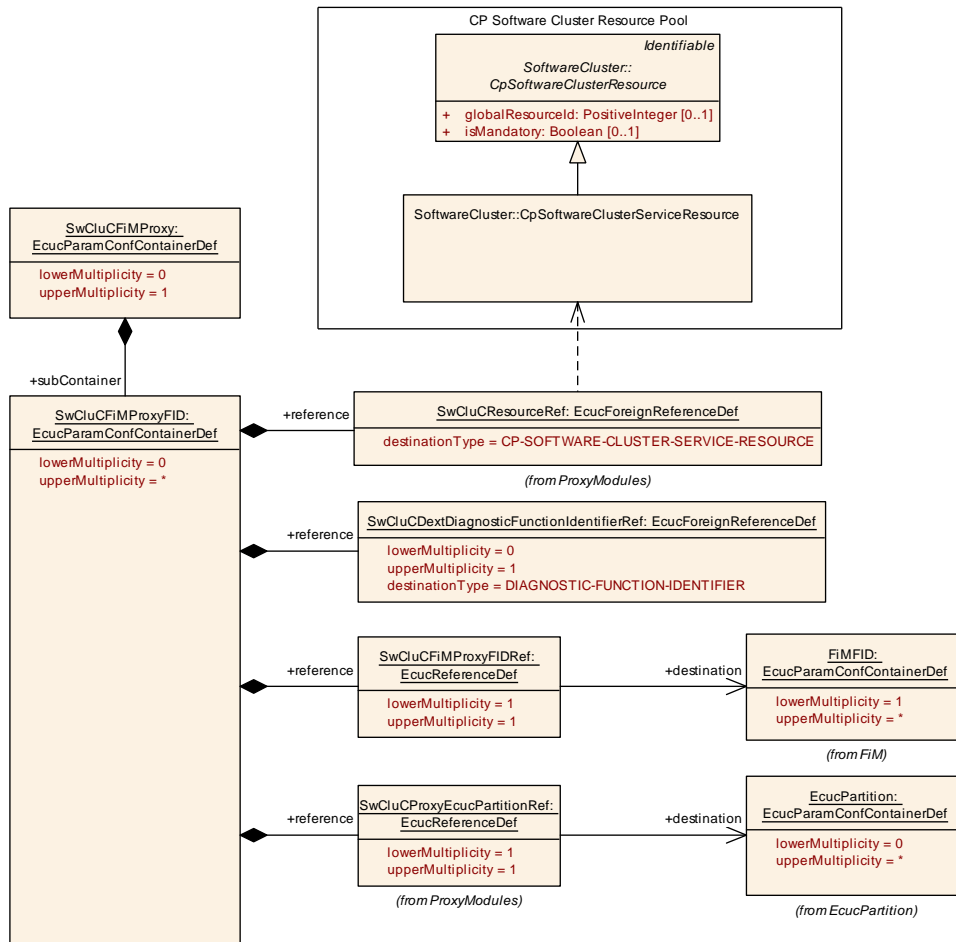


Figure 10.47: FiM Proxy FID

SWS Item	[ECUC_SwCluC_00109]		
Container Name	SwCluCFiMProxyFID		
Parent Container	SwCluCFiMProxy		
Description	Configures a Function Inhibition proxy. Tags: atp.Status=draft		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	SwCluCDextDiagnosticFunctionIdentifierRef [ECUC_SwCluC_00142]
Parent Container	SwCluCFiMProxyFID

Description	Reference to the DiagnosticFunctionIdentifier in DEXT for back-tracing in case of DEXT based configuration approaches Tags: atp.Status=draft
Multiplicity	0..1
Type	Foreign reference to DIAGNOSTIC-FUNCTION-IDENTIFIER
Scope / Dependency	scope: ECU

Name	SwCluCFiMProxyFIDRef [ECUC_SwCluC_00110]
Parent Container	SwCluCFiMProxyFID
Description	Reference to the FIMFID Tags: atp.Status=draft
Multiplicity	1
Type	Reference to FiMFID
Scope / Dependency	scope: ECU

Name	SwCluCProxyEcucPartitionRef [ECUC_SwCluC_00059]		
Parent Container	SwCluCFiMProxyFID		
Description	Reference to the EcucPartition. Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to EcucPartition		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	SwCluCResourceRef [ECUC_SwCluC_00141]
Parent Container	SwCluCFiMProxyFID
Description	Reference to the CpSoftwareClusterServiceResource. Tags: atp.Status=draft
Multiplicity	1
Type	Foreign reference to CP-SOFTWARE-CLUSTER-SERVICE-RESOURCE
Scope / Dependency	scope: ECU

No Included Containers

[SWS_SwCluC_CONSTR_03211]{DRAFT} **SwCluCFiMProxyFID** relates only to a **CpSoftwareClusterServiceResource** of category **SWCLUSTER_RES_FID** [The **SwCluCFiMProxyFID.SwCluCResourceRef** shall only reference a **CpSoftwareClusterServiceResource** of category **SWCLUSTER_RES_FIM_FID**.]()

10.3 Published Information

For details refer to the chapter 10.3 “Published Information” in SWS_BSWGeneral.

A Not applicable requirements

[SWS_SwCluC_00999]{DRAFT} [These requirements are not applicable to this specification.] ([SRS_BSW_00344](#), [SRS_BSW_00404](#), [SRS_BSW_00405](#), [SRS_BSW_00170](#), [SRS_BSW_00380](#), [SRS_BSW_00419](#), [SRS_BSW_00438](#), [SRS_BSW_00375](#), [SRS_BSW_00416](#), [SRS_BSW_00406](#), [SRS_BSW_00467](#), [SRS_BSW_00437](#), [SRS_BSW_00168](#), [SRS_BSW_00423](#), [SRS_BSW_00424](#), [SRS_BSW_00425](#), [SRS_BSW_00426](#), [SRS_BSW_00427](#), [SRS_BSW_00432](#), [SRS_BSW_00433](#), [SRS_BSW_00450](#), [SRS_BSW_00461](#), [SRS_BSW_00451](#), [SRS_BSW_00478](#), [SRS_BSW_00339](#), [SRS_BSW_00422](#), [SRS_BSW_00417](#), [SRS_BSW_00409](#), [SRS_BSW_00452](#), [SRS_BSW_00458](#), [SRS_BSW_00466](#), [SRS_BSW_00488](#), [SRS_BSW_00469](#), [SRS_BSW_00470](#), [SRS_BSW_00471](#), [SRS_BSW_00472](#), [SRS_BSW_00473](#), [SRS_BSW_00161](#), [SRS_BSW_00162](#), [SRS_BSW_00005](#), [SRS_BSW_00415](#), [SRS_BSW_00164](#), [SRS_BSW_00325](#), [SRS_BSW_00342](#), [SRS_BSW_00343](#), [SRS_BSW_00160](#), [SRS_BSW_00453](#), [SRS_BSW_00456](#), [SRS_BSW_00457](#), [SRS_BSW_00479](#), [SRS_BSW_00483](#), [SRS_BSW_00007](#), [SRS_BSW_00413](#), [SRS_BSW_00347](#), [SRS_BSW_00307](#), [SRS_BSW_00373](#), [SRS_BSW_00335](#), [SRS_BSW_00410](#), [SRS_BSW_00411](#), [SRS_BSW_00463](#), [SRS_BSW_00481](#), [SRS_BSW_00346](#), [SRS_BSW_00314](#), [SRS_BSW_00447](#), [SRS_BSW_00348](#), [SRS_BSW_00353](#), [SRS_BSW_00361](#), [SRS_BSW_00301](#), [SRS_BSW_00302](#), [SRS_BSW_00328](#), [SRS_BSW_00312](#), [SRS_BSW_00006](#), [SRS_BSW_00439](#), [SRS_BSW_00448](#), [SRS_BSW_00449](#), [SRS_BSW_00357](#), [SRS_BSW_00377](#), [SRS_BSW_00304](#), [SRS_BSW_00378](#), [SRS_BSW_00306](#), [SRS_BSW_00308](#), [SRS_BSW_00309](#), [SRS_BSW_00484](#), [SRS_BSW_00485](#), [SRS_BSW_00486](#), [SRS_BSW_00414](#), [SRS_BSW_00359](#), [SRS_BSW_00360](#), [SRS_BSW_00440](#), [SRS_BSW_00330](#), [SRS_BSW_00331](#), [SRS_BSW_00454](#), [SRS_BSW_00477](#), [SRS_BSW_00459](#), [SRS_BSW_00460](#))

B Referenced Meta Classes

Class	AbstractAccessPoint (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::AccessCount			
Note	Abstract class indicating an access point from an ExecutableEntity.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , Referrable			
Subclasses	AsynchronousServerCallResultPoint, ExternalTriggeringPointIdent, InternalTriggeringPoint, ModeAccessPointIdent, ModeSwitchPoint , ParameterAccess , ServerCallPoint , VariableAccess			
Attribute	Type	Mult.	Kind	Note
returnValue Provision	RteApiReturnValue ProvisionEnum	0..1	attr	This attribute controls the provision of return values for RTE APIs that correspond to the enclosing access point.

Table B.1: AbstractAccessPoint

Class	ApplicationArrayDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	An application data type which is an array, each element is of the same application data type. Tags: atp.recommendedPackage=ApplicationDataTypes			
Base	ARElement, ARObject, ApplicationCompositeDataType, ApplicationDataType , AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow if it is a variable size array.
element	ApplicationArrayElement	0..1	aggr	This association implements the concept of an array element. That is, in some cases it is necessary to be able to identify single array elements, e.g. as input values for an interpolation routine.

Table B.2: ApplicationArrayDataType

Class	ApplicationArrayElement			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Describes the properties of the elements of an application array data type.			
Base	ARObject, ApplicationCompositeElementDataPrototype , AtpFeature, AtpPrototype, DataPrototype , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
arraySize Handling	ArraySizeHandling Enum	0..1	attr	The way how the size of the array is handled.
arraySize Semantics	ArraySizeSemanticsEnum	0..1	attr	This attribute controls how the information about the array size shall be interpreted.
indexDataType	ApplicationPrimitiveDataType	0..1	ref	This reference can be taken to assign a CompuMethod of category TEXTTABLE to the array. The texttable entries associate a textual value to an index number such that the element with that index number is represented by a symbolic name.





Class	ApplicationArrayElement			
maxNumberOfElements	PositiveInteger	0..1	attr	The maximum number of elements that the array can contain. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table B.3: ApplicationArrayElement

Class	ApplicationCompositeElementDataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	This class represents a data prototype which is aggregated within a composite application data type (record or array). It is introduced to provide a better distinction between target and context in instance Refs.			
Base	ARObject, AtpFeature, AtpPrototype, DataPrototype, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	ApplicationArrayElement, ApplicationRecordElement			
Attribute	Type	Mult.	Kind	Note
type	ApplicationDataType	0..1	tref	This represents the corresponding data type. Stereotypes: isOfType

Table B.4: ApplicationCompositeElementDataPrototype

Class	ApplicationDataType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	ApplicationDataType defines a data type from the application point of view. Especially it should be used whenever something "physical" is at stake. An ApplicationDataType represents a set of values as seen in the application model, such as measurement units. It does not consider implementation details such as bit-size, endianness, etc. It should be possible to model the application level aspects of a VFB system by using ApplicationDataTypes only.			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	ApplicationCompositeDataType, ApplicationPrimitiveDataType			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table B.5: ApplicationDataType

Class	ApplicationError			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This is a user-defined error that is associated with an element of an AUTOSAR interface. It is specific for the particular functionality or service provided by the AUTOSAR software component.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
errorCode	Integer	0..1	attr	The RTE generator is forced to assign this value to the corresponding error symbol. Note that for error codes certain ranges are predefined (see RTE specification).

Table B.6: ApplicationError

Class	ApplicationPrimitiveDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	A primitive data type defines a set of allowed values. Tags: atp.recommendedPackage=ApplicationDataTypes			
Base	<i>ARElement, ARObject, ApplicationDataType, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table B.7: ApplicationPrimitiveDataType

Class	ApplicationRecordDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	An application data type which can be decomposed into prototypes of other application data types. Tags: atp.recommendedPackage=ApplicationDataTypes			
Base	<i>ARElement, ARObject, ApplicationCompositeDataType, ApplicationDataType, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
element (ordered)	ApplicationRecordElement	*	aggr	Specifies an element of a record. The aggregation of ApplicationRecordElement is subject to variability with the purpose to support the conditional existence of elements inside a ApplicationrecordDataType. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table B.8: ApplicationRecordDataType

Class	ApplicationRecordElement			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Describes the properties of one particular element of an application record data type.			
Base	<i>ARObject, ApplicationCompositeElementDataPrototype, AtpFeature, AtpPrototype, DataPrototype, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
isOptional	Boolean	0..1	attr	This attribute represents the ability to declare the enclosing ApplicationRecordElement as optional. This means the that, at runtime, the ApplicationRecordElement may or may not have a valid value and shall therefore be ignored. The underlying runtime software provides means to set the ApplicationRecordElement as not valid at the sending end of a communication and determine its validity at the receiving end.

Table B.9: ApplicationRecordElement

Class	ArgumentDataPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	An argument of an operation, much like a data element, but also carries direction information and is owned by a particular ClientServerOperation.			
Base	<i>ARObject, AtpFeature, AtpPrototype, AutosarDataPrototype, DataPrototype, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
direction	ArgumentDirectionEnum	0..1	attr	This attribute specifies the direction of the argument prototype.
serverArgumentImplPolicy	ServerArgumentImplPolicyEnum	0..1	attr	This defines how the argument type of the servers RunnableEntity is implemented. If the attribute is not defined this has the same semantics as if the attribute is set to the value useArgumentType for primitive arguments and structures.

Table B.10: ArgumentDataPrototype

Enumeration	ArraySizeSemanticsEnum
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes
Note	This type controls how the information about the number of elements in an ApplicationArrayDataType is to be interpreted.
Literal	Description
fixedSize	This means that the ApplicationArrayDataType will always have a fixed number of elements. Tags: atp.EnumerationLiteralIndex=0
variableSize	This implies that the actual number of elements in the ApplicationArrayDataType might vary at run-time. The value of arraySize represents the maximum number of elements in the array. Tags: atp.EnumerationLiteralIndex=1

Table B.11: ArraySizeSemanticsEnum

Class	AssemblySwConnector			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	AssemblySwConnectors are exclusively used to connect SwComponentPrototypes in the context of a CompositionSwComponentType.			
Base	<i>ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable, SwConnector</i>			
Attribute	Type	Mult.	Kind	Note
provider	AbstractProvidedPortPrototype	0..1	iref	Instance of providing port. InstanceRef implemented by: PPortInComposition InstanceRef
requester	AbstractRequiredPortPrototype	0..1	iref	Instance of requiring port. InstanceRef implemented by: RPortInComposition InstanceRef

Table B.12: AssemblySwConnector

Class	AsynchronousServerCallReturnsEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when an asynchronous server call is finished.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , Multilanguage Referrable, RTEEvent , Referrable			
Attribute	Type	Mult.	Kind	Note
eventSource	AsynchronousServerCallResultPoint	0..1	ref	The referenced AsynchronousServerCallResultPoint raises this AsynchronousServerCallReturnsEvent when the asynchronous server call returns.

Table B.13: AsynchronousServerCallReturnsEvent

Class	AutosarDataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Base class for prototypical roles of an AutosarDataType.			
Base	ARObject, AtpFeature, AtpPrototype, DataPrototype , Identifiable , MultilanguageReferrable, Referrable			
Subclasses	ArgumentDataPrototype , ParameterDataPrototype , VariableDataPrototype			
Attribute	Type	Mult.	Kind	Note
type	AutosarDataType	0..1	tref	This represents the corresponding data type. Stereotypes: isOfType

Table B.14: AutosarDataPrototype

Class	AutosarDataType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	Abstract base class for user defined AUTOSAR data types for software.			
Base	ARElement, ARObject, AtpClassifier, AtpType, CollectableElement, Identifiable , Multilanguage Referrable, PackageableElement, Referrable			
Subclasses	AbstractImplementationDataType , ApplicationDataType			
Attribute	Type	Mult.	Kind	Note
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType.

Table B.15: AutosarDataType

Class	BackgroundEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is used to start RunnableEntities that are supposed to be executed in the background.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , Multilanguage Referrable, RTEEvent , Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table B.16: BackgroundEvent

Class	BaseType (abstract)			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This abstract meta-class represents the ability to specify a platform dependent base type.			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	SwBaseType			
Attribute	Type	Mult.	Kind	Note
baseType Definition	BaseTypeDefinition	1	aggr	This is the actual definition of the base type. Tags: xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table B.17: BaseType

Class	BaseTypeDirectDefinition			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	<i>ARObject</i> , <i>BaseTypeDefinition</i>			
Attribute	Type	Mult.	Kind	Note
baseType Encoding	BaseTypeEncodingString	0..1	attr	This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. Tags: xml.sequenceOffset=90
baseTypeSize	PositiveInteger	0..1	attr	Describes the length of the data type specified in the container in bits. Tags: xml.sequenceOffset=70
byteOrder	ByteOrderEnum	0..1	attr	This attribute specifies the byte order of the base type. Tags: xml.sequenceOffset=110
memAlignment	PositiveInteger	0..1	attr	This attribute describes the alignment of the memory object in bits. E.g. "8" specifies, that the object in question is aligned to a byte while "32" specifies that it is aligned four byte. If the value is set to "0" the meaning shall be interpreted as "unspecified". Tags: xml.sequenceOffset=100
native Declaration	NativeDeclarationString	0..1	attr	This attribute describes the declaration of such a base type in the native programming language, primarily in the Programming language C. This can then be used by a code generator to include the necessary declarations into a header file. For example BaseType with shortName: "MyUnsignedInt" native Declaration: "unsigned short" Results in typedef unsigned short MyUnsignedInt; If the attribute is not defined the referring Implementation DataTypes will not be generated as a typedef by RTE.





Class	BaseTypeDirectDefinition			
				<p>△</p> <p>If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseType Size.</p> <p>This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems.</p> <p>Tags:xml.sequenceOffset=120</p>

Table B.18: BaseTypeDirectDefinition

Class	BswExternalTriggerOccurredEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	A BswEvent resulting from a trigger released by another module or cluster.			
Base	ARObject, AbstractEvent, BswEvent, BswScheduleEvent, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
trigger	Trigger	1	ref	The trigger associated with this event. The trigger is external to this module.

Table B.19: BswExternalTriggerOccurredEvent

Class	BswInternalTriggerOccurredEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	A BswEvent, which can happen sporadically. The event is activated by explicit calls from the module to the BSW Scheduler. The main purpose for such an event is to cause a context switch, e.g. from an ISR context into a task context. Activation and switching are handled within the same module or cluster only.			
Base	ARObject, AbstractEvent, BswEvent, BswScheduleEvent, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
eventSource	BswInternalTriggering Point	1	ref	The activation point is the source of this event.

Table B.20: BswInternalTriggerOccurredEvent

Class	BswModeSwitchEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	A BswEvent resulting from a mode switch.			
Base	ARObject, AbstractEvent, BswEvent, BswScheduleEvent, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
activation	ModeActivationKind	1	attr	Kind of activation w.r.t. to the referred mode.
mode (ordered)	ModeDeclaration	0..2	iref	Reference to one or two Modes that initiate the Mode Switch Event. InstanceRef implemented by: ModeInBswModule DescriptionInstanceRef

Table B.21: BswModeSwitchEvent

Class	BswModuleDescription			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswOverview			
Note	Root element for the description of a single BSW module or BSW cluster. In case it describes a BSW module, the short name of this element equals the name of the BSW module. Tags: atp.recommendedPackage=BswModuleDescriptions			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpFeature, AtpStructureElement, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
bswModuleDependency	BswModuleDependency	*	aggr	Describes the dependency to another BSW module. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=bswModuleDependency.shortName, bswModuleDependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
bswModuleDocumentation	SwComponentDocumentation	0..1	aggr	This adds a documentation to the BSW module. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=bswModuleDocumentation, bswModuleDocumentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=6
expectedEntry	BswModuleEntry	*	ref	Indicates an entry which is required by this module. Replacement of outgoingCallback / requiredEntry. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=expectedEntry.bswModuleEntry, expectedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
implementedEntry	BswModuleEntry	*	ref	Specifies an entry provided by this module which can be called by other modules. This includes "main" functions, interrupt routines, and callbacks. Replacement of providedEntry / expectedCallback. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=implementedEntry.bswModuleEntry, implementedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
internalBehavior	BswInternalBehavior	*	aggr	The various BswInternalBehaviors associated with a BswModuleDescription can be distributed over several physical files. Therefore the aggregation is <<atpSplitable>>. Stereotypes: atpSplitable Tags: atp.Splitkey=internalBehavior.shortName xml.sequenceOffset=65
moduleId	PositiveInteger	0..1	attr	Refers to the BSW Module Identifier defined by the AUTOSAR standard. For non-standardized modules, a proprietary identifier can be optionally chosen. Tags: xml.sequenceOffset=5





Class	BswModuleDescription			
providedClientServerEntry	BswModuleClientServerEntry	*	aggr	<p>Specifies that this module provides a client server entry which can be called from another partition or core. This entry is declared locally to this context and will be connected to the requiredClientServerEntry of another or the same module via the configuration of the BSW Scheduler.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=providedClientServerEntry.shortName, providedClientServerEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45</p>
providedData	VariableDataPrototype	*	aggr	<p>Specifies a data prototype provided by this module in order to be read from another partition or core. The providedData is declared locally to this context and will be connected to the requiredData of another or the same module via the configuration of the BSW Scheduler.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=providedData.shortName, providedData.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55</p>
providedModeGroup	ModeDeclarationGroupPrototype	*	aggr	<p>A set of modes which is owned and provided by this module or cluster. It can be connected to the requiredModeGroups of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with modes provided via ports by an associated ServiceSwComponentType, EcuAbstractionSwComponentType or ComplexDeviceDriverSwComponentType.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=providedModeGroup.shortName, providedModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25</p>
releasedTrigger	Trigger	*	aggr	<p>A Trigger released by this module or cluster. It can be connected to the requiredTriggers of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with Triggers provided via ports by an associated ServiceSwComponentType, EcuAbstractionSwComponentType or ComplexDeviceDriverSwComponentType.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=releasedTrigger.shortName, releasedTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=35</p>





Class	BswModuleDescription			
requiredClientServerEntry	BswModuleClientServerEntry	*	aggr	<p>Specifies that this module requires a client server entry which can be implemented on another partition or core. This entry is declared locally to this context and will be connected to the providedClientServerEntry of another or the same module via the configuration of the BSW Scheduler.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=requiredClientServerEntry.shortName, requiredClientServerEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50</p>
requiredData	VariableDataPrototype	*	aggr	<p>Specifies a data prototype required by this module in order to be provided from another partition or core. The required Data is declared locally to this context and will be connected to the providedData of another or the same module via the configuration of the BswScheduler.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=requiredData.shortName, requiredData.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60</p>
requiredModeGroup	ModeDeclarationGroupPrototype	*	aggr	<p>Specifies that this module or cluster depends on a certain mode group. The requiredModeGroup is local to this context and will be connected to the providedModeGroup of another module or cluster via the configuration of the BswScheduler.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=requiredModeGroup.shortName, requiredModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=30</p>
requiredTrigger	Trigger	*	aggr	<p>Specifies that this module or cluster reacts upon an external trigger. This requiredTrigger is declared locally to this context and will be connected to the providedTrigger of another module or cluster via the configuration of the BswScheduler.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=requiredTrigger.shortName, requiredTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40</p>

Table B.22: BswModuleDescription

Class	<i>BswModuleEntity</i> (abstract)			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Specifies the smallest code fragment which can be described for a BSW module or cluster within AUTOSAR.			
Base	<i>AObject</i> , <i>ExecutableEntity</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	BswCalledEntity, BswInterruptEntity, BswSchedulableEntity			
Attribute	Type	Mult.	Kind	Note





Class	BswModuleEntity (abstract)			
accessedMode Group	ModeDeclarationGroup Prototype	*	ref	A mode group which is accessed via API call by this entity. It shall be a ModeDeclarationGroupPrototype required by this module or cluster. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
activationPoint	BswInternalTriggering Point	*	ref	Activation point used by the module entity to activate one or more internal triggers. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
callPoint	BswModuleCallPoint	*	aggr	A call point used in the code of this entity. The variability of this association is especially targeted at debug scenarios: It is possible to have one variant calling into the AUTOSAR debug module and another one which doesn't. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
dataReceive Point	BswVariableAccess	*	aggr	The data is received via the BSW Scheduler. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
dataSendPoint	BswVariableAccess	*	aggr	The data is sent via the BSW Scheduler. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
implemented Entry	BswModuleEntry	1	ref	The entry which is implemented by this module entity.
issuedTrigger	Trigger	*	ref	A trigger issued by this entity via BSW Scheduler API call. It shall be a BswTrigger released (i.e. owned) by this module or cluster. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
managedMode Group	ModeDeclarationGroup Prototype	*	ref	A mode group which is managed by this entity. It shall be a ModeDeclarationGroupPrototype provided by this module or cluster. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
schedulerName Prefix	BswSchedulerName Prefix	0..1	ref	A prefix to be used in generated names for the Bsw ModuleScheduler in the context of this BswModuleEntity, for example entry point prototypes, macros for dealing with exclusive areas, header file names. Details are defined in the SWS RTE. The prefix supersedes default rules for the prefix of those names.

Table B.23: BswModuleEntity

Class	BswSchedulableEntity			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	BSW module entity, which is designed for control by the BSW Scheduler. It may for example implement a so-called "main" function.			
Base	AObject , BswModuleEntity , ExecutableEntity , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note





Class	BswSchedulableEntity			
-	-	-	-	-

Table B.24: BswSchedulableEntity

Class	BswVariableAccess			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	The presence of a BswVariableAccess implies that a BswModuleEntity needs access to a VariableData Prototype via the BSW Scheduler. The kind of access is specified by the role in which the class is used.			
Base	ARObject, <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
accessed Variable	VariableDataPrototype	1	ref	The data accessed via the BSW Scheduler.
context Limitation	BswDistinguished Partition	*	ref	The existence of this reference indicates that the variable is received resp. sent only in the context of the referred BswDistinguishedPartitions.

Table B.25: BswVariableAccess

Class	ClientServerApplicationErrorMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents the ability to map ApplicationErrors onto each other.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
firstApplication Error	ApplicationError	0..1	ref	This represents the first ApplicationError in the context of the ClientServerApplicationErrorMapping.
second ApplicationError	ApplicationError	0..1	ref	This represents the second ApplicationError in the context of the ClientServerApplicationErrorMapping.

Table B.26: ClientServerApplicationErrorMapping

Class	ClientServerInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A client/server interface declares a number of operations that can be invoked on a server by a client. Tags: atp.recommendedPackage=PortInterfaces			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
operation	ClientServerOperation	*	aggr	ClientServerOperation(s) of this ClientServerInterface. Stereotypes: atpVariation Tags: vh.latestBindingTime=blueprintDerivationTime
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface.

Table B.27: ClientServerInterface

Class	ClientServerOperation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	An operation declared within the scope of a client/server interface.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
argument (ordered)	ArgumentDataPrototype	*	aggr	An argument of this ClientServerOperation Stereotypes: atpVariation Tags: vh.latestBindingTime=blueprintDerivationTime
diagArgIntegrity	Boolean	0..1	attr	This attribute shall only be used in the implementation of diagnostic routines to support the case where input and output arguments are allocated in a shared buffer and might unintentionally overwrite input arguments by tentative write operations to output arguments. This situation can happen during sliced execution or while output parameters are arrays (call by reference). The value true means that the ClientServerOperation is aware of the usage of a shared buffer and takes precautions to avoid unintentional overwrite of input arguments. If the attribute does not exist or is set to false the Client ServerOperation does not have to consider the usage of a shared buffer.
possibleError	ApplicationError	*	ref	Possible errors that may be raised by the referring operation.

Table B.28: ClientServerOperation

Class	ClientServerOperationMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines the mapping of two particular ClientServerOperations in context of two different ClientServer Interfaces.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
argument Mapping	DataPrototypeMapping	*	aggr	Defines the mapping of two particular ArgumentData Prototypes with unequal names or unequal semantic (resolution or range) in context of Operations.
firstOperation	ClientServerOperation	0..1	ref	First to-be-mapped ClientServerOperation of a Client ServerInterface.
firstToSecond Data Transformation	DataTransformation	0..1	ref	This reference indicates that a DataTransformation is intended in the context of the ClientServerOperation Mapping.
second Operation	ClientServerOperation	0..1	ref	Second to-be-mapped ClientServerOperation of a Client ServerInterface.

Table B.29: ClientServerOperationMapping

Class	CompositionSwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	<p>A CompositionSwComponentType aggregates SwComponentPrototypes (that in turn are typed by SwComponentTypes) as well as SwConnectors for primarily connecting SwComponentPrototypes among each others and towards the surface of the CompositionSwComponentType. By this means, hierarchical structures of software-components can be created.</p> <p>Tags:atp.recommendedPackage=SwComponentTypes</p>			
Base	<p>ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, SwComponentType</p>			
Attribute	Type	Mult.	Kind	Note
component	SwComponentPrototype	*	aggr	<p>The instantiated components that are part of this composition. The aggregation of SwComponentPrototype is subject to variability with the purpose to support the conditional existence of a SwComponentPrototype. Please be aware: if the conditional existence of SwComponentPrototypes is resolved post-build the deselected SwComponentPrototypes are still contained in the ECUs build but the instances are inactive in that they are not scheduled by the RTE.</p> <p>The aggregation is marked as atpSplitable in order to allow the addition of service components to the ECU extract during the ECU integration.</p> <p>The use case for having 0 components owned by the CompositionSwComponentType could be to deliver an empty CompositionSwComponentType to e.g. a supplier for filling the internal structure.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=component.shortName, component.variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
connector	SwConnector	*	aggr	<p>SwConnectors have the principal ability to establish a connection among PortPrototypes. They can have many roles in the context of a CompositionSwComponentType. Details are refined by subclasses.</p> <p>The aggregation of SwConnectors is subject to variability with the purpose to support variant data flow.</p> <p>The aggregation is marked as atpSplitable in order to allow the extension of the ECU extract with AssemblySwConnectors between ApplicationSwComponentTypes and ServiceSwComponentTypes during the ECU integration.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
constantValueMapping	ConstantSpecificationMappingSet	*	ref	<p>Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortComSpec.</p> <p>Stereotypes: atpSplitable Tags:atp.Splitkey=constantValueMapping</p>
dataTypeMapping	DataTypeMappingSet	*	ref	<p>Reference to the DataTypeMapping to be applied for the used ApplicationDataTypes in PortInterfaces.</p> <p>Background: when developing subsystems it may happen that ApplicationDataTypes are used on the surface of</p>





Class	CompositionSwComponentType			
				<p>CompositionSwComponentTypes. In this case it would be reasonable to be able to also provide the intended mapping to the ImplementationDataTypes. However, this mapping shall be informal and not technically binding for the implementors mainly because the RTE generator is not concerned about the CompositionSwComponentTypes.</p> <p>Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility.</p> <p>Stereotypes: atpSplitable Tags:atp.Splitkey=dataTypeMapping</p>
instantiationRTEEventProps	InstantiationRTEEventProps	*	aggr	<p>This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=instantiationRTEEventProps.shortLabel, instantiationRTEEventProps.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime</p>

Table B.30: CompositionSwComponentType

Class	Compu			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to express one particular computation.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
compuContent	CompuContent	0..1	aggr	<p>This specifies the details of the computation.</p> <p>Tags: xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false</p>
compuDefaultValue	CompuConst	0..1	aggr	<p>This property can be used to specify an output value for a conversion formula, if the value to be converted lies outside the plausibility limit. Although this is possible for all conversion formulae, it is especially valid for variables with tabular conversion formulae.</p> <p>Tags:xml.sequenceOffset=70</p>

Table B.31: Compu

Class	CompuConstFormulaContent			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the fact that the constant value of the computation method is represented by a variation point. This difference is due to compatibility with ASAM HDO.			
Base	ARObject, CompuConstContent			





Class		CompuConstFormulaContent		
Attribute	Type	Mult.	Kind	Note
vf	Numerical	1	attr	<p>Value calculated via a system constant. This element is included in every case where parameters should be generated from numerical values during compile time (not runtime!).</p> <p>Thus for example, the influence of the cylinder number on conversion formulae can be introduced in a repeatable manner.</p> <p>Stereotypes: atpVariation Tags: vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=30</p>

Table B.32: CompuConstFormulaContent

Class		CompuConstTextContent		
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the textual content of a scale.			
Base	ARObject, CompuConstContent			
Attribute	Type	Mult.	Kind	Note
vt	VerbatimString	0..1	attr	This represents a textual constant in the computation method.

Table B.33: CompuConstTextContent

Class		CompuMethod		
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	<p>This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.</p> <p>Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.</p> <p>Tags:atp.recommendedPackage=CompuMethods</p>			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
compuInternalToPhys	Compu	0..1	aggr	<p>This specifies the computation from internal values to physical values.</p> <p>Tags:xml.sequenceOffset=80</p>
compuPhysToInternal	Compu	0..1	aggr	<p>This represents the computation from physical values to the internal values.</p> <p>Tags:xml.sequenceOffset=90</p>
displayFormat	DisplayFormatString	0..1	attr	<p>This property specifies, how the physical value shall be displayed e.g. in documents or measurement and calibration tools.</p> <p>Tags:xml.sequenceOffset=20</p>





Class	CompuMethod			
unit	Unit	0..1	ref	This is the physical unit of the Physical values for which the CompuMethod applies. Tags: xml.sequenceOffset=30

Table B.34: CompuMethod

Class	CompuNominatorDenominator			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This class represents the ability to express a polynomial either as Nominator or as Denominator.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
v (ordered)	Numerical	*	attr	this is the list of polynomial factors. Note that the first v represents the power=0. The polynomial is $v[0] * x^0 + v[1] * x^1 \dots$ Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table B.35: CompuNominatorDenominator

Class	CompuRationalCoeffs			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to express a rational function by specifying the coefficients of nominator and denominator.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
compu Denominator	CompuNominatorDenominator	0..1	aggr	This is the denominator of the expression. Tags: xml.sequenceOffset=30
compu Numerator	CompuNominatorDenominator	0..1	aggr	This is the numerator of the rational expression. Tags: xml.sequenceOffset=20

Table B.36: CompuRationalCoeffs

Class	CompuScale			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to specify one segment of a segmented computation method.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note





Class	CompuScale			
compuInverseValue	CompuConst	0..1	aggr	This is the inverse value of the constraint. This supports the case that the scale is not reversible per se. Tags: xml.sequenceOffset=60
compuScaleContents	CompuScaleContents	0..1	aggr	This represents the computation details of the scale. Tags: xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=70 xml.typeElement=false xml.typeWrapperElement=false
desc	MultiLanguageOverviewParagraph	0..1	aggr	<desc> represents a general but brief description of the object in question. Tags: xml.sequenceOffset=30
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the scale. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
mask	PositiveInteger	0..1	attr	In difference to all the other computational methods every COMPU-SCALE will be applied including the bit MASK. Therefore it is allowed for this type of COMPU-METHOD, that COMPU-SCALES overlap. To calculate the string reverse to a value, the string has to be split and the according value for each substring has to be summed up. The sum is finally transmitted. The processing has to be done in order of the COMPU-SCALE elements. Tags: xml.sequenceOffset=35
shortLabel	Identifier	0..1	attr	This element specifies a short name for the particular scale. The name can for example be used to derive a programming language identifier. Tags: xml.sequenceOffset=20
symbol	CIdentifier	0..1	attr	The symbol, if provided, is used by code generators to get a C identifier for the CompuScale. The name will be used as is for the code generation, therefore it needs to be unique within the generation context. Tags: xml.sequenceOffset=25
upperLimit	Limit	0..1	attr	This specifies the upper limit of a of the scale. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=50

Table B.37: CompuScale

Class	CompuScaleConstantContents			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the fact that a particular scale of the computation method is constant.			
Base	ARObject, CompuScaleContents			
Attribute	Type	Mult.	Kind	Note





Class		CompuScaleConstantContents		
compuConst	CompuConst	0..1	aggr	<p>This represents the fact that the scale is a constant. The use case is mainly a non interpolated scale. It is a simplification of the fact that a constant scale can also be expressed as rational function of order 0.</p> <p>Tags:xml.sequenceOffset=90</p>

Table B.38: CompuScaleConstantContents

Class		CompuScales		
Package		M2::MSR::AsamHdo::ComputationMethod		
Note		This meta-class represents the ability to stepwise express a computation method.		
Base		ARObject, CompuContent		
Attribute	Type	Mult.	Kind	Note
compuScale (ordered)	CompuScale	*	aggr	<p>This represents one scale within the compu method. Note that it contains a Variationpoint in order to support blueprints of enumerations.</p> <p>Stereotypes: atpVariation</p> <p>Tags: vh.latestBindingTime=blueprintDerivationTime xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false</p>

Table B.39: CompuScales

Class		CpSoftwareCluster		
Package		M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster		
Note		<p>This meta class provides the ability to define a CP Software Cluster. Each CP Software Cluster can be integrated and build individually. It defines the sub-set of hierarchical tree(s) of Software Components belonging to this CP Software Cluster. Resources required or provided by this CP Software Cluster are given in the according mappings.</p> <p>Tags: atp.Status=draft atp.recommendedPackage=CpSoftwareClusters</p>		
Base		ARElement, ARObject, CollectableElement, Identifiable , MultilanguageReferrable , PackageableElement , Referrable		
Attribute	Type	Mult.	Kind	Note
swComponent Assignment	SwComponent PrototypeAssignment	*	aggr	<p>This is the collection of SwComponentPrototype Assignments</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=swComponentAssignment, swComponent Assignment.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild</p>





Class	CpSoftwareCluster			
swComposition	CompositionSw ComponentType	*	ref	<p>Software Components in the context of a CompositionSw ComponentType belonging to this CP Software Cluster. This reference can be used to describe the belonging SWCs when the CP Software Cluster is described out of the context of a System, e.g. reusable CP Software Cluster.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=swComposition.compositionSwComponentType, swComposition.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=systemDesignTime</p>

Table B.40: CpSoftwareCluster

Class	CpSoftwareClusterBinaryManifestDescriptor			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	<p>This meta-class has the ability to act as a hub for all information related to the binary manifest of a given CP software cluster. The manifest is subject to integrator work and therefore not a part of the definition of the CP software cluster itself.</p> <p>Tags: atp.Status=draft atp.recommendedPackage=CpSoftwareClusterBinaryManifestDescriptors</p>			
Base	<i>ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
cpSoftwareCluster	CpSoftwareCluster	0..1	ref	<p>This reference identifies the CpSoftwareCluster to which the enclosing CpSoftwareClusterBinaryManifestDescriptor belongs,</p> <p>The CpSoftwareClusterBinaryManifestDescriptor is defined in an integration phase while the referenced CpSoftwareCluster represents a design element. Therefore, it makes sense to use a reference rather than an aggregation in the relation of the two meta-classes.</p> <p>Tags:atp.Status=draft</p>
metaDataField	BinaryManifestMeta DataField	*	aggr	<p>This aggregation identifies the collection of meta-data contained in the enclosing binary manifest.</p> <p>Tags:atp.Status=draft</p>
provideResource	BinaryManifestProvide Resource	*	aggr	<p>This aggregation represents the collection of provided resources in the enclosing binary manifest.</p> <p>Tags:atp.Status=draft</p>
requireResource	BinaryManifestRequire Resource	*	aggr	<p>This aggregation represents the collection of required resources in the enclosing binary manifest.</p> <p>Tags:atp.Status=draft</p>
resourceDefinition	BinaryManifest ResourceDefinition	*	aggr	<p>This aggregation represents the collection of binary manifest resource definitions that belong to the enclosing CpSoftwareClusterBinaryManifestDescriptor.</p> <p>Tags:atp.Status=draft</p>





Class		CpSoftwareClusterBinaryManifestDescriptor		
softwareClusterId	PositiveInteger	0..1	attr	This attribute represents the value of the id of the corresponding CP software cluster. This id is only assigned by an integrator and can therefore not be part of the description of the CP software cluster itself. Tags: atp.Status=draft

Table B.41: CpSoftwareClusterBinaryManifestDescriptor

Class		CpSoftwareClusterCommunicationResource		
Package		M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster		
Note		Represents a single resource required or provided by a CP Software Cluster which relates to the port based communication on VFB level. Tags: atp.Status=draft		
Base		AObject, CpSoftwareClusterResource, Identifiable, MultilanguageReferrable, Referrable		
Attribute	Type	Mult.	Kind	Note
communicationResourceProps	CpSoftwareClusterCommunicationResourceProps	0..1	aggr	This aggregation supports the further qualification of the enclosing CpSoftwareClusterCommunicationResource by means of additional attributes depending on the nature of the CpSoftwareClusterCommunicationResource.

Table B.42: CpSoftwareClusterCommunicationResource

Class		CpSoftwareClusterResource (abstract)		
Package		M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster		
Note		Represents a single resource required or provided by a CP Software Cluster. Tags: atp.Status=draft atp.recommendedPackage=Resources		
Base		AObject, Identifiable, MultilanguageReferrable, Referrable		
Subclasses		CpSoftwareClusterCommunicationResource, CpSoftwareClusterServiceResource		
Attribute	Type	Mult.	Kind	Note
dependentResource	RoleBasedResourceDependency	*	aggr	Link to a resource which depends on this resource to implement them. Tags: atp.Status=draft
globalResourceId	PositiveInteger	0..1	attr	A unique identifiers per resource used for the connection process. The identifier is required to be unique in the scope of a single machine. If software clusters are designed to be reused on multiple machines the uniqueness requirements applies for all the intended machines. Tags: atp.Status=draft
isMandatory	Boolean	0..1	attr	This attribute indicates, that the resource is mandatory to operate the Software Cluster. If the resource is not provided on the machine the connection process of any Software Cluster requiring this resource gets aborted. Tags: atp.Status=draft

Table B.43: CpSoftwareClusterResource

Class	CpSoftwareClusterResourcePool			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	Represents the pool of resources which can be provided or required by CP Software Clusters. Tags: atp.Status=draft atp.recommendedPackage=CpSoftwareClusterResourcePools			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
ecuScope	EcuInstance	*	ref	This reference identifies the EcuInstance in which the resource pool is defined. Stereotypes: atpSplittable Tags: atp.Splitkey=ecuScope atp.Status=draft
resource	CpSoftwareClusterResource	*	aggr	This aggregation represents the collection of resources in the enclosing resource pool. Stereotypes: atpSplittable Tags: atp.Splitkey=resource.shortName atp.Status=draft

Table B.44: CpSoftwareClusterResourcePool

Class	CpSoftwareClusterServiceResource			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the BSW. Tags: atp.Status=draft			
Base	ARObject, CpSoftwareClusterResource, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
resourceNeeds	EcucContainerValue	*	ref	Reference(s) to one or multiple EcucContainerValue(s) qualifying the characteristics of the resource. Tags: atp.Status=draft

Table B.45: CpSoftwareClusterServiceResource

Class	DataComProps			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the port based communication on VFB level. Tags: atp.Status=draft			
Base	ARObject, CpSoftwareClusterCommunicationResourceProps			
Attribute	Type	Mult.	Kind	Note
sendIndication	SendIndicationEnum	0..1	attr	Send indication behavior for last-is-the best data communication.

Table B.46: DataComProps

Class	DataConstr			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to specify constraints on data. Tags: atp.recommendedPackage=DataConstrs			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable , Multilanguage Referrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
dataConstrRule	DataConstrRule	*	aggr	This is one particular rule within the data constraints. Tags: xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table B.47: DataConstr

Class	DataConstrRule			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to express one specific data constraint rule.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
constrLevel	Integer	0..1	attr	This attribute describes the category of a constraint. One of its functions is in the area of constraint violation, where it can be used from a certain level, to produce error messages. The lower the level, the more stringent the check. Used to distinguish hard or soft limits. Tags: xml.sequenceOffset=20
internalConstrs	InternalConstrs	0..1	aggr	Describes the limitations applicable on the internal domain (as opposed to the physical domain). Tags: xml.sequenceOffset=40
physConstrs	PhysConstrs	0..1	aggr	Describes the limitations applicable on the physical domain (as opposed to the internal domain). Tags: xml.sequenceOffset=30

Table B.48: DataConstrRule

Class	DataMapping (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of port elements (data elements and parameters) to frames and signals.			
Base	ARObject			
Subclasses	ClientServerToSignalMapping, SenderReceiverCompositeElementToSignalMapping, SenderReceiverToSignalGroupMapping, SenderReceiverToSignalMapping, TriggerToSignalMapping			
Attribute	Type	Mult.	Kind	Note





Class	DataMapping (abstract)			
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the data mapping.

Table B.49: DataMapping

Class	DataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Base class for prototypical roles of any data type.			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable , MultilanguageReferrable , Referrable			
Subclasses	ApplicationCompositeElementDataPrototype , AutosarDataPrototype			
Attribute	Type	Mult.	Kind	Note
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level.

Table B.50: DataPrototype

Class	DataReceivedEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the referenced data element is received.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , RTEEvent , Referrable			
Attribute	Type	Mult.	Kind	Note
data	VariableDataPrototype	0..1	iref	The referenced VariableDataPrototype raises this Data ReceivedEvent when the data has been received. InstanceRef implemented by: RVVariableInAtomicSwc InstanceRef

Table B.51: DataReceivedEvent

Class	DiagnosticEventInfoNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This meta-class represents the needs of a software-component interested to get information regarding specific DTCs.			
Base	ARObject, DiagnosticCapabilityElement , Identifiable , MultilanguageReferrable , Referrable , ServiceNeeds			
Attribute	Type	Mult.	Kind	Note
obdDtcNumber	PositiveInteger	0..1	attr	This represents a reasonable Diagnostic Trouble Code. This allows to predefine the Diagnostic Trouble Code, e.g. if the function developer has received a particular requirement from the OEM or from a standardization body. This attribute applies for the OBD diagnostics use case.





Class		DiagnosticEventInfoNeeds		
udsDtcNumber	PositiveInteger	0..1	attr	<p>This represents a reasonable Diagnostic Trouble Code.</p> <p>This allows to predefine the Diagnostic Trouble Code, e.g. if the function developer has received a particular requirement from the OEM or from a standardization body.</p> <p>This attribute applies for the UDS diagnostics use case.</p>

Table B.52: DiagnosticEventInfoNeeds

Class		DiagnosticEventNeeds		
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	<p>Specifies the abstract needs on the configuration of the Diagnostic Event Manager for one diagnostic event. Its shortName can be regarded as a symbol identifying the diagnostic event from the viewpoint of the component or module which owns this element.</p> <p>In case the diagnostic event specifies a production error, the shortName shall be the name of the production error.</p>			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	Type	Mult.	Kind	Note
deferringFid	FunctionInhibitionNeeds	*	ref	This reference contains the link to a function identifier within the FiM which is used by the monitor before delivering a result.
diagEventDebounceAlgorithm	DiagEventDebounceAlgorithm	0..1	aggr	Specifies the abstract need on the Debounce Algorithm applied by the Diagnostic Event Manager.
prestoredFreezeFrameStoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestoredFreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm).
usesMonitorData	Boolean	0..1	attr	This attribute defines whether additional monitor data shall be added to the reporting of events.

Table B.53: DiagnosticEventNeeds

Class		EcuInstance		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	<p>ECUInstances are used to define the ECUs used in the topology. The type of the ECU is defined by a reference to an ECU specified with the ECU resource description.</p> <p>Tags:atp.recommendedPackage=EcuInstances</p>			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note





Class	EcuInstance			
associatedComIPduGroup	ISignalIPduGroup	*	ref	<p>With this reference it is possible to identify which ISignalIPduGroups are applicable for which Communication Connector/ ECU.</p> <p>Only top level ISignalIPduGroups shall be referenced by an EcuInstance. If an ISignalIPduGroup contains other ISignalIPduGroups than these contained ISignalIPdu Groups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an Ecu Instance via the top level ISignalIPduGroup.</p>
associatedConsumedProvidedServiceInstanceGroup	ConsumedProvidedServiceInstanceGroup	*	ref	<p>With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
associatedPdurIPduGroup	PdurIPduGroup	*	ref	<p>With this reference it is possible to identify which PdurIPdu Groups are applicable for which Communication Connector/ ECU.</p>
clientIdRange	ClientIdRange	0..1	aggr	<p>Restriction of the Client Identifier for this Ecu to an allowed range of numerical values. The Client Identifier of the transaction handle is generated by the client RTE for inter-Ecu Client/Server communication.</p>
comConfigurationGwTimeBase	TimeValue	0..1	attr	<p>The period between successive calls to Com_Main FunctionRouteSignals of the AUTOSAR COM module in seconds.</p>
comConfigurationRxTimeBase	TimeValue	0..1	attr	<p>The period between successive calls to Com_Main FunctionRx of the AUTOSAR COM module in seconds.</p>
comConfigurationTxTimeBase	TimeValue	0..1	attr	<p>The period between successive calls to Com_Main FunctionTx of the AUTOSAR COM module in seconds.</p>
comEnableMDTForCyclicTransmission	Boolean	0..1	attr	<p>Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclic Timing assigned or eventControlledTiming with numberOfRepetitions > 0).</p>
commController	CommunicationController	1..*	aggr	<p>CommunicationControllers of the ECU.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
connector	CommunicationConnector	*	aggr	<p>All channels controlled by a single controller.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
dltConfig	DltConfig	0..1	aggr	<p>Describes the Dlt configuration on this EcuInstance.</p>
dolpConfig	DolpConfig	0..1	aggr	<p>Dolp configuration on this EcuInstance.</p> <p>Tags:atp.Status=draft</p>
ecuTaskProxy	OsTaskProxy	*	ref	<p>Reference to OsTaskProxies assigned to the Ecu Instance.</p> <p>Stereotypes: atpSplittable Tags:atp.Splitkey=ecuTaskProxy</p>
ethSwitchPortGroupDerivation	Boolean	0..1	attr	<p>Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done.</p>
partition	EcuPartition	*	aggr	<p>Optional definition of Partitions within an Ecu.</p>
pncPrepareSleepTimer	TimeValue	0..1	attr	<p>Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.</p>





Class	EcuInstance			
pnc Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs. This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping.
pnResetTime	TimeValue	0..1	attr	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.
sleepMode Supported	Boolean	1	attr	Specifies whether the ECU instance may be put to a "low power mode" <ul style="list-style-type: none"> • true: sleep mode is supported • false: sleep mode is not supported Note: This flag may only be set to "true" if the feature is supported by both hardware and basic software.
tcplpcmpProps	EthTcplpcmpProps	0..1	ref	EcuInstance specific ICMP (Internet Control Message Protocol) attributes
tcplpProps	EthTcplpProps	0..1	ref	EcuInstance specific Tcplp Stack attributes.
v2xSupported	V2xSupportEnum	0..1	attr	This attribute is used to control the existence of the V2X stack on the given EcuInstance.
wakeUpOver BusSupported	Boolean	1	attr	Driver support for wakeup over Bus.

Table B.54: EcuInstance

Class	EcucContainerDef (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Base class used to gather common attributes of configuration container definitions.			
Base	ARObject, AtpDefinition, EcucDefinitionElement, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	EcucChoiceContainerDef, EcucParamConfContainerDef			
Attribute	Type	Mult.	Kind	Note
destinationUri	EcucDestinationUriDef	*	ref	Several destinationUris can be defined for an Ecuc ContainerDef. With such destinationUris an Ecuc ContainerDef is applicable for several EcucUriReference Defs. Stereotypes: atpUriDef
multiplicity ConfigClass	EcucMultiplicity ConfigurationClass	*	aggr	Specifies which MultiplicityConfigurationClass this container is available for which ConfigurationVariant. This aggregation is optional if the surrounding EcucModuleDef has the Category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModule Def is set to VENDOR_SPECIFIC_MODULE_DEFINITION and if the upperMultiplicity is greater than the lowerMultiplicity then this aggregation is mandatory. Tags: xml.name Plural=MULTIPLICITY-CONFIG-CLASSES
postBuildVariant Multiplicity	Boolean	0..1	attr	Indicates if a container may have different number of instances in different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
requiresIndex	Boolean	0..1	attr	Used to define whether the value element for this definition shall be provided with an index.

Table B.55: EcucContainerDef

Class	EcucContainerValue			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Represents a Container definition in the ECU Configuration Description.			
Base	ARObject, EcucIndexableValue, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
definition	EcucContainerDef	0..1	ref	Reference to the definition of this Container in the ECU Configuration Parameter Definition. Stereotypes: atpIdentityContributor Tags: xml.sequenceOffset=-10
parameterValue	EcucParameterValue	*	aggr	Aggregates all ECU Configuration Values within this Container. atpVariation: [RS_ECUC_00079] Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=parameterValue.definition, parameterValue.variationPoint.shortLabel vh.latestBindingTime=postBuild
referenceValue	EcucAbstractReferenceValue	*	aggr	Aggregates all References with this container. atpVariation: [RS_ECUC_00079] Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=referenceValue.definition, referenceValue.variationPoint.shortLabel vh.latestBindingTime=postBuild
subContainer	EcucContainerValue	*	aggr	Aggregates all sub-containers within this container. atpVariation: [RS_ECUC_00078] Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=subContainer.shortName, subContainer.definition, subContainer.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table B.56: EcucContainerValue

Class	EcucEnumerationParamDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Configuration parameter type for Enumeration.			
Base	ARObject, AtpDefinition, EcucCommonAttributes, EcucDefinitionElement, EcucParameterDef, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
defaultValue	Identifier	0..1	attr	Default value of the enumeration configuration parameter. This string needs to be one of the literals specified for this enumeration.





Class	EcucEnumerationParamDef			
literal	EcucEnumerationLiteral Def	*	aggr	Aggregation on the literals used to define this enumeration parameter. This aggregation is optional if the surrounding EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_MODULE_DEFINITION then this aggregation is mandatory. Stereotypes: atpSplittable Tags: atp.Splitkey=literal.shortName

Table B.57: EcucEnumerationParamDef

Class	EcucForeignReferenceDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specify a reference to an XML description of an entity described in another AUTOSAR template.			
Base	ARObject, AtpDefinition, EcucAbstractExternalReferenceDef, EcucAbstractReferenceDef, EcucCommonAttributes, EcucDefinitionElement, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
destinationType	String	0..1	attr	The type in the AUTOSAR Metamodel to which instance this reference is allowed to point to.

Table B.58: EcucForeignReferenceDef

Class	EcucModuleConfigurationValues			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	<p>Head of the configuration of one Module. A Module can be a BSW module as well as the RTE and ECU Infrastructure.</p> <p>As part of the BSW module description, the EcucModuleConfigurationValues element has two different roles:</p> <p>The recommendedConfiguration contains parameter values recommended by the BSW module vendor.</p> <p>The preconfiguredConfiguration contains values for those parameters which are fixed by the implementation and cannot be changed.</p> <p>These two EcucModuleConfigurationValues are used when the base EcucModuleConfigurationValues (as part of the base ECU configuration) is created to fill parameters with initial values.</p> <p>Tags:atp.recommendedPackage=EcucModuleConfigurationValues</p>			
Base	ARElement, ARObject, CollectableElement, Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
container	EcucContainerValue	*	aggr	Aggregates all containers that belong to this module configuration. atpVariation: [RS_ECUC_00078] Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=container.shortName, container.definition, container.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=10





Class		EcucModuleConfigurationValues		
definition	EcucModuleDef	0..1	ref	Reference to the definition of this EcucModule ConfigurationValues element. Typically, this is a vendor specific module configuration. Stereotypes: atpIdentityContributor Tags: xml.sequenceOffset=-10
ecucDefEdition	RevisionLabelString	0..1	attr	This is the version info of the ModuleDef ECUC Parameter definition to which this values conform to / are based on. For the Definition of ModuleDef ECUC Parameters the AdminData shall be used to express the semantic changes. The compatibility rules between the definition and value revision labels is up to the module's vendor.
implementation ConfigVariant	EcucConfiguration VariantEnum	0..1	attr	Specifies the kind of deliverable this EcucModule ConfigurationValues element provides. If this element is not used in a particular role (e.g. preconfigured Configuration or recommendedConfiguration) then the value shall be one of VariantPreCompile, VariantLink Time, VariantPostBuild.
module Description	BswImplementation	0..1	ref	Referencing the BSW module description, which this EcucModuleConfigurationValues element is configuring. This is optional because the EcucModuleConfiguration Values element is also used to configure the ECU infrastructure (memory map) or Application SW-Cs. However in case the EcucModuleConfigurationValues are used to configure the module, the reference is mandatory in order to fetch module specific "common" published information.
postBuildVariant Used	Boolean	0..1	attr	Indicates whether a module implementation has or plans to have (i.e., introduced at link or post-build time) new post-build variation points. TRUE means yes, FALSE means no. If the attribute is not defined, FALSE semantics shall be assumed.

Table B.59: EcucModuleConfigurationValues

Class		EcucModuleDef		
Package		M2::AUTOSARTemplates::ECUCParameterDefTemplate		
Note		Used as the top-level element for configuration definition for Software Modules, including BSW and RTE as well as ECU Infrastructure. Tags: atp.recommendedPackage=EcucModuleDefs		
Base		<i>ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpDefinition, CollectableElement, EcucDefinitionElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>		
Attribute	Type	Mult.	Kind	Note
apiServicePrefix	CIdentifier	0..1	attr	For CDD modules this attribute holds the apiService Prefix. The shortName of the module definition of a Complex Driver is always "Cdd". Therefore for CDD modules the module apiServicePrefix is described with this attribute.
container	EcucContainerDef	*	aggr	Aggregates the top-level container definitions of this specific module definition. Stereotypes: atpSplitable Tags: atp.Splitkey=container.shortName xml.sequenceOffset=11





Class	EcucModuleDef			
postBuildVariantSupport	Boolean	0..1	attr	Indicates if a module supports different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
refinedModuleDef	EcucModuleDef	0..1	ref	Optional reference from the Vendor Specific Module Definition to the Standardized Module Definition it refines. In case this EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION this reference shall not be provided. In case this EcucModuleDef has the category VENDOR_SPECIFIC_MODULE_DEFINITION this reference is mandatory. Stereotypes: atpUriDef
supportedConfigVariant	EcucConfigurationVariantEnum	*	attr	Specifies which ConfigurationVariants are supported by this software module. This attribute is optional if the EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_MODULE_DEFINITION then this attribute is mandatory.

Table B.60: EcucModuleDef

Class	EcucParamConfContainerDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Used to define configuration containers that can hierarchically contain other containers and/or parameter definitions.			
Base	<i>ARObject, AtpDefinition, EcucContainerDef, EcucDefinitionElement, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
parameter	EcucParameterDef	*	aggr	The parameters defined within the EcucParamConfContainerDef. Stereotypes: atpSplitable Tags: atp.Splitkey=parameter.shortName
reference	EcucAbstractReferenceDef	*	aggr	The references defined within the EcucParamConfContainerDef. Stereotypes: atpSplitable Tags: atp.Splitkey=reference.shortName
subContainer	EcucContainerDef	*	aggr	The containers defined within the EcucParamConfContainerDef. Stereotypes: atpSplitable Tags: atp.Splitkey=subContainer.shortName

Table B.61: EcucParamConfContainerDef

Class	<i>ExecutableEntity</i> (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Abstraction of executable code.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Subclasses	<i>BswModuleEntity, RunnableEntity</i>			
Attribute	Type	Mult.	Kind	Note





Class	ExecutableEntity (abstract)			
activationReason	ExecutableEntity ActivationReason	*	aggr	If the ExecutableEntity provides at least one activationReason element the RTE resp. BSW Scheduler shall provide means to read the activation vector of this executable entity execution. If no activationReason element is provided the feature of being able to determine the activating RTEEvent is disabled for this ExecutableEntity.
canEnter	ExclusiveArea	*	ref	This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
canEnterExclusiveArea	ExclusiveArea	*	ref	This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls. Tags: atp.Status=obsolete
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	ref	This represents the set of ExclusiveAreaNestingOrders recognized by this ExecutableEntity.
minimumStartInterval	TimeValue	0..1	attr	Specifies the time in seconds by which two consecutive starts of an ExecutableEntity are guaranteed to be separated.
reentrancyLevel	ReentrancyLevelEnum	0..1	attr	The reentrancy level of this ExecutableEntity. See the documentation of the enumeration type ReentrancyLevelEnum for details. Please note that nonReentrant interfaces can have also reentrant or multicoreReentrant implementations, and reentrant interfaces can also have multicoreReentrant implementations.
runsInside	ExclusiveArea	*	ref	The executable entity runs completely inside the referenced exclusive area. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
runsInsideExclusiveArea	ExclusiveArea	*	ref	The executable entity runs completely inside the referenced exclusive area. Tags: atp.Status=obsolete
swAddrMethod	SwAddrMethod	0..1	ref	Addressing method related to this code entity. Via an association to the same SwAddrMethod, it can be specified that several code entities (even of different modules or components) shall be located in the same memory without already specifying the memory section itself.

Table B.62: ExecutableEntity

Class	ExternalTriggerOccurredEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the referenced Trigger has occurred.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , RTEEvent , Referrable			
Attribute	Type	Mult.	Kind	Note





Class	ExternalTriggerOccurredEvent			
trigger	Trigger	0..1	iref	The referenced Trigger raises this ExternalTrigger OccurredEvent. InstanceRef implemented by: RTriggerInAtomicSwc InstanceRef

Table B.63: ExternalTriggerOccurredEvent

Class	ExternalTriggeringPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::Trigger			
Note	If a RunnableEntity owns an ExternalTriggeringPoint it is entitled to raise an ExternalTriggerOccurred Event.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
ident	ExternalTriggeringPoint Ident	0..1	aggr	The aggregation in the role ident provides the ability to make the ExternalTriggeringPoint identifiable. From the semantical point of view, the ExternalTriggering Point is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let ModeAccessPoint directly inherit from Identifiable). Stereotypes: atplIdentityContributor Tags: xml.sequenceOffset=-100
trigger	Trigger	0..1	iref	The trigger taken for the ExternalTriggeringPoint. Tags: xml.namePlural=TRIGGER-IREF xml.roleElement=false xml.roleWrapperElement=true xml.typeElement=true xml.typeWrapperElement=false InstanceRef implemented by: PTriggerInAtomicSwcType InstanceRef

Table B.64: ExternalTriggeringPoint

Class	FlatInstanceDescriptor			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	Represents exactly one node (e.g. a component instance or data element) of the instance tree of a software system. The purpose of this element is to map the various nested representations of this instance to a flat representation and assign a unique name (shortName) to it. Use cases: <ul style="list-style-type: none"> • Specify unique names of measurable data to be used by MCD tools • Specify unique names of calibration data to be used by MCD tool • Specify a unique name for an instance of a component prototype in the ECU extract of the system description Note that in addition it is possible to assign alias names via AliasNameAssignment.			
Base	<i>ARObject</i> , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note





Class	FlatInstanceDescriptor			
ecuExtract Reference	AtpFeature	0..1	iref	<p>Refers to the instance in the ECU extract. This is valid only, if the FlatMap is used in the context of an ECU extract.</p> <p>The reference shall be such that it uniquely defines the object instance. For example, if a data prototype is declared as a role within an SwcInternalBehavior, it is not enough to state the SwcInternalBehavior as context and the aggregated data prototype as target. In addition, the reference shall also include the complete path identifying instance of the component prototype and the Atomic SoftwareComponentType, which is referred by the particular SwcInternalBehavior.</p> <p>Tags:xml.sequenceOffset=40 InstanceRef implemented by:AnyInstanceRef</p>
role	Identifier	0..1	attr	<p>The role denotes the particular role of the downstream memory location described by this FlatInstanceDescriptor.</p> <p>It applies to use case where one upstream object results in multiple downstream objects, e.g. ModeDeclaration GroupPrototypes which are measurable. In this case the RTE will provide locations for current mode, previous mode and next mode.</p>
rtePluginProps	RtePluginProps	0..1	aggr	<p>The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in.</p> <p>Stereotypes: atpSplitable Tags:atp.Splitkey=rtePluginProps</p>
swDataDef Props	SwDataDefProps	0..1	aggr	<p>The properties of this FlatInstanceDescriptor.</p>
upstream Reference	AtpFeature	0..1	iref	<p>Refers to the instance in the context of an "upstream" descriptions, which could be the system or system extract description, the basic software module description or (if a flat map is used in preliminary context) a description of an atomic component or composition. This reference is optional in case the flat map is used in ECU context.</p> <p>The reference shall be such that it uniquely defines the object instance in the given context. For example, if a data prototype is declared as a role within an SwcInternalBehavior, it is not enough to state the SwcInternalBehavior as context and the aggregated data prototype as target. In addition, the reference shall also include the complete path identifying the instance of the component prototype that contains the particular instance of Swc InternalBehavior.</p> <p>Tags:xml.sequenceOffset=20 InstanceRef implemented by:AnyInstanceRef</p>

Table B.65: FlatInstanceDescriptor

Class	FunctionInhibitionAvailabilityNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs on the configuration of the Function Inhibition Manager to provide the control function for one Function Identifier (FID).			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>ServiceNeeds</i>			
Attribute	Type	Mult.	Kind	Note





Class	FunctionInhibitionAvailabilityNeeds			
controlledFid	FunctionInhibitionNeeds	0..1	ref	This reference represents the controlled FID

Table B.66: FunctionInhibitionAvailabilityNeeds

Class	FunctionInhibitionNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs on the configuration of the Function Inhibition Manager for one Function Identifier (FID). This class currently contains no attributes. Its name can be regarded as a symbol identifying the FID from the viewpoint of the component or module which owns this class.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable , ServiceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table B.67: FunctionInhibitionNeeds

Enumeration	HandleInvalidEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication
Note	Strategies of handling the reception of invalidValue.
Literal	Description
dontInvalidate	Invalidation is switched off. Tags: atp.EnumerationLiteralIndex=0
externalReplacement	Replace a received invalidValue. The replacement value is sourced from the externalReplacement. Tags: atp.EnumerationLiteralIndex=1
keep	The application software is supposed to handle signal invalidation on RTE API level either by Data ReceiveErrorEvent or check of error code on read access. Tags: atp.EnumerationLiteralIndex=2
replace	Replace a received invalidValue. The replacement value is specified by the initialValue. Tags: atp.EnumerationLiteralIndex=3

Table B.68: HandleInvalidEnum

Class	Identifiable (abstract)
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable
Note	Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables.
Base	ARObject, MultilanguageReferrable , Referrable
Subclasses	ARPackage, AbstractDolpLogicAddressProps , AbstractEvent , AbstractImplementationDataTypeElement , AbstractSecurityEventFilter , AbstractSecurityIdsmInstanceFilter , AbstractServiceInstance , AppOsTaskProxyToEcuTaskProxyMapping, ApplicationEndpoint, ApplicationError , ApplicationPartitionToEcuPartitionMapping, AsynchronousServerCallResultPoint, AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpFeature , AutosarOperationArgumentInstance, AutosarVariableInstance, BinaryManifestAddressableObject , BinaryManifestItemDefinition , BinaryManifestResource , BinaryManifestResourceDefinition, BlockState, BswInternalTriggeringPoint, BswModuleDependency, BuildActionEntity , BuildActionEnvironment, CanTpAddress, CanTpChannel, CanTpNode, Chapter, ClassContentConditional, ClientIdDefinition,





Class	Identifiable (abstract)			
	<p style="text-align: center;">△</p> <p>ClientServerOperation, Code, CollectableElement, ComManagementMapping, CommConnectorPort, CommunicationConnector, CommunicationController, Compiler, ConsistencyNeeds, ConsumedEvent Group, CouplingPort, CouplingPortStructuralElement, CpSoftwareClusterResource, CpSoftwareCluster ResourceToApplicationPartitionMapping, CpSoftwareClusterToEcuInstanceMapping, CpSoftwareCluster ToResourceMapping, CryptoServiceMapping, DataPrototypeGroup, DataTransformation, Dependency OnArtifact, DiagEventDebounceAlgorithm, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticDebounceAlgorithmProps, DiagnosticFunctionInhibitSource, DiagnosticRoutineSubfunction, DltApplication, DltArgument, DltLogChannel, DltMessage, DolpInterface, DolpLogicAddress, Dolp RoutingActivation, ECUMapping, EOCExecutableEntityRefAbstract, EcuPartition, EcucContainerValue, EcucDefinitionElement, EcucDestinationUriDef, EcucEnumerationLiteralDef, EcucQuery, EcucValidation Condition, EndToEndProtection, EthernetWakeupSleepOnDatalineConfig, EventHandler, ExclusiveArea, ExecutableEntity, ExecutionTime, FMAttributeDef, FMFeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeatureRelation, FMFeatureRestriction, FMFeatureSelection, FlatInstance Descriptor, FlexrayArTpNode, FlexrayTpConnectionControl, FlexrayTpNode, FlexrayTpPduPool, Frame Triggering, GeneralParameter, GlobalTimeGateway, GlobalTimeMaster, GlobalTimeSlave, HeapUsage, HwAttributeDef, HwAttributeLiteralDef, HwPin, HwPinGroup, IPSecRule, IPv6ExtHeaderFilterList, ISignal ToIpPduMapping, ISignalTriggering, IdentCaption, InternalTriggeringPoint, J1939SharedAddressCluster, J1939TpNode, Keyword, LifeCycleState, LinScheduleTable, LinTpNode, Linker, MacMulticastGroup, Mc DataInstance, MemorySection, ModeDeclaration, ModeDeclarationMapping, ModeSwitchPoint, Network Endpoint, NmCluster, NmEcu, NmNode, NvBlockDescriptor, PackageableElement, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, PerInstanceMemory, Physical Channel, PortElementToCommunicationResourceMapping, PortGroup, PortInterfaceMapping, Possible ErrorReaction, ResourceConsumption, RootSwCompositionPrototype, RptComponent, RptContainer, RptExecutableEntity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptServicePoint, Rte EventInCompositionSeparation, RteEventInCompositionToOsTaskProxyMapping, RteEventInSystem Separation, RteEventInSystemToOsTaskProxyMapping, RunnableEntityGroup, SdgAttribute, SdgClass, SecureCommunicationAuthenticationProps, SecureCommunicationFreshnessProps, SecurityEvent ContextProps, ServerCallPoint, ServiceNeeds, SignalServiceTranslationElementProps, SignalService TranslationEventProps, SignalServiceTranslationProps, SocketAddress, SomeipTpChannel, Spec ElementReference, StackUsage, StaticSocketConnection, StructuredReq, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SwcToApplicationPartitionMapping, SwcToEcuMapping, SwcTo ImplMapping, SystemMapping, TDCpSoftwareClusterMapping, TDCpSoftwareClusterResourceMapping, TcpOptionFilterList, TimingCondition, TimingConstraint, TimingDescription, TimingExtensionResource, TimingModelInstance, TlsCryptoCipherSuite, TlsCryptoCipherSuiteProps, Topic1, TpAddress, Traceable Table, TraceableText, TracedFailure, TransformationProps, TransformationTechnology, Trigger, Variable Access, VariationPointProxy, ViewMap, VlanConfig, WaitPoint</p>			
Attribute	Type	Mult.	Kind	Note
adminData	AdminData	0..1	aggr	<p>This represents the administrative data for the identifiable object.</p> <p>Stereotypes: atpSplittable Tags: atp.Splitkey=adminData xml.sequenceOffset=-40</p>
annotation	Annotation	*	aggr	<p>Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes.</p> <p>Tags:xml.sequenceOffset=-25</p>
category	CategoryString	0..1	attr	<p>The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.</p> <p>Tags:xml.sequenceOffset=-50</p>





Class	Identifiable (abstract)			
desc	MultiLanguageOverviewParagraph	0..1	aggr	<p>This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question.</p> <p>More elaborate documentation, (in particular how the object is built or used) should go to "introduction".</p> <p>Tags:xml.sequenceOffset=-60</p>
introduction	DocumentationBlock	0..1	aggr	<p>This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock.</p> <p>Tags:xml.sequenceOffset=-30</p>
uuid	String	0..1	attr	<p>The purpose of this attribute is to provide a globally unique identifier for an instance of a meta-class. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is "DCE:2fac1234-31f8-11b4-a222-08002b34c003". The uuid attribute has no semantic meaning for an AUTOSAR model and there is no requirement for AUTOSAR tools to manage the timestamp.</p> <p>Tags:xml.attribute=true</p>

Table B.69: Identifiable

Class	ImplementationDataType			
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
Note	<p>Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code.</p> <p>Tags:atp.recommendedPackage=ImplementationDataTypes</p>			
Base	<p><i>ARElement</i>, <i>AObject</i>, <i>AbstractImplementationDataType</i>, <i>AtpBlueprint</i>, <i>AtpBlueprintable</i>, <i>AtpClassifier</i>, <i>AtpType</i>, <i>AutosarDataType</i>, <i>CollectableElement</i>, <i>Identifiable</i>, <i>MultilanguageReferrable</i>, <i>PackageableElement</i>, <i>Referrable</i></p>			
Attribute	Type	Mult.	Kind	Note
dynamicArraySizeProfile	String	0..1	attr	<p>Specifies the profile which the array will follow in case this data type is a variable size array.</p>
isStructWithOptionalElement	Boolean	0..1	attr	<p>This attribute is only valid if the attribute category is set to STRUCTURE.</p> <p>If set to True, this attribute indicates that the ImplementationDataType has been created with the intention to define at least one element of the structure as optional.</p>





Class	ImplementationDataType			
subElement (ordered)	ImplementationDataTypeElement	*	aggr	Specifies an element of an array, struct, or union data type. The aggregation of ImplementationDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a ImplementationDataType representing a structure. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the ImplementationDataType. Stereotypes: atpSplittable Tags: atp.Splitkey=symbolProps.shortName
typeEmitter	NameToken	0..1	attr	This attribute is used to control which part of the AUTOSAR toolchain is supposed to trigger data type definitions.

Table B.70: ImplementationDataType

Class	ImplementationDataTypeElement			
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
Note	<p>Declares a data object which is locally aggregated. Such an element can only be used within the scope where it is aggregated.</p> <p>This element either consists of further subElements or it is further defined via its swDataDefProps.</p> <p>There are several use cases within the system of ImplementationDataTypes for such a local declaration:</p> <ul style="list-style-type: none"> • It can represent the elements of an array, defining the element type and array size • It can represent an element of a struct, defining its type • It can be the local declaration of a debug element. 			
Base	ARObject , AbstractImplementationDataTypeElement , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
arrayImplPolicy	ArrayImplPolicyEnum	0..1	attr	This attribute controls the implementation of the payload of an array. It shall only be used if the enclosing ImplementationDataType constitutes an array.
arraySize	PositiveInteger	0..1	attr	The existence of this attributes (if bigger than 0) defines the size of an array and declares that this ImplementationDataTypeElement represents the type of each single array element. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
arraySizeHandling	ArraySizeHandlingEnum	0..1	attr	The way how the size of the array is handled in case of a variable size array.
arraySizeSemantics	ArraySizeSemanticsEnum	0..1	attr	This attribute controls the meaning of the value of the array size.
isOptional	Boolean	0..1	attr	This attribute represents the ability to declare the enclosing ImplementationDataTypeElement as optional. This means that, at runtime, the ImplementationDataTypeElement may or may not have a valid value and shall therefore be ignored. The underlying runtime software provides means to set the CppImplementationDataTypeElement as not valid at the sending end of a communication and determine its validity at the receiving end.





Class	ImplementationDataTypeElement			
subElement (ordered)	ImplementationDataTypeElement	*	aggr	<p>Element of an array, struct, or union in case of a nested declaration (i.e. without using "typedefs").</p> <p>The aggregation of ImplementationDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a ImplementationDataType representing a structure.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=preCompileTime</p>
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this ImplementationDataTypeElement.

Table B.71: ImplementationDataTypeElement

Class	InternalBehavior (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters.			
Base	ARObject , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	BswInternalBehavior, SwcInternalBehavior			
Attribute	Type	Mult.	Kind	Note
constantMemory	ParameterDataPrototype	*	aggr	<p>Describes a read only memory object containing characteristic value(s) implemented by this InternalBehavior.</p> <p>The shortName of ParameterDataPrototype has to be equal to the "C" identifier of the described constant.</p> <p>The characteristic value(s) might be shared between SwComponentPrototypes of the same SwComponentType.</p> <p>The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=constantMemory.shortName, constantMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
constantValueMapping	ConstantSpecificationMappingSet	*	ref	<p>Reference to the ConstantSpecificationMapping to be applied for the particular InternalBehavior</p> <p>Stereotypes: atpSplitable Tags:atp.Splitkey=constantValueMapping</p>
dataTypeMapping	DataTypeMappingSet	*	ref	<p>Reference to the DataTypeMapping to be applied for the particular InternalBehavior</p> <p>Stereotypes: atpSplitable Tags:atp.Splitkey=dataTypeMapping</p>





Class	InternalBehavior (abstract)			
exclusiveArea	ExclusiveArea	*	aggr	<p>This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module. The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the conditional existence of RunnableEntities or BswModuleEntities.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=exclusiveArea.shortName, exclusiveArea.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	aggr	<p>This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=exclusiveAreaNestingOrder.shortName, exclusiveAreaNestingOrder.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
staticMemory	VariableDataPrototype	*	aggr	<p>Describes a read and writeable static memory object representing measurement variables implemented by this software component. The term "static" is used in the meaning of "non-temporary" and does not necessarily specify a linker encapsulation. This kind of memory is only supported if supportsMultipleInstantiation is FALSE.</p> <p>The shortName of the VariableDataPrototype has to be equal with the "C" identifier of the described variable.</p> <p>The aggregation of staticMemory is subject to variability with the purpose to support variability in the software component's implementations.</p> <p>Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=staticMemory.shortName, staticMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>

Table B.72: InternalBehavior

Class	InternalConstrs			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to express internal constraints.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
lowerLimit	Limit	0..1	attr	<p>This specifies the lower limit of the constraint.</p> <p>Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=20</p>
maxDiff	Numerical	0..1	attr	<p>Maximum difference that is permitted between two consecutive values if the constraint is applied to an axis.</p> <p>Tags:xml.sequenceOffset=60</p>





Class	InternalConstrs			
maxGradient	Numerical	0..1	attr	This element specifies the maximum slope that may be used in maps and curves. Tags: xml.sequenceOffset=50
monotony	MonotonyEnum	0..1	attr	This element specifies the monotony characteristics of the current internal or physical limits. The following table shows the monotony characteristics which are to be filled through the corresponding values. If the element has no contents or if it is omitted, "no Monotony" is the default content. Tags: xml.sequenceOffset=70
scaleConstr (ordered)	ScaleConstr	*	aggr	This is one particular scale which contributes to the data constraints. Tags: xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false
upperLimit	Limit	0..1	attr	This specifies the upper limit defined by the constraint. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

Table B.73: InternalConstrs

Class	InternalTriggerOccurredEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the referenced InternalTriggeringPoint has occurred.			
Base	<i>ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, Multilanguage Referrable, RTEEvent, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
eventSource	InternalTriggeringPoint	0..1	ref	The referenced InternalTriggeringPoint raises this Internal TriggerOccurredEvent.

Table B.74: InternalTriggerOccurredEvent

Class	InvalidationPolicy			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Specifies whether the component can actively invalidate a particular dataElement. If no invalidationPolicy points to a dataElement this is considered to yield the identical result as if the handleInvalid attribute was set to dontInvalidate.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	0..1	ref	Reference to the dataElement for which the Invalidation Policy applies.
handleInvalid	HandleInvalidEnum	0..1	attr	This attribute controls how invalidation is applied to the dataElement.

Table B.75: InvalidationPolicy

Class	McDataInstance			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport			
Note	<p>Describes the specific properties of one data instance in order to support measurement and/or calibration of this data instance.</p> <p>The most important attributes are:</p> <ul style="list-style-type: none"> • Its shortName is copied from the ECU Flat map (if applicable) and will be used as identifier and for display by the MC system. • The category is copied from the corresponding data type (ApplicationDataType if defined, otherwise ImplementationDataType) as far as applicable. • The symbol is the one used in the programming language. It will be used to find out the actual memory address by the final generation tool with the help of linker generated information. <p>It is assumed that in the M1 model this part and all the aggregated and referred elements (with the exception of the Flat Map and the references from ImplementationElementInParameterInstanceRef and McAccessDetails) are completely generated from "upstream" information. This means, that even if an element like e.g. a CompuMethod is only used via reference here, it will be copied into the M1 artifact which holds the complete McSupportData for a given Implementation.</p>			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
arraySize	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of number of elements.
displayIdentifier	McIdentifier	0..1	attr	An optional attribute to be used to set the ASAM ASAP2 DISPLAY_IDENTIFIER attribute.
flatMapEntry	FlatInstanceDescriptor	0..1	ref	Reference to the corresponding entry in the ECU Flat Map. This allows to trace back to the original specification of the generated data instance. This link shall be added by the RTE generator mainly for documentation purposes. The reference is optional because <ul style="list-style-type: none"> • The McDataInstance may represent an array or struct in which only the subElements correspond to FlatMap entries. • The McDataInstance may represent a task local buffer for rapid prototyping access which is different from the "main instance" used for measurement access.
instanceInMemory	ImplementationElementInParameterInstanceRef	0..1	aggr	Reference to the corresponding data instance in the description of calibration data structures published by the RTE generator. This is used to support emulation methods inside the ECU, it is not required for A2L generation.
mcDataAccessDetails	McDataAccessDetails	0..1	aggr	Refers to "upstream" information on how the RTE uses this data instance. Use Case: Rapid Prototyping
mcDataAssignment	RoleBasedMcDataAssignment	*	aggr	An assignment between McDataInstances. This supports the indication of related McDataElement implementing the of "RP global buffer", "RP global measurement buffer", "RP enabler flag".
resultingProperties	SwDataDefProps	0..1	aggr	These are the generated properties resulting from decisions taken by the RTE generator for the actually implemented data instance. Only those properties are relevant here, which are needed for the measurement and calibration system.
resultingRptSwPrototypingAccess	RptSwPrototypingAccess	0..1	aggr	Describes the implemented accessibility of data and modes by the rapid prototyping tooling.
role	Identifier	0..1	attr	An optional attribute to be used for additional information on the role of this data instance, for example in the context of rapid prototyping.





Class	McDataInstance			
rptImplPolicy	RptImplPolicy	0..1	aggr	Describes the implemented code preparation for rapid prototyping at data accesses for a hook based bypassing.
subElement (ordered)	McDataInstance	*	aggr	This relation indicates, that the target element is part of a "struct" which is given by the source element. This information will be used by the final generator to set up the correct addressing scheme. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
symbol	SymbolString	0..1	attr	This String is used to determine the memory address during final generation of the MC configuration data (e.g. "A2L" file) . It shall be the name of the element in the programming language such that it can be identified in linker generated information. In case the McDataInstance is part of composite data in the programming language, the symbol String may include parts denoting the element context, unless the context is given by the symbol attribute of an enclosing McDataInstance. This means in particular for the C language that the "." character shall be used as a separator between the name of a "struct" variable the name of one of its elements. The symbol can differ from the shortName in case of generated C data declarations. It is an optional attribute since it may be missing in case the instance represents an element (e.g. a single array element) which has no name in the linker map. Stereotypes: atpSplitable Tags: atp.Splitkey=symbol

Table B.76: McDataInstance

Class	McSupportData			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport			
Note	Root element for all measurement and calibration support data related to one Implementation artifact on an ECU. There shall be one such element related to the RTE implementation (if it owns MC data) and a separate one for each module or component, which owns private MC data.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
emulation Support	McSwEmulationMethod Support	*	aggr	Describes the calibration method used by the RTE. This information is not needed for A2L generation, but to setup software emulation in the ECU. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
mcParameter Instance	McDataInstance	*	aggr	A data instance to be used for calibration. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=mcParameterInstance.shortName, mc ParameterInstance.variationPoint.shortLabel vh.latestBindingTime=postBuild





Class	McSupportData			
mcVariable Instance	McDataInstance	*	aggr	A data instance to be used for measurement. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=mcVariableInstance.shortName, mcVariableInstance.variationPoint.shortLabel vh.latestBindingTime=postBuild
measurable System ConstantValues	SwSystemconstant ValueSet	*	ref	Sets of system constant values to be transferred to the MCD system, because the system constants have been specified with "swCalibrationAccess" = readonly.
rptSupportData	RptSupportData	0..1	aggr	The rapid prototyping support data belonging to this implementation. The aggregation is <<atpSplitable>> because in case of an already existing BSW Implementation model, this description will be added later in the process, namely at code generation time. Stereotypes: atpSplitable Tags: atp.Splitkey=rptSupportData

Table B.77: McSupportData

Class	ModeAccessPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ModeDeclarationGroup			
Note	A ModeAccessPoint is required by a RunnableEntity owned by a Mode Manager or Mode User. Its semantics implies the ability to access the current mode (provided by the RTE) of a ModeDeclaration GroupPrototype's ModeDeclarationGroup.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
ident	ModeAccessPointIdent	0..1	aggr	The aggregation in the role ident provides the ability to make the ModeAccessPoint identifiable. From the semantical point of view, the ModeAccessPoint is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let ModeAccessPoint directly inherit from Identifiable). Stereotypes: atpIdentityContributor Tags: xml.sequenceOffset=-100
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	The mode declaration group that is accessed by this runnable. Tags: xml.typeElement=true InstanceRef implemented by: ModeGroupInAtomicSwc InstanceRef

Table B.78: ModeAccessPoint

Class	ModeDeclaration			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
value	PositiveInteger	0..1	attr	The RTE shall take the value of this attribute for generating the source code representation of this Mode Declaration.

Table B.79: ModeDeclaration

Class	ModeDeclarationGroup			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. Tags: atp.recommendedPackage=ModeDeclarationGroups			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
initialMode	ModeDeclaration	0..1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred.
mode Declaration	ModeDeclaration	*	aggr	The ModeDeclarations collected in this ModeDeclaration Group. Stereotypes: atpVariation Tags: vh.latestBindingTime=blueprintDerivationTime
modeManager ErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the ability to define the error behavior expected by the mode manager in case of errors on the mode user side (e.g. terminated mode user).
modeTransition	ModeTransition	*	aggr	This represents the available ModeTransitions of the ModeDeclarationGroup
modeUserError Behavior	ModeErrorBehavior	0..1	aggr	This represents the definition of the error behavior expected by the mode user in case of errors on the mode manager side (e.g. terminated mode manager).
onTransition Value	PositiveInteger	0..1	attr	The value of this attribute shall be taken into account by the RTE generator for programmatically representing a value used for the transition between two statuses.

Table B.80: ModeDeclarationGroup

Class	ModeDeclarationGroupPrototype			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context.			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	This allows for specifying whether or not the enclosing ModeDeclarationGroupPrototype can be measured at run-time.
type	ModeDeclarationGroup	0..1	trf	The "collection of ModeDeclarations" (= ModeDeclaration Group) supported by a component Stereotypes: isOfType

Table B.81: ModeDeclarationGroupPrototype

Class	ModeDeclarationMappingSet			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class implements a container for ModeDeclarationGroupMappings Tags: atp.recommendedPackage=PortInterfaceMappingSets			
Base	ARElement, ARObject, AtpClassifier, AtpType, CollectableElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note





Class		ModeDeclarationMappingSet		
mode Declaration Mapping	ModeDeclaration Mapping	*	aggr	This represents the collection of ModeDeclaration Mappings owned by the enclosing ModeDeclaration MappingSet.

Table B.82: ModeDeclarationMappingSet

Class		ModeErrorBehavior		
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	This represents the ability to define the error behavior in the context of mode handling.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
defaultMode	ModeDeclaration	0..1	ref	This represents the ModeDeclaration that is considered the error mode in the context of the enclosing Mode DeclarationGroup.
errorReaction Policy	ModeErrorReaction PolicyEnum	0..1	attr	This represents the ability to define the policy in terms of which default model shall apply in case an error occurs.

Table B.83: ModeErrorBehavior

Class		ModeSwitchInterface		
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A mode switch interface declares a ModeDeclarationGroupPrototype to be sent and received. Tags: atp.recommendedPackage=PortInterfaces			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , PortInterface , Referrable			
Attribute	Type	Mult.	Kind	Note
modeGroup	ModeDeclarationGroup Prototype	0..1	aggr	The ModeDeclarationGroupPrototype of this mode interface.

Table B.84: ModeSwitchInterface

Class		ModeSwitchPoint		
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ModeDeclarationGroup			
Note	A ModeSwitchPoint is required by a RunnableEntity owned a Mode Manager. Its semantics implies the ability to initiate a mode switch.			
Base	ARObject , AbstractAccessPoint , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	The mode declaration group that is switched by this runnable. InstanceRef implemented by: PModeGroupInAtomic SwcInstanceRef

Table B.85: ModeSwitchPoint

Class		ModeTransition		
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			





Class	ModeTransition			
Note	This meta-class represents the ability to describe possible ModeTransitions in the context of a Mode DeclarationGroup.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
enteredMode	ModeDeclaration	0..1	ref	This represents the entered model of the ModeTransition.
exitedMode	ModeDeclaration	0..1	ref	This represents the exited mode of the ModeTransition

Table B.86: ModeTransition

Enumeration	MonotonyEnum
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	This enumerator denotes the values for specification of monotony for e.g. curves.
Literal	Description
decreasing	This indicates that the related curve needs to be monotony decreasing. Tags: atp.EnumerationLiteralIndex=0
increasing	This indicates that the related curve needs to be monotony increasing. Tags: atp.EnumerationLiteralIndex=1
monotonous	This indicates that the values shall be monotonously decreasing or increasing, depending on the trend set by the first values of the series. Tags: atp.EnumerationLiteralIndex=2
noMonotony	This indicates that the related curve needs not to be monotony. Tags: atp.EnumerationLiteralIndex=3
strictlyDecreasing	This indicates that the related curve needs to be strictly monotony decreasing. Tags: atp.EnumerationLiteralIndex=4
strictlyIncreasing	This indicates that the related curve needs to be strictly monotony increasing. Tags: atp.EnumerationLiteralIndex=5
strictMonotonous	This indicates that the values shall be strict monotonously decreasing or increasing, depending on the trend set by the first values of the series. Tags: atp.EnumerationLiteralIndex=6

Table B.87: MonotonyEnum

Class	NonqueuedReceiverComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes specific to non-queued receiving.			
Base	ARObject, RPortComSpec, ReceiverComSpec			
Attribute	Type	Mult.	Kind	Note
aliveTimeout	TimeValue	0..1	attr	Specify the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been received according to the specified timing description. If the aliveTimeout attribute is 0 no timeout monitoring shall be performed.
enableUpdate	Boolean	0..1	attr	This attribute controls whether application code is entitled to check whether the value of the corresponding Variable DataPrototype has been updated.





Class		NonqueuedReceiverComSpec		
filter	DataFilter	0..1	aggr	The applicable filter algorithm for filtering the value of the corresponding dataElement.
handleData Status	Boolean	0..1	attr	If this attribute is set to true, then the Rte_IStatus API shall exist. If the attribute does not exist or is set to false, then the Rte_IStatus API may still exist in response to the existence of further conditions.
handleNever Received	Boolean	0..1	attr	This attribute specifies whether for the corresponding VariableDataPrototype the "never received" flag is available. If yes, the RTE is supposed to assume that initially the VariableDataPrototype has not been received before. After the first reception of the corresponding VariableDataPrototype the flag is cleared. <ul style="list-style-type: none"> • If the value of this attribute is set to "true" the flag is required. • If set to "false", the RTE shall not support the "never received" functionality for the corresponding VariableDataPrototype.
handleTimeout Type	HandleTimeoutEnum	0..1	attr	This attribute controls the behavior with respect to the handling of timeouts.
initValue	ValueSpecification	0..1	aggr	Initial value to be used in case the sending component is not yet initialized. If the sender also specifies an initial value, then the receiver's value will be used.
timeout Substitution Value	ValueSpecification	0..1	aggr	This attribute represents the substitution value applicable in the case of a timeout.

Table B.88: NonqueuedReceiverComSpec

Class		NonqueuedSenderComSpec		
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for non-queued sender/receiver communication (sender side)			
Base	ARObject, PPortComSpec, SenderComSpec			
Attribute	Type	Mult.	Kind	Note
dataFilter	DataFilter	0..1	aggr	The applicable filter algorithm for filtering the value of the corresponding dataElement.
initValue	ValueSpecification	0..1	aggr	Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.

Table B.89: NonqueuedSenderComSpec

Class		NvBlockNeeds		
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs on the configuration of a single NVRAM Block.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	Type	Mult.	Kind	Note
calcRamBlock Crc	Boolean	0..1	attr	Defines if CRC (re)calculation for the permanent RAM Block is required.
checkStatic BlockId	Boolean	0..1	attr	Defines if the Static Block Id check shall be enabled.
cyclicWriting Period	TimeValue	0..1	attr	This represents the period for cyclic writing of NvData to store the associated RAM Block.





Class	NvBlockNeeds			
nDataSets	PositiveInteger	0..1	attr	Number of data sets to be provided by the NVRAM manager for this block. This is the total number of ROM Blocks and RAM Blocks.
nRomBlocks	PositiveInteger	0..1	attr	Number of ROM Blocks to be provided by the NVRAM manager for this block. Please note that these multiple ROM Blocks are given in a contiguous area.
ramBlockStatusControl	RamBlockStatusControlEnum	0..1	attr	This attribute defines how the management of the RAM Block status is controlled.
readonly	Boolean	0..1	attr	True: data of this NVRAM Block are write protected for normal operation (but protection can be disabled) false: no restriction
reliability	NvBlockNeedsReliabilityEnum	0..1	attr	Reliability against data loss on the non-volatile medium.
resistantToChangedSw	Boolean	0..1	attr	Defines whether an NVRAM Block shall be treated resistant to configuration changes (true) or not (false). For details how to handle initialization in the latter case, please refer to the NVRAM specification.
restoreAtStart	Boolean	0..1	attr	Defines whether the associated RAM Block shall be implicitly restored during startup by the basic software.
selectBlockForFirstInitAll	Boolean	0..1	attr	If this attribute is set to true the NvM shall process this block in the NvM_FirstInitAll() function.
storeAtShutdown	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored during shutdown by the basic software.
storeCyclic	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored periodically by the basic software.
storeEmergency	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored in case of ECU failure (e.g. loss of power) by the basic software. If the attribute storeEmergency is set to true the associated RAM Block shall be configured to have immediate priority.
storeImmediate	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored immediately during or after execution of the according SW-C RunnableEntity by the basic software.
storeOnChange	Boolean	0..1	attr	This attribute defines whether the associated RAM Block shall be stored immediately if the written value is different to the value stored in the associated RAM Block(s) during or after execution of the according SW-C RunnableEntity.
useAutoValidationAtShutDown	Boolean	0..1	attr	If set to true the RAM Block shall be auto validated during shutdown phase.
useCRCCompMechanism	Boolean	0..1	attr	If set to true the CRC of the RAM Block shall be compared during a write job with the CRC which was calculated during the last successful read or write job in order to skip unnecessary NVRAM writings.
writeOnlyOnce	Boolean	0..1	attr	Defines write protection after first write: true: This block is prevented from being changed/erased or being replaced with the default ROM data after first initialization by the software-component. false: No such restriction.
writeVerification	Boolean	0..1	attr	Defines if Write Verification shall be enabled for this NVRAM Block.
writingFrequency	PositiveInteger	0..1	attr	Provides the amount of updates to this block from the application point of view. It has to be provided in "number of write access per year".





Class	NvBlockNeeds			
writingPriority	NvBlockNeedsWritingPriorityEnum	0..1	attr	Requires the priority of writing this block in case of concurrent requests to write other blocks.

Table B.90: NvBlockNeeds

Class	NvDataInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A non volatile data interface declares a number of VariableDataPrototypes to be exchanged between non volatile block components and atomic software components. Tags: atp.recommendedPackage=PortInterfaces			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, DataInterface, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable			
Attribute	Type	Mult.	Kind	Note
nvData	VariableDataPrototype	*	aggr	The VariableDataPrototype of this nv data interface.

Table B.91: NvDataInterface

Class	OperationInvokedEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the ClientServerOperation referenced in OperationInvokedEvent.operation shall be invoked.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, RTEEvent, Referrable			
Attribute	Type	Mult.	Kind	Note
operation	ClientServerOperation	0..1	iref	This represents the ClientServerOperation which shall be invoked. InstanceRef implemented by: POperationInAtomicSwc InstanceRef

Table B.92: OperationInvokedEvent

Class	OsTaskExecutionEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This RTEEvent is supposed to execute RunnableEntities which have to react on the execution of specific OsTasks. Therefore, this event is unconditionally raised whenever the OsTask on which it is mapped is executed. The main use case for this event is scheduling of Runnables of Complex Drivers which have to react on task executions. Tags: atp.Status=draft			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, RTEEvent, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table B.93: OsTaskExecutionEvent

Class	PPortPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Component port providing a certain port interface.			





Class	PPortPrototype			
Base	<i>ARObject, AbstractProvidedPortPrototype, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, PortPrototype, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
provided Interface	PortInterface	0..1	tref	The interface that this port provides. Stereotypes: isOfType

Table B.94: PPortPrototype

Class	PRPortPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This kind of PortPrototype can take the role of both a required and a provided PortPrototype.			
Base	<i>ARObject, AbstractProvidedPortPrototype, AbstractRequiredPortPrototype, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, PortPrototype, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
provided Required Interface	PortInterface	0..1	tref	This represents the PortInterface used to type the PRPort Prototype Stereotypes: isOfType

Table B.95: PRPortPrototype

Class	ParameterAccess			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements			
Note	The presence of a ParameterAccess implies that a RunnableEntity needs access to a ParameterData Prototype.			
Base	<i>ARObject, AbstractAccessPoint, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
accessed Parameter	AutosarParameterRef	0..1	aggr	Reference to the accessed calibration parameter.
swDataDef Props	SwDataDefProps	0..1	aggr	This allows denote instance and access specific properties, mainly input values and common axis.

Table B.96: ParameterAccess

Class	ParameterDataPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	A parameter element used for parameter interface and internal behavior, supporting signal like parameter and characteristic value communication patterns and parameter and characteristic value definition.			
Base	<i>ARObject, AtpFeature, AtpPrototype, AutosarDataPrototype, DataPrototype, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the ParameterDataPrototype

Table B.97: ParameterDataPrototype

Class	ParameterInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A parameter interface declares a number of parameter and characteristic values to be exchanged between parameter components and software components. Tags: atp.recommendedPackage=PortInterfaces			
Base	<i>ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, DataInterface, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
parameter	ParameterData Prototype	*	aggr	The ParameterDataPrototype of this ParameterInterface.

Table B.98: ParameterInterface

Class	ParameterRequireComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	"Communication" specification that applies to parameters on the required side of a connection.			
Base	<i>ARObject, RPortComSpec</i>			
Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	The initial value applicable for the corresponding ParameterDataPrototype.
parameter	ParameterData Prototype	0..1	ref	The ParameterDataPrototype to which the Parameter RequireComSpec applies.

Table B.99: ParameterRequireComSpec

Class	PhysConstrs			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to express physical constraints. Therefore it has (in opposite to InternalConstrs) a reference to a Unit.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the constraint. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
maxDiff	Numerical	0..1	attr	Maximum difference that is permitted between two consecutive values if the constraint is applied to an axis. Tags: xml.sequenceOffset=60
maxGradient	Numerical	0..1	attr	This element specifies the maximum slope that may be used in curves and maps. Tags: xml.sequenceOffset=50
monotony	MonotonyEnum	0..1	attr	This specifies the monotony constraints on the data object. Note that this applies only to curves and maps. Tags: xml.sequenceOffset=70





Class	PhysConstrs			
scaleConstr (ordered)	ScaleConstr	*	aggr	This is one particular scale which contributes to the data constraints. Tags: xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false
unit	Unit	0..1	ref	This is the unit to which the physical constraints relate to. In particular, it is the physical unit of the specified limits. Tags: xml.sequenceOffset=80
upperLimit	Limit	0..1	attr	This specifies the upper limit of the constraint. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

Table B.100: PhysConstrs

Class	PhysicalDimension			
Package	M2::MSR::AsamHdo::Units			
Note	<p>This class represents a physical dimension. If the physical dimension of two units is identical, then a conversion between them is possible. The conversion between units is related to the definition of the physical dimension.</p> <p>Note that the equivalence of the exponents does not per se define the convertibility. For example Energy and Torque share the same exponents (Nm).</p> <p>Please note further the value of an exponent does not necessarily have to be an integer number. It is also possible that the value yields a rational number, e.g. to compute the square root of a given physical quantity. In this case the exponent value would be a rational number where the numerator value is 1 and the denominator value is 2.</p> <p>Tags:atp.recommendedPackage=PhysicalDimensions</p>			
Base	<i>ARElement, ARObjct, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
currentExp	Numerical	0..1	attr	This attribute represents the exponent of the physical dimension "electric current". Tags: xml.sequenceOffset=50
lengthExp	Numerical	0..1	attr	The exponent of the physical dimension "length". Tags: xml.sequenceOffset=20
luminous IntensityExp	Numerical	0..1	attr	The exponent of the physical dimension "luminous intensity". Tags: xml.sequenceOffset=80
massExp	Numerical	0..1	attr	The exponent of the physical dimension "mass". Tags: xml.sequenceOffset=30
molarAmount Exp	Numerical	0..1	attr	The exponent of the physical dimension "quantity of substance". Tags: xml.sequenceOffset=70
temperatureExp	Numerical	0..1	attr	The exponent of the physical dimension "temperature". Tags: xml.sequenceOffset=60





Class	PhysicalDimension			
timeExp	Numerical	0..1	attr	The exponent of the physical dimension "time". Tags: xml.sequenceOffset=40

Table B.101: PhysicalDimension

Class	PortElementToCommunicationResourceMapping			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	This meta class maps a communication resource to CP Software Clusters. In this case the kind of Port Prototype specified whether the Software Cluster has to provide or to require the resource. Tags: atp.Status=draft			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
clientServer Operation	ClientServerOperation	0..1	iref	ClientServerOperation instance qualifying the communication resource Tags: atp.Status=draft InstanceRef implemented by: OperationInSystemInstanceRef
communication Resource	CpSoftwareClusterCommunicationResource	0..1	ref	Communication resource for which the mapping applies. Tags: atp.Status=draft
mode Declaration GroupPrototype	ModeDeclarationGroupPrototype	0..1	iref	ModeDeclarationGroupPrototype instance qualifying the communication resource Tags: atp.Status=draft InstanceRef implemented by: ModeDeclarationGroupPrototypeInSystemInstanceRef
parameterData Prototype	ParameterDataPrototype	0..1	iref	ParameterDataPrototype instance qualifying the communication resource. Tags: atp.Status=draft InstanceRef implemented by: ParameterDataPrototypeInSystemInstanceRef
trigger	Trigger	0..1	iref	Trigger instance qualifying the communication resource. Tags: atp.Status=draft InstanceRef implemented by: TriggerInSystemInstanceRef
variableData Prototype	VariableDataPrototype	0..1	iref	VariableDataPrototype instance qualifying the communication resource Tags: atp.Status=draft InstanceRef implemented by: VariableDataPrototypeInSystemInstanceRef

Table B.102: PortElementToCommunicationResourceMapping

Class	PortInterface (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Abstract base class for an interface that is either provided or required by a port of a software component.			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Subclasses	ClientServerInterface , DataInterface , ModeSwitchInterface , TriggerInterface			
Attribute	Type	Mult.	Kind	Note





Class	PortInterface (abstract)			
isService	Boolean	0..1	attr	This flag is set if the PortInterface is to be used for communication between an <ul style="list-style-type: none"> • ApplicationSwComponentType or • ServiceProxySwComponentType or • SensorActuatorSwComponentType or • ComplexDeviceDriverSwComponentType • ServiceSwComponentType • EcuAbstractionSwComponentType and a ServiceSwComponentType (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set.
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service.

Table B.103: PortInterface

Class	PortInterfaceMapping (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Specifies one PortInterfaceMapping to support the connection of Ports typed by two different Port Interfaces with PortInterface elements having unequal names and/or unequal semantic (resolution or range).			
Base	<i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	ClientServerInterfaceMapping, ModelInterfaceMapping, TriggerInterfaceMapping, VariableAndParameterInterfaceMapping			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table B.104: PortInterfaceMapping

Class	PortPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Base class for the ports of an AUTOSAR software component. The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.			
Base	<i>ARObject</i> , <i>AtpBlueprintable</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	<i>AbstractProvidedPortPrototype</i> , <i>AbstractRequiredPortPrototype</i>			
Attribute	Type	Mult.	Kind	Note
clientServerAnnotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/server communication.
delegatedPortAnnotation	DelegatedPortAnnotation	0..1	aggr	Annotations on this delegated port.
ioHwAbstractionServerAnnotation	IoHwAbstractionServerAnnotation	*	aggr	Annotations on this IO Hardware Abstraction port.
logAndTraceMessageCollectionSet	LogAndTraceMessageCollectionSet	0..1	ref	Reference to a collection of Log or Trace messages that will be used by the application. Tags: atp.Status=draft
modePortAnnotation	ModePortAnnotation	*	aggr	Annotations on this mode port.





Class	PortPrototype (abstract)			
nvDataPort Annotation	NvDataPortAnnotation	*	aggr	Annotations on this non volatile data port.
parameterPort Annotation	ParameterPort Annotation	*	aggr	Annotations on this parameter port.
senderReceiver Annotation	SenderReceiver Annotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

Table B.105: PortPrototype

Class	QueuedReceiverComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes specific to queued receiving.			
Base	ARObject, RPortComSpec, ReceiverComSpec			
Attribute	Type	Mult.	Kind	Note
queueLength	PositiveInteger	0..1	attr	Length of queue for received events.

Table B.106: QueuedReceiverComSpec

Class	RModelInAtomicSwcInstanceRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components::InstanceRefs			
Note				
Base	ARObject, AtpInstanceRef			
Attribute	Type	Mult.	Kind	Note
base	AtomicSwcComponent Type	0..1	ref	Stereotypes: atpDerived Tags: xml.sequenceOffset=10
contextMode Declaration GroupPrototype	ModeDeclarationGroup Prototype	0..1	ref	Tags: xml.sequenceOffset=30
contextPort	AbstractRequiredPort Prototype	0..1	ref	Tags: xml.sequenceOffset=20
targetMode Declaration	ModeDeclaration	0..1	ref	Tags: xml.sequenceOffset=40

Table B.107: RModelInAtomicSwcInstanceRef

Class	RPortPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Component port requiring a certain port interface.			
Base	ARObject, AbstractRequiredPortPrototype, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable , MultilanguageReferrable , PortPrototype , Referrable			
Attribute	Type	Mult.	Kind	Note
mayBe Unconnected	Boolean	0..1	attr	If set to true, this attribute indicates that the enclosing RPortPrototype may be left unconnected and that this aspect has explicitly been considered in the software-component's design.





Class	RPortPrototype			
required Interface	PortInterface	0..1	tref	The interface that this port requires. Stereotypes: isOfType

Table B.108: RPortPrototype

Class	RTEEvent (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	Abstract base class for all RTE-related events			
Base	<i>ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, Multilanguage Referrable, Referrable</i>			
Subclasses	AsynchronousServerCallReturnsEvent , BackgroundEvent , DataReceiveErrorEvent , DataReceivedEvent , DataSendCompletedEvent , DataWriteCompletedEvent , ExternalTriggerOccurredEvent , InitEvent , InternalTriggerOccurredEvent , ModeSwitchedAckEvent , OperationInvokedEvent , OsTaskExecutionEvent , SwcModeManagerErrorEvent , SwcModeSwitchEvent , TimingEvent , TransformerHardErrorEvent			
Attribute	Type	Mult.	Kind	Note
disabledMode	ModeDeclaration	*	iref	Reference to the Modes that disable the Event. Stereotypes: atpSplittable Tags: atp.Splitkey=disabledMode.contextPort, disabledMode.contextModeDeclarationGroupPrototype, disabledMode.targetModeDeclaration InstanceRef implemented by: RModelInAtomicSwc InstanceRef
startOnEvent	RunnableEntity	0..1	ref	The referenced RunnableEntity starts when the corresponding RTEEvent is raised.

Table B.109: RTEEvent

Class	RTriggerInAtomicSwcInstanceRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components::InstanceRefs			
Note				
Base	<i>ARObject, AtpInstanceRef, TriggerInAtomicSwcInstanceRef</i>			
Attribute	Type	Mult.	Kind	Note
contextRPort	AbstractRequiredPort Prototype	0..1	ref	Tags: xml.sequenceOffset=20
targetTrigger	Trigger	0..1	ref	Tags: xml.sequenceOffset=30

Table B.110: RTriggerInAtomicSwcInstanceRef

Class	Referrable (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
Base	<i>ARObject</i>			
Subclasses	<i>AtpDefinition, BswDistinguishedPartition, BswModuleCallPoint, BswModuleClientServerEntry, Bsw VariableAccess, CouplingPortTrafficClassAssignment, DiagnosticEnvModeElement, EthernetPriority Regeneration, ExclusiveAreaNestingOrder, HwDescriptionEntity, ImplementationProps, LinSlaveConfig Ident, ModeTransition, MultilanguageReferrable, PncMappingIdent, SingleLanguageReferrable, SoConfl PdulIdentifier, SocketConnectionBundle, TimeSyncServerConfiguration, TpConnectionIdent</i>			
Attribute	Type	Mult.	Kind	Note





Class		Referrable (abstract)		
shortName	Identifier	1	attr	This specifies an identifying shortName for the object. It needs to be unique within its context and is intended for humans but even more for technical reference. Stereotypes: atpIdentityContributor Tags: xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortName Fragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. Tags: xml.sequenceOffset=-90

Table B.111: Referrable

Class		RtePluginProps		
Package		M2::AUTOSARTemplates::CommonStructure::FlatMap		
Note		The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in.		
Base		ARObject		
Attribute	Type	Mult.	Kind	Note
associated CrossSwCluster ComRtePlugin	EcucContainerValue	0..1	ref	This associates a communication graph to a specific RTE Implementation Plug-in handling cross Software Cluster communication.
associatedRte Plugin	EcucContainerValue	0..1	ref	This associates a communication graph to a specific RTE Implementation Plug-in handling local Software Cluster communication or communication in a non-cluster ECU.

Table B.112: RtePluginProps

Class		RunnableEntity		
Package		M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior		
Note		A RunnableEntity represents the smallest code-fragment that is provided by an AtomicSwComponent Type and are executed under control of the RTE. RunnableEntities are for instance set up to respond to data reception or operation invocation on a server.		
Base		ARObject, AtpClassifier, AtpFeature, AtpStructureElement, ExecutableEntity , Identifiable , Multilanguage Referrable , Referrable		
Attribute	Type	Mult.	Kind	Note
argument (ordered)	RunnableEntity Argument	*	aggr	This represents the formal definition of a an argument to a RunnableEntity.
asynchronous ServerCall ResultPoint	AsynchronousServer CallResultPoint	*	aggr	The server call result point admits a runnable to fetch the result of an asynchronous server call. The aggregation of AsynchronousServerCallResultPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes and the variant existence of server call result points in the implementation. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=asynchronousServerCallResultPoint.short Name, asynchronousServerCallResultPoint.variation Point.shortLabel vh.latestBindingTime=preCompileTime





Class	RunnableEntity			
canBeInvoked Concurrently	Boolean	0..1	attr	If the value of this attribute is set to "true" the enclosing RunnableEntity can be invoked concurrently (even for one instance of the corresponding AtomicSwComponent Type). This implies that it is the responsibility of the implementation of the RunnableEntity to take care of this form of concurrency.
dataRead Access	VariableAccess	*	aggr	RunnableEntity has implicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The aggregation of dataReadAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataReadAccess in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=dataReadAccess.shortName, dataReadAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive PointBy Argument	VariableAccess	*	aggr	RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The result is passed back to the application by means of an argument in the function signature. The aggregation of dataReceivePointByArgument is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data receive points in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=dataReceivePointByArgument.shortName, dataReceivePointByArgument.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive PointByValue	VariableAccess	*	aggr	RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The result is passed back to the application by means of the return value. The aggregation of dataReceivePointByValue is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of data receive points in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=dataReceivePointByValue.shortName, dataReceivePointByValue.variationPoint.shortLabel vh.latestBindingTime=preCompileTime





Class	RunnableEntity			
dataSendPoint	VariableAccess	*	aggr	<p>RunnableEntity has explicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The aggregation of dataSendPoint is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data send points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=dataSendPoint.shortName, dataSendPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
dataWrite Access	VariableAccess	*	aggr	<p>RunnableEntity has implicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The aggregation of dataWriteAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataWriteAccess in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=dataWriteAccess.shortName, dataWriteAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
external TriggeringPoint	ExternalTriggeringPoint	*	aggr	<p>The aggregation of ExternalTriggeringPoint is subject to variability with the purpose to support the conditional existence of trigger ports or the variant existence of external triggering points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=externalTriggeringPoint.ident.shortName, externalTriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
internal TriggeringPoint	InternalTriggeringPoint	*	aggr	<p>The aggregation of InternalTriggeringPoint is subject to variability with the purpose to support the variant existence of internal triggering points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=internalTriggeringPoint.shortName, internalTriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
modeAccess Point	ModeAccessPoint	*	aggr	<p>The runnable has a mode access point. The aggregation of ModeAccessPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode access points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=modeAccessPoint.ident.shortName, modeAccessPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>





Class	RunnableEntity			
modeSwitch Point	ModeSwitchPoint	*	aggr	<p>The runnable has a mode switch point. The aggregation of ModeSwitchPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode switch points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=modeSwitchPoint.shortName, modeSwitchPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
parameter Access	ParameterAccess	*	aggr	<p>The presence of a ParameterAccess implies that a RunnableEntity needs read only access to a ParameterDataPrototype which may either be local or within a PortPrototype.</p> <p>The aggregation of ParameterAccess is subject to variability with the purpose to support the conditional existence of parameter ports and component local parameters as well as the variant existence of ParameterAccess (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=parameterAccess.shortName, parameterAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
readLocal Variable	VariableAccess	*	aggr	<p>The presence of a readLocalVariable implies that a RunnableEntity needs read access to a VariableDataPrototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable.</p> <p>The aggregation of readLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicitInterRunnableVariable or the variant existence of readLocalVariable (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=readLocalVariable.shortName, readLocalVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
serverCallPoint	ServerCallPoint	*	aggr	<p>The RunnableEntity has a ServerCallPoint. The aggregation of ServerCallPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes or the variant existence of server call points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=serverCallPoint.shortName, serverCallPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
symbol	CIdentifier	0..1	attr	<p>The symbol describing this RunnableEntity's entry point. This is considered the API of the RunnableEntity and is required during the RTE contract phase.</p>
waitPoint	WaitPoint	*	aggr	<p>The WaitPoint associated with the RunnableEntity.</p>





Class	RunnableEntity			
writtenLocalVariable	VariableAccess	*	aggr	<p>The presence of a writtenLocalVariable implies that a RunnableEntity needs write access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable.</p> <p>The aggregation of writtenLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicitInterRunnableVariable or the variant existence of writtenLocalVariable (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=writtenLocalVariable.shortName, writtenLocalVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>

Table B.113: RunnableEntity

Enumeration	SendIndicationEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster
Note	<p>This meta-class provides a way to specify in which way redundancy shall be applied on collection level.</p> <p>Tags:atp.Status=draft</p>
Literal	Description
anySendOperation	<p>This value represents the requirement that any send operation of the Software Cluster is indicated.</p> <p>Tags: atp.EnumerationLiteralIndex=2 atp.Status=draft</p>
none	<p>This value represents the requirement that send operations of the Software Cluster are not indicated.</p> <p>Tags: atp.EnumerationLiteralIndex=1 atp.Status=draft</p>

Table B.114: SendIndicationEnum

Class	SenderReceiverInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	<p>A sender/receiver interface declares a number of data elements to be sent and received.</p> <p>Tags:atp.recommendedPackage=PortInterfaces</p>			
Base	<i>ARElement, ARObjct, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, DataInterface, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	*	aggr	The data elements of this SenderReceiverInterface.
invalidationPolicy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement
metaDataItemSet	MetaDataItemSet	*	aggr	This aggregation defines fixed sets of meta-data items associated with dataElements of the enclosing SenderReceiverInterface

Table B.115: SenderReceiverInterface

Class	ServerCallPoint (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServerCall			
Note	If a RunnableEntity owns a ServerCallPoint it is entitled to invoke a particular ClientServerOperation of a specific RPortPrototype of the corresponding AtomicSwComponentType			
Base	ARObject, AbstractAccessPoint , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	AsynchronousServerCallPoint, SynchronousServerCallPoint			
Attribute	Type	Mult.	Kind	Note
operation	ClientServerOperation	0..1	iref	The operation that is called by this runnable. InstanceRef implemented by: ROperationInAtomicSwc InstanceRef
timeout	TimeValue	0..1	attr	Time in seconds before the server call times out and returns with an error message. It depends on the call type (synchronous or asynchronous) how this is reported.

Table B.116: ServerCallPoint

Class	ServerComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for a server port (PPortPrototype and ClientServerInterface).			
Base	ARObject, PPortComSpec			
Attribute	Type	Mult.	Kind	Note
operation	ClientServerOperation	0..1	ref	Operation these communication attributes apply to.
queueLength	PositiveInteger	0..1	attr	Length of call queue on the server side. The queue is implemented by the RTE. The value shall be greater or equal to 1. Setting the value of queueLength to 1 implies that incoming requests are rejected while another request that arrived earlier is being processed.
transformation ComSpecProps	TransformationCom SpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation.

Table B.117: ServerComSpec

Class	SubElementMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class allows for the definition of mappings of elements of a composite data type.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
firstElement	SubElementRef	0..1	aggr	This represents the first element referenced in the scope of the mapping. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
secondElement	SubElementRef	0..1	aggr	This represents the second element referenced in the scope of the mapping. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
textTable Mapping	TextTableMapping	0..2	aggr	This allows for the text-table translation of individual elements of a composite data type.

Table B.118: SubElementMapping

Class	SwBaseType			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This meta-class represents a base type used within ECU software. Tags: atp.recommendedPackage=BaseTypes			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, BaseType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table B.119: SwBaseType

Enumeration	SwCalibrationAccessEnum			
Package	M2::MSR::DataDictionary::DataDefProperties			
Note	Determines the access rights to a data object w.r.t. measurement and calibration.			
Literal	Description			
notAccessible	The element will not be accessible via MCD tools, i.e. will not appear in the ASAP file. Tags: atp.EnumerationLiteralIndex=0			
readOnly	The element will only appear as read-only in an ASAP file. Tags: atp.EnumerationLiteralIndex=1			
readWrite	The element will appear in the ASAP file with both read and write access. Tags: atp.EnumerationLiteralIndex=2			

Table B.120: SwCalibrationAccessEnum

Class	SwComponentPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	Role of a software component within a composition.			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
type	SwComponentType	0..1	tref	Type of the instance. Stereotypes: isOfType

Table B.121: SwComponentPrototype

Class	<<atpVariation>> SwDataDefProps			
Package	M2::MSR::DataDictionary::DataDefProperties			
Note	<p>This class is a collection of properties relevant for data objects under various aspects. One could consider this class as a "pattern of inheritance by aggregation". The properties can be applied to all objects of all classes in which SwDataDefProps is aggregated.</p> <p>Note that not all of the attributes or associated elements are useful all of the time. Hence, the process definition (e.g. expressed with an OCL or a Document Control Instance MSR-DCI) has the task of implementing limitations.</p> <p>SwDataDefProps covers various aspects:</p> <ul style="list-style-type: none"> • Structure of the data element for calibration use cases: is it a single value, a curve, or a map, but also the recordLayouts which specify how such elements are mapped/converted to the Data Types in the programming language (or in AUTOSAR). This is mainly expressed by properties like swRecordLayout and swCalprmAxisSet 			





Class	<<atpVariation>> SwDataDefProps			
	<p>△</p> <ul style="list-style-type: none"> Implementation aspects, mainly expressed by swImplPolicy, swVariableAccessImplPolicy, swAddrMethod, swPointerTargetProps, baseType, implementationDataType and additionalNativeTypeQualifier Access policy for the MCD system, mainly expressed by swCalibrationAccess Semantics of the data element, mainly expressed by compuMethod and/or unit, dataConstr, invalidValue Code generation policy provided by swRecordLayout <p>Tags:vh.latestBindingTime=codeGenerationTime</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
additionalNativeTypeQualifier	NativeDeclarationString	0..1	attr	<p>This attribute is used to declare native qualifiers of the programming language which can neither be deduced from the baseType (e.g. because the data object describes a pointer) nor from other more abstract attributes. Examples are qualifiers like "volatile", "strict" or "enum" of the C-language. All such declarations have to be put into one string.</p> <p>Tags:xml.sequenceOffset=235</p>
annotation	Annotation	*	aggr	<p>This aggregation allows to add annotations (yellow pads ...) related to the current data object.</p> <p>Tags: xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false</p>
baseType	SwBaseType	0..1	ref	<p>Base type associated with the containing data object.</p> <p>Tags:xml.sequenceOffset=50</p>
compuMethod	CompuMethod	0..1	ref	<p>Computation method associated with the semantics of this data object.</p> <p>Tags:xml.sequenceOffset=180</p>
dataConstr	DataConstr	0..1	ref	<p>Data constraint for this data object.</p> <p>Tags:xml.sequenceOffset=190</p>
displayFormat	DisplayFormatString	0..1	attr	<p>This property describes how a number is to be rendered e.g. in documents or in a measurement and calibration system.</p> <p>Tags:xml.sequenceOffset=210</p>
displayPresentation	DisplayPresentationEnum	0..1	attr	<p>This attribute controls the presentation of the related data for measurement and calibration tools.</p>
implementationDataType	AbstractImplementationDataType	0..1	ref	<p>This association denotes the ImplementationDataType of a data declaration via its aggregated SwDataDefProps. It is used whenever a data declaration is not directly referring to a base type. Especially</p> <ul style="list-style-type: none"> redefinition of an ImplementationDataType via a "typedef" to another ImplementationDatatype the target type of a pointer (see SwPointerTargetProps), if it does not refer to a base type directly <p>▽</p>





Class	<<atpVariation>> SwDataDefProps			
				<p>△</p> <ul style="list-style-type: none"> the data type of an array or record element within an ImplementationDataType, if it does not refer to a base type directly the data type of an SwServiceArg, if it does not refer to a base type directly <p>Tags:xml.sequenceOffset=215</p>
invalidValue	ValueSpecification	0..1	aggr	<p>Optional value to express invalidity of the actual data element.</p> <p>Tags:xml.sequenceOffset=255</p>
stepSize	Float	0..1	attr	<p>This attribute can be used to define a value which is added to or subtracted from the value of a DataPrototype when using up/down keys while calibrating.</p>
swAddrMethod	SwAddrMethod	0..1	ref	<p>Addressing method related to this data object. Via an association to the same SwAddrMethod it can be specified that several DataPrototypes shall be located in the same memory without already specifying the memory section itself.</p> <p>Tags:xml.sequenceOffset=30</p>
swAlignment	AlignmentType	0..1	attr	<p>The attribute describes the intended typical alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memoryAllocationKeywordPolicy of the referenced Sw AddrMethod.</p> <p>Tags:xml.sequenceOffset=33</p>
swBit Representation	SwBitRepresentation	0..1	aggr	<p>Description of the binary representation in case of a bit variable.</p> <p>Tags:xml.sequenceOffset=60</p>
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	<p>Specifies the read or write access by MCD tools for this data object.</p> <p>Tags:xml.sequenceOffset=70</p>
swCalprmAxis Set	SwCalprmAxisSet	0..1	aggr	<p>This specifies the properties of the axes in case of a curve or map etc. This is mainly applicable to calibration parameters.</p> <p>Tags:xml.sequenceOffset=90</p>
swComparison Variable	SwVariableRefProxy	*	aggr	<p>Variables used for comparison in an MCD process.</p> <p>Tags: xml.sequenceOffset=170 xml.typeElement=false</p>
swData Dependency	SwDataDependency	0..1	aggr	<p>Describes how the value of the data object has to be calculated from the value of another data object (by the MCD system).</p> <p>Tags:xml.sequenceOffset=200</p>
swHostVariable	SwVariableRefProxy	0..1	aggr	<p>Contains a reference to a variable which serves as a host-variable for a bit variable. Only applicable to bit objects.</p> <p>Tags: xml.sequenceOffset=220 xml.typeElement=false</p>
swImplPolicy	SwImplPolicyEnum	0..1	attr	<p>Implementation policy for this data object.</p> <p>Tags:xml.sequenceOffset=230</p>





Class	<<atpVariation>> SwDataDefProps			
swIntendedResolution	Numerical	0..1	attr	<p>The purpose of this element is to describe the requested quantization of data objects early on in the design process.</p> <p>The resolution ultimately occurs via the conversion formula present (compuMethod), which specifies the transition from the physical world to the standardized world (and vice-versa) (here, "the slope per bit" is present implicitly in the conversion formula).</p> <p>In the case of a development phase without a fixed conversion formula, a pre-specification can occur through swIntendedResolution.</p> <p>The resolution is specified in the physical domain according to the property "unit".</p> <p>Tags:xml.sequenceOffset=240</p>
swInterpolationMethod	Identifier	0..1	attr	<p>This is a keyword identifying the mathematical method to be applied for interpolation. The keyword needs to be related to the interpolation routine which needs to be invoked.</p> <p>Tags:xml.sequenceOffset=250</p>
swIsVirtual	Boolean	0..1	attr	<p>This element distinguishes virtual objects. Virtual objects do not appear in the memory, their derivation is much more dependent on other objects and hence they shall have a swDataDependency .</p> <p>Tags:xml.sequenceOffset=260</p>
swPointerTargetProps	SwPointerTargetProps	0..1	aggr	<p>Specifies that the containing data object is a pointer to another data object.</p> <p>Tags:xml.sequenceOffset=280</p>
swRecordLayout	SwRecordLayout	0..1	ref	<p>Record layout for this data object.</p> <p>Tags:xml.sequenceOffset=290</p>
swRefreshTiming	MultidimensionalTime	0..1	aggr	<p>This element specifies the frequency in which the object involved shall be or is called or calculated. This timing can be collected from the task in which write access processes to the variable run. But this cannot be done by the MCD system.</p> <p>So this attribute can be used in an early phase to express the desired refresh timing and later on to specify the real refresh timing.</p> <p>Tags:xml.sequenceOffset=300</p>
swTextProps	SwTextProps	0..1	aggr	<p>the specific properties if the data object is a text object.</p> <p>Tags:xml.sequenceOffset=120</p>
swValueBlockSize	Numerical	0..1	attr	<p>This represents the size of a Value Block</p> <p>Stereotypes: atpVariation</p> <p>Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=80</p>





Class	<<atpVariation>> SwDataDefProps			
swValueBlockSizeMult (ordered)	Numerical	*	attr	<p>This attribute is used to specify the dimensions of a value block (VAL_BLK) for the case that that value block has more than one dimension.</p> <p>The dimensions given in this attribute are ordered such that the first entry represents the first dimension, the second entry represents the second dimension, and so on.</p> <p>For one-dimensional value blocks the attribute swValueBlockSize shall be used and this attribute shall not exist.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=preCompileTime</p>
unit	Unit	0..1	ref	<p>Physical unit associated with the semantics of this data object. This attribute applies if no compuMethod is specified. If both units (this as well as via compuMethod) are specified the units shall be compatible.</p> <p>Tags:xml.sequenceOffset=350</p>
valueAxisDataType	ApplicationPrimitiveDataType	0..1	ref	<p>The referenced ApplicationPrimitiveDataType represents the primitive data type of the value axis within a compound primitive (e.g. curve, map). It supersedes CompuMethod, Unit, and BaseType.</p> <p>Tags:xml.sequenceOffset=355</p>

Table B.122: SwDataDefProps

Enumeration	SwImplPolicyEnum
Package	M2::MSR::DataDictionary::DataDefProperties
Note	Specifies the implementation strategy with respect to consistency mechanisms of variables.
Literal	Description
const	<p>forced implementation such that the running software within the ECU shall not modify it. For example implemented with the "const" modifier in C. This can be applied for parameters (not for those in NVRAM) as well as argument data prototypes.</p> <p>Tags:atp.EnumerationLiteralIndex=0</p>
fixed	<p>This data element is fixed. In particular this indicates, that it might also be implemented e.g. as in place data, (#DEFINE).</p> <p>Tags:atp.EnumerationLiteralIndex=1</p>
measurementPoint	<p>The data element is created for measurement purposes only. The data element is never read directly within the ECU software. In contrast to a "standard" data element in an unconnected provide port is, this unconnection is guaranteed for measurementPoint data elements.</p> <p>Tags:atp.EnumerationLiteralIndex=2</p>
queued	<p>The content of the data element is queued and the data element has 'event' semantics, i.e. data elements are stored in a queue and all data elements are processed in 'first in first out' order. The queuing is intended to be implemented by RTE Generator. This value is not applicable for parameters.</p> <p>Tags:atp.EnumerationLiteralIndex=3</p>
standard	<p>This is applicable for all kinds of data elements. For variable data prototypes the 'last is best' semantics applies. For parameter there is no specific implementation directive.</p> <p>Tags:atp.EnumerationLiteralIndex=4</p>

Table B.123: SwImplPolicyEnum

Class	SwRecordLayout			
Package	M2::MSR::DataDictionary::RecordLayout			
Note	Defines how the data objects (variables, calibration parameters etc.) are to be stored in the ECU memory. As an example, this definition specifies the sequence of axis points in the ECU memory. Iterations through axis values are stored within the sub-elements swRecordLayoutGroup. Tags: atp.recommendedPackage=SwRecordLayouts			
Base	ARElement, ARObject, CollectableElement, Identifiable , MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
swRecordLayoutGroup	SwRecordLayoutGroup	0..1	aggr	This is the top level record layout group. Tags: xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table B.124: SwRecordLayout

Class	SwcModeSwitchEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the specified mode change occurs.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable, RTEEvent , Referrable			
Attribute	Type	Mult.	Kind	Note
activation	ModeActivationKind	0..1	attr	Specifies if the event is raised on entering or exiting a specific mode or is raised on the transition between two modes.
mode (ordered)	ModeDeclaration	0..2	iref	The referenced mode or the transition between two modes raises this SwcModeSwitchEvent. InstanceRef implemented by: RModeInAtomicSwcInstanceRef

Table B.125: SwcModeSwitchEvent

Class	SynchronousServerCallPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServerCall			
Note	This means that the RunnableEntity is supposed to perform a blocking wait for a response from the server.			
Base	ARObject, AbstractAccessPoint , AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable, Referrable , ServerCallPoint			
Attribute	Type	Mult.	Kind	Note
calledFromWithinExclusiveArea	ExclusiveAreaNestingOrder	0..1	ref	This indicates that the call point is located at the deepest level inside one or more ExclusiveAreas that are nested in the given order.

Table B.126: SynchronousServerCallPoint

Class	System			
Package	M2::AUTOSARTemplates::SystemTemplate			





Class	System			
Note	<p>The top level element of the System Description. The System description defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints.</p> <p>The System element directly aggregates the elements describing the Software, Mapping and Mapping Constraints; it contains a reference to an ASAM FIBEX description specifying Communication and Topology.</p> <p>Tags:atp.recommendedPackage=Systems</p>			
Base	<p><i>ARElement, ARObject, AtpClassifier, AtpFeature, AtpStructureElement, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i></p>			
Attribute	Type	Mult.	Kind	Note
clientId DefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System.
containerIPdu HeaderByte Order	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus.
ecuExtract Version	RevisionLabelString	0..1	attr	Version number of the Ecu Extract.
fibexElement	FibexElement	*	ref	<p>Reference to ASAM FIBEX elements specifying Communication and Topology.</p> <p>All Fibex Elements used within a System Description shall be referenced from the System Element.</p> <p>atpVariation: In order to describe a product-line, all Fibex Elements can be optional.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
interpolation Routine MappingSet	InterpolationRoutine MappingSet	*	ref	This reference identifies the InterpolationRoutineMapping Sets that are relevant in the context of the enclosing System.
j1939Shared AddressCluster	J1939SharedAddress Cluster	*	aggr	<p>Collection of J1939Clusters that share a common address space for the routing of messages.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=j1939SharedAddressCluster.shortName, j1939SharedAddressCluster.variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
mapping	SystemMapping	*	aggr	<p>Aggregation of all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints).</p> <p>In order to support OEM / Tier 1 interaction and shared development for one common System this aggregation is atpSplitable and atpVariation. The content of System Mapping can be provided by several parties using different names for the SystemMapping.</p> <p>This element is not required when the System description is used for a network-only use-case.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=mapping.shortName, mapping.variation Point.shortLabel vh.latestBindingTime=postBuild</p>
pncVector Length	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).
pncVectorOffset	PositiveInteger	0..1	attr	Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.





Class	System			
rootSoftwareComposition	RootSwCompositionPrototype	0..1	aggr	<p>Aggregation of the root software composition, containing all software components in the System in a hierarchical structure. This element is not required when the System description is used for a network-only use-case.</p> <p>atpVariation: The RootSwCompositionPrototype can vary.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=rootSoftwareComposition.shortName, rootSoftwareComposition.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime</p>
swCluster	CpSoftwareCluster	*	ref	<p>CP Software Clusters of this System</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=swCluster.cpSoftwareCluster, swCluster.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=systemDesignTime</p>
systemDocumentation	Chapter	*	aggr	<p>Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=systemDocumentation.shortName, systemDocumentation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10</p>
systemVersion	RevisionLabelString	1	attr	Version number of the System Description.

Table B.127: System

Class	TimingEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is used to start RunnableEntities that shall be executed periodically.			
Base	<i>ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, RTEEvent, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
offset	TimeValue	0..1	attr	The value makes an assumption about the time offset of the first activation of the RunnableEntity triggered by the mapped TimingEvent relative to the periodic activation of the time base of this TimingEvent. Unit: second.
period	TimeValue	0..1	attr	Period of timing event in seconds. The value of this attribute shall be greater than zero.

Table B.128: TimingEvent

Class	Trigger			
Package	M2::AUTOSARTemplates::CommonStructure::TriggerDeclaration			
Note	A trigger which is provided (i.e. released) or required (i.e. used to activate something) in the given context.			
Base	<i>ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note





Class	Trigger			
swImplPolicy	SwImplPolicyEnum	0..1	attr	This attribute, when set to value queued, allows for a queued processing of Triggers.
triggerPeriod	MultidimensionalTime	0..1	aggr	Optional definition of a period in case of a periodically (time or angle) driven external trigger.

Table B.129: Trigger

Class	TriggerInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A trigger interface declares a number of triggers that can be sent by an trigger source. Tags: atp.recommendedPackage=PortInterfaces			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , PortInterface , Referrable			
Attribute	Type	Mult.	Kind	Note
trigger	Trigger	*	aggr	The Trigger of this trigger interface.

Table B.130: TriggerInterface

Class	Unit			
Package	M2::MSR::AsamHdo::Units			
Note	<p>This is a physical measurement unit. All units that might be defined should stem from SI units. In order to convert one unit into another factor and offset are defined.</p> <p>For the calculation from SI-unit to the defined unit the factor (factorSiToUnit) and the offset (offsetSiToUnit) are applied as follows:</p> $x \{unit\} := y * \{siUnit\} * factorSiToUnit \{unit\} / \{siUnit\} + offsetSiToUnit \{unit\}$ <p>For the calculation from a unit to SI-unit the reciprocal of the factor (factorSiToUnit) and the negation of the offset (offsetSiToUnit) are applied.</p> $y \{siUnit\} := (x * \{unit\} - offsetSiToUnit \{unit\}) / (factorSiToUnit \{unit\} / \{siUnit\})$ <p>Tags:atp.recommendedPackage=Units</p>			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
displayName	SingleLanguageUnit Names	0..1	aggr	This specifies how the unit shall be displayed in documents or in user interfaces of tools.The displayName corresponds to the Unit.Display in an ASAM MCD-2MC file. Tags: xml.sequenceOffset=20
factorSiToUnit	Float	0..1	attr	This is the factor for the conversion from SI Units to units. The inverse is used for conversion from units to SI Units. Tags: xml.sequenceOffset=30
offsetSiToUnit	Float	0..1	attr	This is the offset for the conversion from and to siUnits. Tags: xml.sequenceOffset=40
physical Dimension	PhysicalDimension	0..1	ref	This association represents the physical dimension to which the unit belongs to. Note that only values with units of the same physical dimensions might be converted. Tags: xml.sequenceOffset=50

Table B.131: Unit

Class	VariableAccess			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements			
Note	The presence of a VariableAccess implies that a RunnableEntity needs access to a VariableData Prototype. The kind of access is specified by the role in which the class is used.			
Base	ARObject, AbstractAccessPoint , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
accessed Variable	AutosarVariableRef	0..1	aggr	This denotes the accessed variable.
scope	VariableAccessScope Enum	0..1	attr	This attribute allows for constraining the scope of the corresponding communication. For example, it possible to express whether the communication is intended to cross the boundary of an ECU or whether it is intended not to cross the boundary of a single partition.

Table B.132: VariableAccess

Class	VariableDataPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	A VariableDataPrototype is used to contain values in an ECU application. This means that most likely a VariableDataPrototype allocates "static" memory on the ECU. In some cases optimization strategies might lead to a situation where the memory allocation can be avoided. In particular, the value of a VariableDataPrototype is likely to change as the ECU on which it is used executes.			
Base	ARObject, AtpFeature , AtpPrototype , AutosarDataPrototype , DataPrototype , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype

Table B.133: VariableDataPrototype

C Referenced ECUC Configuration Parameters

C.1 EcuC

C.1.1 EcucPartition

SWS Item	[ECUC_EcuC_00005]		
Container Name	EcucPartition		
Parent Container	EcucPartitionCollection		
Description	Definition of one Partition on this ECU. One Partition will be implemented using one Os-Application.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	–	
	Post-build time	–	

Configuration Parameters

Name	EcucDefaultBswPartition [ECUC_EcuC_00037]		
Parent Container	EcucPartition		
Description	Denotes the default BSW partition. This partition will host all BSW Modules, which are not explicitly mapped to a different partition. For partitions other than the default BSW partition this parameter can be omitted.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	PartitionCanBeRestarted [ECUC_EcuC_00006]		
Parent Container	EcucPartition		
Description	Specifies the requirement whether the Partition can be restarted. If set to true all software executing in this partition shall be capable of handling a restart.		
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	–	
Scope / Dependency			

Name	EcucEcuPartitionRef [ECUC_EcuC_00083]		
Parent Container	EcucPartition		
Description	Reference to the EcuPartition to define the link to the partition described in the System description. Tags: atp.Status=draft		

Multiplicity	0..1		
Type	Foreign reference to ECU-PARTITION		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	EcucPartitionBswModuleDistinguishedPartition [ECUC_EcuC_00068]		
Parent Container	EcucPartition		
Description	This maps the abstract partition of the Bsw Module to a concrete Partition existing in the ECU.		
Multiplicity	0..*		
Type	Foreign reference to BSW-DISTINGUISHED-PARTITION		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	EcucPartitionSoftwareComponentInstanceRef [ECUC_EcuC_00036]		
Parent Container	EcucPartition		
Description	References the SW Component instances from the Ecu Extract that shall be executed in this partition.		
Multiplicity	0..*		
Type	Instance reference to SW-COMPONENT-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

No Included Containers

C.1.2 Pdu

SWS Item	[ECUC_EcuC_00001]		
Container Name	Pdu		
Parent Container	EcucPduCollection		
Description	One Pdu flowing through the COM-Stack. This Pdu is used by all Com-Stack modules to agree on referencing the same Pdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Name	DynamicLength [ECUC_EcuC_00078]		
Parent Container	Pdu		
Description	This parameter defines whether the Pdu has dynamic length (true) or not (false). Please note that the usage of this attribute is restricted by [constr_3448].		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency			

Name	J1939Requestable [ECUC_EcuC_00072]		
Parent Container	Pdu		
Description	Pdu can be triggered by the J1939 request message.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency			

Name	PduLength [ECUC_EcuC_00003]		
Parent Container	Pdu		
Description	Length of the Pdu in bytes. It should be noted that in former AUTOSAR releases (Rel 2.1, Rel 3.0, Rel 3.1, Rel 4.0 Rev. 1) this parameter was defined in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency			

Name	EcucPduDefaultPartitionRef [ECUC_EcuC_00082]		
Parent Container	Pdu		
Description	Reference to EcucPartition, where the according Pdu is assigned to.		
Multiplicity	0..1		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	MetaDataTypeRef [ECUC_EcuC_00077]		
Parent Container	Pdu		
Description	Reference to meta data that is transported in the Pdu through the AUTOSAR layers.		
Multiplicity	0..1		
Type	Reference to MetaDataType		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	SysTPduToFrameTriggeringRef [ECUC_EcuC_00052]		
Parent Container	Pdu		
Description	Reference to the FrameTriggering from the SystemTemplate which this Pdu belongs to. SysTPduToFrameTriggeringRef shall be used for UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. This reference shall not be used if SysTPduToPduTriggeringRef exists.		
Multiplicity	0..1		
Type	Foreign reference to FRAME-TRIGGERING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	dependency: SysTPduToFrameTriggeringRef shall be used for UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. This reference shall not be used if SysTPduToPduTriggeringRef exists.		

Name	SysTPduToPduTriggeringRef [ECUC_EcuC_00054]		
Parent Container	Pdu		
Description	Reference to the PduTriggering from the SystemTemplate which this Pdu represents. SysTPduToPduTriggeringRef shall be used for all Pdus except UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. For these Pdus, SysTPduToFrameTriggeringRef shall be used.		
Multiplicity	0..1		
Type	Foreign reference to PDU-TRIGGERING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	dependency: SysTPduToPduTriggeringRef shall be used for all Pdus except UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. This reference shall not be used if SysTPduToFrameTriggeringRef exists.		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
EcucPduDedicatedPartition	0..*	Module specific container for Pdu to partition assignment.

C.1.3 EcucPduDedicatedPartition

SWS Item	[ECUC_EcuC_00079]		
Container Name	EcucPduDedicatedPartition		
Parent Container	Pdu		
Description	Module specific container for Pdu to partition assignment.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	EcucPduDedicatedPartitionBswModuleRef [ECUC_EcuC_00080]		
Parent Container	EcucPduDedicatedPartition		
Description	Reference to BSW module, for which the according dedicated Pdu assignment is valid.		
Multiplicity	1		
Type	Foreign reference to ECUC-MODULE-CONFIGURATION-VALUES		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	EcucPduDedicatedPartitionRef [ECUC_EcuC_00081]		
Parent Container	EcucPduDedicatedPartition		
Description	Module specific reference to EcucPartition, where the according Pdu is assigned to. The dedicated partition reference shall overrule the default partition reference for the respective module.		
Multiplicity	0..1		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

C.2 RTE

C.2.1 RteEventToTaskMapping

SWS Item	[ECUC_Rte_09020]
-----------------	------------------

Container Name	RteEventToTaskMapping
Parent Container	RteSwComponentInstance
Description	Maps an instance of a RunnableEntity onto one OsTask based on the activating RTEEvent. In the case of a RunnableEntity executed via a direct or trusted function call this RteEventToTaskMapping is still specified but no RteMappedToTask element is included. The RtePositionInTask parameter is necessary to provide an ordering of events invoked by the same RTE API.
Configuration Parameters	

Name	RteActivationOffset [ECUC_Rte_09018]		
Parent Container	RteEventToTaskMapping		
Description	Activation offset in seconds.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. INF]		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteImmediateRestart [ECUC_Rte_09092]		
Parent Container	RteEventToTaskMapping		
Description	When RteImmediateRestart is set to true the RunnableEntity shall be immediately re-started after termination if it was activated by this RTEEvent while it was already started. This parameter shall not be set to true when the mapped RTEEvent refers to a RunnableEntity which minimumStartInterval attribute is > 0.		
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		

Name	RteOsSchedulePoint [ECUC_Rte_09022]		
Parent Container	RteEventToTaskMapping		
Description	<p>Introduce a schedule point by explicitly calling Os Schedule service after the execution of the ExecutableEntity. The Rte generator is allowed to optimize several consecutive calls to Os schedule into one single call if the ExecutableEntity executions in between have been skipped.</p> <p>The absence of this parameter is interpreted as "NONE".</p> <p>It shall be considered an invalid configuration if the task is preemptable and the value of this parameter is not set to "NONE" or the parameter is absent.</p>		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	CONDITIONAL	A Schedule Point shall be introduced at the end of the execution of this ExecutableEntity. The Schedule Point can be skipped if several Schedule Points would be called without any ExecutableEntity execution in between.	
	NONE	No Schedule Point shall be introduced at the end of the execution of this ExecutableEntity.	
	UNCONDITIONAL	A Schedule Point shall always be introduced at the end of the execution of this ExecutableEntity.	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RtePositionInTask [ECUC_Rte_09023]		
Parent Container	RteEventToTaskMapping		
Description	<p>Each RunnableEntity mapped to an OsTask has a specific position within the task execution. For periodic activation this is the order of execution. For event driver activation this is the order of evaluation which actual RunnableEntity has to be executed. In case of direct or trusted function calls this parameter is necessary to provide an ordering of events when several ExecutableEntities are invoked by the same RTE API.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		

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

Name	RteServerQueueLength [ECUC_Rte_09133]		
Parent Container	RteEventToTaskMapping		
Description	Specifies the length of the queue for the server call serialization. This value overwrites the queueLength specified at the ServerComSpec.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteEventPredecessorSyncPointRef [ECUC_Rte_09128]		
Parent Container	RteEventToTaskMapping		
Description	<p>The RteEventPredecessorSyncPointRef is necessary to provide a cross core synchronization in case of RteEvents triggered by the same event source but mapped to tasks belonging to different partitions on different cores.</p> <p>The synchronization point must be reached by all referencing RteEvents before the execution in all related tasks is continued.</p> <p>In case of RteEventPredecessorSyncPointRef the RunnableEntity activated by the mapped RteEvent is executed after the synchronization point is passed.</p>		
Multiplicity	0..1		
Type	Reference to RteSyncPoint		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteEventRef [ECUC_Rte_09019]		
Parent Container	RteEventToTaskMapping		
Description	Reference to the description of the RTEEvent which is pointing to the RunnableEntity being mapped. This allows a fine grained mapping of RunnableEntites based on the activating RTEEvent.		
Multiplicity	1..*		
Type	Foreign reference to RTE-EVENT		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteEventSuccessorSyncPointRef [ECUC_Rte_09129]		
Parent Container	RteEventToTaskMapping		
Description	<p>The RteEventSuccessorSyncPointRef is necessary to provide a cross core synchronization in case of RteEvents triggered by the same event source but mapped to tasks belonging to different partitions on different cores.</p> <p>The synchronization point must be reached by all referencing RteEvents before the execution in all related tasks is continued.</p> <p>In case of RteEventSuccessorSyncPointRef the RunnableEntity activated by the mapped RteEvent is executed before the synchronization point is entered.</p>		
Multiplicity	0..1		
Type	Reference to RteSyncPoint		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteMappedToTaskRef [ECUC_Rte_09021]		
Parent Container	RteEventToTaskMapping		
Description	<p>Reference to the OsTask the RunnableEntity activated by the RteEventRef is mapped to.</p> <p>If no reference to the OsTask is specified the RunnableEntity shall be executed via a direct or trusted function call.</p> <p>The fact that no reference to an OsTask is specified for a RunnableEntity does not necessarily imply that every RTE generator has to support the implementation of this RunnableEntity as a direct or trusted function call. The standard set of use cases for direct or trusted function calls that has to be supported by every RTE generator is explicitly stated as requirements in this document. For further optimization RTE vendors are free to support additional scenarios of direct or trusted function call implementations that are not explicitly required in this document.</p>		
Multiplicity	0..1		
Type	Reference to OsTask		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteRipsFillRoutineRef [ECUC_Rte_89005]		
Parent Container	RteEventToTaskMapping		
Description	Reference to a Buffer-Fill Routine implemented by an RTE Implementation Plug-In. This routine gets invoked directly before the ExecutableEntity is started. Attributes: requiresIndex=true		
Multiplicity	0..*		
Type	Reference to destinationUri [RteRipsUriDefSet/RteRipsPluginFillFlush Routine]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteRipsFlushRoutineRef [ECUC_Rte_89006]		
Parent Container	RteEventToTaskMapping		
Description	Reference to a Buffer-Flush Routine implemented by an RTE Implementation Plug-In. This routine gets invoked directly after the ExecutableEntity has terminated. Attributes: requiresIndex=true		
Multiplicity	0..*		
Type	Reference to destinationUri [RteRipsUriDefSet/RteRipsPluginFillFlush Routine]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteRipsInvocationHandlerRef [ECUC_Rte_89008]		
Parent Container	RteEventToTaskMapping		
Description	Reference to a Buffer-Fill Routine implemented by an RTE Implementation Plug-In. This routine gets invoked directly before the ExecutableEntity is started.		
Multiplicity	0..1		
Type	Reference to destinationUri [RteRipsUriDefSet/RteRipsInvocation Handler]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteUsedInitFnc [ECUC_Rte_09116]		
Parent Container	RteEventToTaskMapping		
Description	The RunnableEntity is executed during initialization in the context of the Rte_Init_<InitContainer> function.		
Multiplicity	0..1		
Type	Reference to RteInitializationRunnableBatch		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteUsedOsAlarmRef [ECUC_Rte_09024]		
Parent Container	RteEventToTaskMapping		
Description	If an OsAlarm is used to activate the OsTask this RteEvent is mapped to it shall be referenced here.		
Multiplicity	0..1		
Type	Reference to OsAlarm		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteUsedOsEventRef [ECUC_Rte_09025]		
Parent Container	RteEventToTaskMapping		
Description	If an OsEvent is used to activate the OsTask this RteEvent is mapped to it shall be referenced here.		
Multiplicity	0..1		
Type	Reference to OsEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteUsedOsSchTblExpiryPointRef [ECUC_Rte_09026]		
Parent Container	RteEventToTaskMapping		
Description	If an OsScheduleTableExpiryPoint is used to activate the OsTask this RteEvent is mapped to it shall be referenced here.		
Multiplicity	0..1		
Type	Reference to OsScheduleTableExpiryPoint		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	RteVirtuallyMappedToTaskRef [ECUC_Rte_09027]		
Parent Container	RteEventToTaskMapping		
Description	Optional reference to an OsTask where the activation of this RteEvent shall be evaluated. The actual execution of the Runnable Entity shall happen in the OsTask referenced by RteMappedToTaskRef.		
Multiplicity	0..1		
Type	Reference to OsTask		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

C.3 Os

C.3.1 OsAlarm

SWS Item	[ECUC_Os_00003]
-----------------	-----------------

Container Name	OsAlarm
Parent Container	Os
Description	An OsAlarm may be used to asynchronously inform or activate a specific task. It is possible to start alarms automatically at system start-up depending on the application mode.
Configuration Parameters	

Name	OsAlarmAccessingApplication [ECUC_Os_00004]		
Parent Container	OsAlarm		
Description	Reference to applications which have an access to this object.		
Multiplicity	0..*		
Type	Reference to OsApplication		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	OsAlarmCounterRef [ECUC_Os_00005]		
Parent Container	OsAlarm		
Description	Reference to the assigned counter for that alarm		
Multiplicity	1		
Type	Reference to OsCounter		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
OsAlarmAction	1	This container defines which type of notification is used when the alarm expires.
OsAlarmAutostart	0..1	If present this container defines if an alarm is started automatically at system start-up depending on the application mode.

C.3.2 OsApplication

SWS Item	[ECUC_Os_00114]
Container Name	OsApplication
Parent Container	Os
Description	<p>An AUTOSAR OS must be capable of supporting a collection of OS objects (tasks, interrupts, alarms, hooks etc.) that form a cohesive functional unit. This collection of objects is termed an OS-Application.</p> <p>All objects which belong to the same OS-Application have access to each other. Access means to allow to use these objects within API services.</p> <p>Access by other applications can be granted separately.</p>
Configuration Parameters	

Name	OsTrusted [ECUC_Os_00115]		
Parent Container	OsApplication		
Description	<p>Parameter to specify if an OS-Application is trusted or not.</p> <p>true: OS-Application is trusted false: OS-Application is not trusted (default)</p>		
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	<p>scope: ECU</p> <p>dependency: Required for scalability class 3 and 4.</p>		

Name	OsTrustedApplicationDelayTimingViolationCall [ECUC_Os_00395]		
Parent Container	OsApplication		
Description	<p>Parameter to specify if a timing violation which occurs within an trusted OS-Application is raised immediately if it is delayed until the current task returns to the calling OS-Application (return of CallTrustedFunction) true: violation / call to ProtectionHook() is delayed false: timing violation cause an immediate call to the ProtectionHook().</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsTrustedApplicationWithProtection [ECUC_Os_00394]		
Parent Container	OsApplication		
Description	Parameter to specify if a trusted OS-Application is executed with memory protection or not. true: OS-Application runs within a protected environment. This means that write access is limited. false: OS-Application has full write access (default)		
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: ECU		

Name	OsAppAlarmRef [ECUC_Os_00231]		
Parent Container	OsApplication		
Description	Specifies the OsAlarms that belong to the OsApplication.		
Multiplicity	0..*		
Type	Reference to OsAlarm		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsAppCounterRef [ECUC_Os_00234]		
Parent Container	OsApplication		
Description	References the OsCounters that belong to the OsApplication.		
Multiplicity	0..*		
Type	Reference to OsCounter		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsAppEcucPartitionRef [ECUC_Os_00392]		
Parent Container	OsApplication		
Description	Denotes which "EcucPartition" is implemented by this "OSApplication".		
Multiplicity	0..1		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	OsApplsRRef [ECUC_Os_00221]		
Parent Container	OsApplication		
Description	references which OsIsrs belong to the OsApplication		
Multiplicity	0..*		
Type	Reference to OsIsr		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsApplicationCoreRef [ECUC_Os_00393]		
Parent Container	OsApplication		
Description	Reference to the Core Definition in the Ecuc Module where the CoreId is defined. This reference is used to describe to which Core the OsApplication is bound.		
Multiplicity	0..1		
Type	Reference to EcucCoreDefinition		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsAppScheduleTableRef [ECUC_Os_00230]		
Parent Container	OsApplication		
Description	References the OsScheduleTables that belong to the OsApplication.		
Multiplicity	0..*		
Type	Reference to OsScheduleTable		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsAppTaskRef [ECUC_Os_00116]		
Parent Container	OsApplication		
Description	references which OsTasks belong to the OsApplication		
Multiplicity	0..*		
Type	Reference to OsTask		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsMemoryMappingCodeLocationRef [ECUC_Os_00402]		
Parent Container	OsApplication		
Description	Reference to the memory mapping containing details about the section where the code is placed.		
Multiplicity	0..1		
Type	Foreign reference to SW-ADDR-METHOD		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsRestartTask [ECUC_Os_00120]		
Parent Container	OsApplication		
Description	<p>Optionally one task of an OS-Application may be defined as Restart Task.</p> <p>Multiplicity = 1: Restart Task is activated by the Operating System if the protection hook requests it.</p> <p>Multiplicity = 0: No task is automatically started after a protection error happened.</p>		
Multiplicity	0..1		
Type	Reference to OsTask		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: ECU dependency: Required for scalability class 3 and 4.
---------------------------	---

Included Containers		
Container Name	Multiplicity	Scope / Dependency
OsApplicationHooks	1	Container to structure the OS-Application-specific hooks
OsApplicationTrusted Function	0..*	Container to structure the configuration parameters of trusted functions

C.3.3 OsCounter

SWS Item	[ECUC_Os_00026]
Container Name	OsCounter
Parent Container	Os
Description	Configuration information for the counters that belong to the OsApplication.
Configuration Parameters	

Name	OsCounterMaxAllowedValue [ECUC_Os_00027]		
Parent Container	OsCounter		
Description	Maximum possible allowed value of the system counter in ticks.		
Multiplicity	1		
Type	EcuIntegerParamDef		
Range	1 .. 18446744073709551615		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsCounterMinCycle [ECUC_Os_00028]		
Parent Container	OsCounter		
Description	The MINCYCLE attribute specifies the minimum allowed number of counter ticks for a cyclic alarm linked to the counter.		
Multiplicity	1		
Type	EcuIntegerParamDef		
Range	1 .. 18446744073709551615		
Default Value			
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsCounterTicksPerBase [ECUC_Os_00029]		
Parent Container	OsCounter		
Description	The TICKSPERBASE attribute specifies the number of ticks required to reach a counterspecific unit. The interpretation is implementation-specific.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsCounterType [ECUC_Os_00255]		
Parent Container	OsCounter		
Description	This parameter contains the natural type or unit of the counter.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	HARDWARE	This counter is driven by some hardware e.g. a hardware timer unit.	
	SOFTWARE	The counter is driven by some software which calls the IncrementCounter service.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsSecondsPerTick [ECUC_Os_00030]		
Parent Container	OsCounter		
Description	Time of one counter tick in seconds.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default Value			

Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsCounterAccessingApplication [ECUC_Os_00031]		
Parent Container	OsCounter		
Description	Reference to applications which have an access to this object.		
Multiplicity	0..*		
Type	Reference to OsApplication		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
OsDriver	0..1	<p>This Container contains the information who will drive the counter. This configuration is only valid if the counter has OsCounterType set to HARDWARE.</p> <p>If the container does not exist (multiplicity=0) the timer is managed by the OS internally (OSINTERNAL).</p> <p>If the container exists the OS can use the GPT interface to manage the timer. The user have to supply the GPT channel.</p> <p>If the counter is driven by some other (external to the OS) source (like a TPU for example) this must be described as a vendor specific extension.</p>

OsTimeConstant	0..*	<p>Allows the user to define constants which can be e.g. used to compare time values with timer tick values.</p> <p>A time value will be converted to a timer tick value during generation and can later on accessed via the OsConstName. The conversation is done by rounding time values to the nearest fitting tick value.</p>
----------------	------	---

C.3.4 OsEvent

SWS Item	[ECUC_Os_00033]
Container Name	OsEvent
Parent Container	Os
Description	Representation of OS events in the configuration context. Adopted from the ISO 17356-6 specification.
Configuration Parameters	

Name	OsEventMask [ECUC_Os_00034]		
Parent Container	OsEvent		
Description	If event mask would be set to AUTO in OIL, this parameter should be omitted here.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 ..	18446744073709551615	
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

C.3.5 OsResource

SWS Item	[ECUC_Os_00252]
Container Name	OsResource

Parent Container	Os
Description	An OsResource object is used to co-ordinate the concurrent access by tasks and ISRs to a shared resource, e.g. the scheduler, any program sequence, memory or any hardware area.
Configuration Parameters	

Name	OsResourceProperty [ECUC_Os_00050]		
Parent Container	OsResource		
Description	This specifies the type of the resource.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	INTERNAL	The resource is an internal resource.	
	LINKED	The resource is a linked resource (a second name for a existing resource).	
	STANDARD	The resource is a standard resource.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsResourceAccessingApplication [ECUC_Os_00051]		
Parent Container	OsResource		
Description	Reference to applications which have an access to this object.		
Multiplicity	0..*		
Type	Reference to OsApplication		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsResourceLinkedResourceRef [ECUC_Os_00052]		
Parent Container	OsResource		
Description	The link to the resource. Must be valid if OsResourceProperty is LINKED. If OsResourceProperty is not LINKED the value is ignored.		
Multiplicity	0..1		
Type	Reference to OsResource		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

C.3.6 OsScheduleTable

SWS Item	[ECUC_Os_00141]
Container Name	OsScheduleTable
Parent Container	Os
Description	An OsScheduleTable addresses the synchronization issue by providing an encapsulation of a statically defined set of alarms that cannot be modified at runtime.
Configuration Parameters	

Name	OsScheduleTableDuration [ECUC_Os_00053]		
Parent Container	OsScheduleTable		
Description	This parameter defines the modulus of the schedule table (in ticks).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 ..		
	18446744073709551615		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsScheduleTableRepeating [ECUC_Os_00144]		
Parent Container	OsScheduleTable		
Description	true: first expiry point on the schedule table shall be processed at final expiry point delay ticks after the final expiry point is processed. false: the schedule table processing stops when the final expiry point is processed.		
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	–	
Scope / Dependency	scope: ECU		

Name	OsScheduleTableCounterRef [ECUC_Os_00145]		
Parent Container	OsScheduleTable		
Description	This parameter contains a reference to the counter which drives the schedule table.		
Multiplicity	1		
Type	Reference to OsCounter		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsSchTblAccessingApplication [ECUC_Os_00054]		
Parent Container	OsScheduleTable		
Description	Reference to applications which have an access to this object.		
Multiplicity	0..*		
Type	Reference to OsApplication		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: local
---------------------------	--------------

Included Containers		
Container Name	Multiplicity	Scope / Dependency
OsScheduleTable Autostart	0..1	This container specifies if and how the schedule table is started on startup of the Operating System. The options to start a schedule table correspond to the API calls to start schedule tables during runtime.
OsScheduleTableExpiryPoint	1..*	The point on a Schedule Table at which the OS activates tasks and/or sets events
OsScheduleTableSync	0..1	This container specifies the synchronization parameters of the schedule table.

C.3.7 OsScheduleTableExpiryPoint

SWS Item	[ECUC_Os_00143]
Container Name	OsScheduleTableExpiryPoint
Parent Container	OsScheduleTable
Description	The point on a Schedule Table at which the OS activates tasks and/or sets events
Configuration Parameters	

Name	OsScheduleTblExpPointOffset [ECUC_Os_00062]		
Parent Container	OsScheduleTableExpiryPoint		
Description	The offset from zero (in ticks) at which the expiry point is to be processed.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 ..	18446744073709551615	
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
OsScheduleTableEvent Setting	0..*	Event that is triggered by that schedule table.
OsScheduleTableTask Activation	0..*	Task that is triggered by that schedule table.
OsScheduleTbl AdjustableExpPoint	0..1	Adjustable expiry point

C.3.8 OsSpinlock

SWS Item	[ECUC_Os_00258]
Container Name	OsSpinlock
Parent Container	Os
Description	An OsSpinlock object is used to co-ordinate concurrent access by TASKs/ISR2s on different cores to a shared resource.
Configuration Parameters	

Name	OsSpinlockLockMethod [ECUC_Os_01038]		
Parent Container	OsSpinlock		
Description	Lock method which is used when a spinlock is taken. Note that it is possible that a user (e.g. a Task) might hold more than one spinlock. In this case the last lock taken is forced to use at least a lock method which locks as strong as the current one.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	LOCK_ALL_INTERRUPTS		
	LOCK_CAT2_INTERRUPTS		
	LOCK_NOTHING		
	LOCK_WITH_RESCHEDULER		
Default Value	LOCK_NOTHING		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsSpinlockAccessingApplication [ECUC_Os_01021]		
Parent Container	OsSpinlock		
Description	Reference to OsApplications that have an access to this object.		
Multiplicity	1..*		
Type	Reference to OsApplication		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: local
---------------------------	--------------

Name	OsSpinlockSuccessor [ECUC_Os_01022]		
Parent Container	OsSpinlock		
Description	<p>Reference to OsApplications that have an access to this object.</p> <p>To check whether a spinlock can be occupied (in a nested way) without any danger of deadlock, a linked list of spinlocks can be defined. A spinlock can only be occupied in the order of the linked list. It is allowed to skip a spinlock.</p> <p>If no linked list is specified, spinlocks cannot be nested.</p>		
Multiplicity	0..1		
Type	Reference to OsSpinlock		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

C.3.9 OsTask

SWS Item	[ECUC_Os_00073]
Container Name	OsTask
Parent Container	Os
Description	This container represents an ISO 17356 task.
Configuration Parameters	

Name	OsTaskActivation [ECUC_Os_00074]	
Parent Container	OsTask	
Description	<p>This attribute defines the maximum number of queued activation requests for the task. A value equal to "1" means that at any time only a single activation is permitted for this task. Note that the value must be a natural number starting at 1.</p>	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 4294967295	

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

Name	OsTaskPeriod [ECUC_Os_00404]		
Parent Container	OsTask		
Description	<p>This parameter specifies the period in seconds of this task in case of a cyclically activated task.</p> <p>If this parameter is not given the task can be activated sporadically or cyclically with a unknown period value.</p> <p>This value is information, e.g. for time base calculations in the RTE in case TimingEvents are mapped onto this OsTask.Be aware, that this parameter is not supposed to be relevant for the OS! This information is given as part of the OS configuration to support configuration work flows using a fixed set of OsTasks.</p>		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[-INF .. INF]		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsTaskPriority [ECUC_Os_00075]		
Parent Container	OsTask		
Description	<p>The priority of a task is defined by the value of this attribute. This value has to be understood as a relative value, i.e. the values show only the relative ordering of the tasks.</p> <p>ISO 17356-3 defines the lowest priority as zero (0); larger values correspond to higher priorities.</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsTaskSchedule [ECUC_Os_00076]		
Parent Container	OsTask		
Description	<p>The OsTaskSchedule attribute defines the preemptability of the task.</p> <p>If this attribute is set to NON, no internal resources may be assigned to this task.</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	FULL		Task is preemptable.
	NON		Task is not preemptable.
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsMemoryMappingCodeLocationRef [ECUC_Os_00402]		
Parent Container	OsTask		
Description	Reference to the memory mapping containing details about the section where the code is placed.		
Multiplicity	0..1		
Type	Foreign reference to SW-ADDR-METHOD		
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	OsTaskAccessingApplication [ECUC_Os_00077]		
Parent Container	OsTask		
Description	Reference to applications which have an access to this object.		
Multiplicity	0..*		
Type	Reference to OsApplication		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsTaskEventRef [ECUC_Os_00078]		
Parent Container	OsTask		
Description	This reference defines the list of events the extended task may react on.		
Multiplicity	0..*		
Type	Reference to OsEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	OsTaskResourceRef [ECUC_Os_00079]		
Parent Container	OsTask		
Description	This reference defines a list of resources accessed by this task.		
Multiplicity	0..*		
Type	Reference to OsResource		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
OsTaskAutostart	0..1	This container determines whether the task is activated during the system start-up procedure or not for some specific application modes. If the task shall be activated during the system start-up, this container is present and holds the references to the application modes in which the task is auto-started.
OsTaskTimingProtection	0..1	This container contains all parameters regarding timing protection of the task.

C.4 NvM

C.4.1 NvMBlockDescriptor

SWS Item	[ECUC_NvM_00061]
Container Name	NvMBlockDescriptor
Parent Container	NvM
Description	Container for a management structure to configure the composition of a given NVRAM Block Management Type. Its multiplicity describes the number of configured NVRAM blocks, one block is required to be configured. The NVRAM block descriptors are condensed in the NVRAM block descriptor table.
Configuration Parameters	

Name	NvMBlockCrcType [ECUC_NvM_00476]		
Parent Container	NvMBlockDescriptor		
Description	Defines CRC data width for the NVRAM block. Default: NVM_CRC16, i.e. CRC16 will be used if NVM_BLOCK_USE_CRC==true		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	NVM_CRC16		(Default) CRC16 will be used if NVM_BLOCK_USE_CRC==true.
	NVM_CRC32		CRC32 is selected for this NVRAM block if NVM_BLOCK_USE_CRC==true.
	NVM_CRC8		CRC8 is selected for this NVRAM block if NVM_BLOCK_USE_CRC==true.
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local dependency: NVM_BLOCK_USE_CRC, NVM_CALC_RAM_BLOCK_CRC		

Name	NvMBlockHeaderInclude [ECUC_NvM_00554]		
Parent Container	NvMBlockDescriptor		
Description	Defines the header file where the owner of the NVRAM block has the declarations of the permanent RAM data block, ROM data block (if configured) and the callback function prototype for each configured callback. If no permanent RAM block, ROM block or callback functions are configured then this configuration parameter shall be ignored.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMBlockJobPriority [ECUC_NvM_00477]		
Parent Container	NvMBlockDescriptor		
Description	Defines the job priority for a NVRAM block (0 = Immediate priority).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMBlockManagementType [ECUC_NvM_00062]		
Parent Container	NvMBlockDescriptor		
Description	Defines the block management type for the NVRAM block.[SWS_NvM_00137]		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	NVM_BLOCK_DATASET	NVRAM block is configured to be of dataset type.	
	NVM_BLOCK_NATIVE	NVRAM block is configured to be of native type.	
	NVM_BLOCK_REDUNDANT	NVRAM block is configured to be of redundant type.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMBlockUseAutoValidation [ECUC_NvM_00557]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines whether the RAM Block shall be auto validated during shutdown phase.</p> <p>true: if auto validation mechanism is used, false: otherwise</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

Name	NvMBlockUseCompression [ECUC_NvM_00563]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines whether the data is compressed before written. true: data compression activated (takes more time to read and write) false: no compression</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

Name	NvMBlockUseCrc [ECUC_NvM_00036]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines CRC usage for the NVRAM block, i.e. memory space for CRC is reserved in RAM and NV memory.</p> <p>true: CRC will be used for this NVRAM block. false: CRC will not be used for this NVRAM block.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMBlockUseCRCCompMechanism [ECUC_NvM_00556]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines whether the CRC of the RAM Block shall be compared during a write job with the CRC which was calculated during the last successful read or write job.</p> <p>true: if compare mechanism is used, false: otherwise</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local dependency: False if NvMBlockUseCrc = False		

Name	NvMBlockUsePort [ECUC_NvM_00559]		
Parent Container	NvMBlockDescriptor		
Description	<p>If this parameter is true it defines whether:</p> <ul style="list-style-type: none"> the port with interface 'NvMMirror' for synchronization mechanism callbacks are generated if the parameter NvMBlockUseSyncMechanism is configured TRUE; the port with interface 'NvMNotifyInitBlock' for initialization block callback is generated if NvMInitBlockCallback parameter is configured (independent of the content); the port with interface 'NvMNotifyJobFinished' for single block callback is generated if NvMSingleBlockCallback parameter is configured (independent of the content); the port with interface 'NvMAdmin' for SetBlockProtection operation is generated. 		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMBlockUseSetRamBlockStatus [ECUC_NvM_00552]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines if NvMSetRamBlockStatusApi shall be used for this block or not.</p> <p>Note: If NvMSetRamBlockStatusApi is disabled this configuration parameter shall be ignored.</p> <p>true: calling of NvMSetRamBlockStatus for this RAM block shall set the status of the RAM block.</p> <p>false: calling of NvMSetRamBlockStatus for this RAM block shall be ignored.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMBlockUseSyncMechanism [ECUC_NvM_00519]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines whether an explicit synchronization mechanism with a RAM mirror and callback routines for transferring data to and from NvM module's RAM mirror is used for NV block. true if synchronization mechanism is used, false otherwise.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMBlockWriteProt [ECUC_NvM_00033]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines an initial write protection of the NV block</p> <p>true: Initial block write protection is enabled. false: Initial block write protection is disabled.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMBswMBlockStatusInformation [ECUC_NvM_00551]		
Parent Container	NvMBlockDescriptor		
Description	<p>This parameter specifies whether BswM is informed about the current status of the specified block.</p> <p>True: Call BswM_NvM_CurrentBlockMode on changes False: Don't inform BswM at all</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMCalcRamBlockCrc [ECUC_NvM_00119]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines CRC (re)calculation for the permanent RAM block or NVRAM blocks which are configured to use explicit synchronization mechanism.</p> <p>true: CRC will be (re)calculated for this permanent RAM block. false: CRC will not be (re)calculated for this permanent RAM block.</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local dependency: NVM_BLOCK_USE_CRC		

Name	NvMMaxNumOfReadRetries [ECUC_NvM_00533]		
Parent Container	NvMBlockDescriptor		
Description	Defines the maximum number of read retries.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default Value	0		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMMaxNumOfWriteRetries [ECUC_NvM_00499]		
Parent Container	NvMBlockDescriptor		
Description	Defines the maximum number of write retries for a NVRAM block with [ECUC_NvM_00061]. Regardless of configuration a consistency check (and maybe write retries) are always forced for each block which is processed by the request NvM_WriteAll and NvM_WriteBlock.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMNvBlockBaseNumber [ECUC_NvM_00478]		
Parent Container	NvMBlockDescriptor		
Description	<p>Configuration parameter to perform the link between the NVM_NVRAM_BLOCK_IDENTIFIER used by the SW-Cs and the FEE_BLOCK_NUMBER expected by the memory abstraction modules. The parameter value equals the FEE_BLOCK_NUMBER or EA_BLOCK_NUMBER shifted to the right by NvMDatasetSelectionBits bits. (ref. to chapter 7.1.2.1).</p> <p>Calculation Formula: value = TargetBlockReference.[Ea/Fee]BlockConfiguration.[Ea/Fee]BlockNumber » NvMDatasetSelectionBits</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 65534		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local dependency: FEE_BLOCK_NUMBER, EA_BLOCK_NUMBER		

Name	NvMNvBlockLength [ECUC_NvM_00479]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines the NV block data length in bytes.</p> <p>Note: The implementer can add the attribute 'withAuto' to the parameter definition which indicates that the length can be calculated by the generator automatically (e.g. by using a parser that searches and analyzes the data structure corresponding to the block). When 'withAuto' is set to 'true' for this parameter definition the 'isAutoValue' can be set to 'true'. If 'isAutoValue' is set to 'true' the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards.</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMNvBlockNum [ECUC_NvM_00480]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines the number of multiple NV blocks in a contiguous area according to the given block management type.</p> <p>1-255 For NVRAM blocks to be configured of block management type NVM_BLOCK_DATASET. The actual range is limited according to SWS_NvM_00444.</p> <p>1 For NVRAM blocks to be configured of block management type NVM_BLOCK_NATIVE</p> <p>2 For NVRAM blocks to be configured of block management type NVM_BLOCK_REDUNDANT</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local dependency: NVM_BLOCK_MANAGEMENT_TYPE		

Name	NvMNvramBlockIdentifier [ECUC_NvM_00481]		
Parent Container	NvMBlockDescriptor		
Description	<p>Identification of a NVRAM block via a unique block identifier.</p> <p>Implementation Type: NvM_BlockIdType.</p> <p>min = 2 max = 2¹⁶ - NVM_DATASET_SELECTION_BITS - 1</p> <p>Reserved NVRAM block IDs: 0 -> to derive multi block request results via NvM_GetErrorStatus 1 -> redundant NVRAM block which holds the configuration ID (generation tool should check that this block is correctly configured from type,CRC and size point of view)</p>		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	2 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: local dependency: NVM_DATASET_SELECTION_BITS		

Name	NvMNvramDeviceId [ECUC_NvM_00035]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines the NVRAM device ID where the NVRAM block is located.</p> <p>Calculation Formula: value = TargetBlockReference.[Ea/Fee]DeviceIndex</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 1		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local dependency: EA_DEVICE_INDEX, FEE_DEVICE_INDEX		

Name	NvMRamBlockDataAddress [ECUC_NvM_00482]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines the start address of the RAM block data.</p> <p>If this is not configured, no permanent RAM data block is available for the selected block management type.</p>		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMReadRamBlockFromNvCallback [ECUC_NvM_00521]		
Parent Container	NvMBlockDescriptor		
Description	<p>Entry address of a block specific callback routine which shall be called in order to let the application copy data from the NvM module's mirror to RAM block. Implementation type: Std_ReturnType</p> <p>E_OK: copy was successful E_NOT_OK: copy was not successful, callback routine to be called again</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

Name	NvMResistantToChangedSw [ECUC_NvM_00483]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines whether a NVRAM block shall be treated resistant to configuration changes or not. If there is no default data available at configuration time then the application shall be responsible for providing the default initialization data. In this case the application has to use NvM_GetErrorStatus() to be able to distinguish between first initialization and corrupted data.</p> <p>true: NVRAM block is resistant to changed software. false: NVRAM block is not resistant to changed software.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

Name	NvMRomBlockDataAddress [ECUC_NvM_00484]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines the start address of the ROM block data.</p> <p>If not configured, no ROM block is available for the selected block management type.</p>		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

Name	NvMRomBlockNum [ECUC_NvM_00485]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines the number of multiple ROM blocks in a contiguous area according to the given block management type.</p> <p>0-254 For NVRAM blocks to be configured of block management type NVM_BLOCK_DATASET. The actual range is limited according to SWS_NvM_00444.</p> <p>0-1 For NVRAM blocks to be configured of block management type NVM_BLOCK_NATIVE</p> <p>0-1 For NVRAM blocks to be configured of block management type NVM_BLOCK_REDUNDANT</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 254		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local dependency: NVM_BLOCK_MANAGEMENT_TYPE, NVM_NV_BLOCK_NUM		

Name	NvMSelectBlockForFirstInitAll {NVM_SELECT_BLOCK_FOR_FIRST_INIT_ALL} [ECUC_NvM_00558]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines whether a block will be processed or not by NvM_FirstInitAll. A block can be configured to be processed even if it doesn't have permanent RAM and/or explicit synchronization.</p> <p>TRUE: block will be processed by NvM_FirstInitAll</p> <p>FALSE: block will not be processed by NvM_FirstInitAll</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMSelectBlockForReadAll [ECUC_NvM_00117]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines whether a NVRAM block shall be processed during NvM_ReadAll or not. This configuration parameter has only influence on those NVRAM blocks which are configured to have a permanent RAM block or which are configured to use explicit synchronization mechanism.</p> <p>true: NVRAM block shall be processed by NvM_ReadAll false: NVRAM block shall not be processed by NvM_ReadAll</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Scope / Dependency	scope: local dependency: NVM_RAM_BLOCK_DATA_ADDRESS
---------------------------	--

Name	NvMSelectBlockForWriteAll [ECUC_NvM_00549]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines whether a NVRAM block shall be processed during NvM_WriteAll or not. This configuration parameter has only influence on those NVRAM blocks which are configured to have a permanent RAM block or which are configured to use explicit synchronization mechanism.</p> <p>true: NVRAM block shall be processed by NvM_WriteAll false: NVRAM block shall not be processed by NvM_WriteAll</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local dependency: NVM_RAM_BLOCK_DATA_ADDRESS		

Name	NvMStaticBlockIDCheck [ECUC_NvM_00532]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines if the Static Block ID check is enabled.</p> <p>false: Static Block ID check is disabled. true: Static Block ID check is enabled.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMWriteBlockOnce [ECUC_NvM_00072]		
Parent Container	NvMBlockDescriptor		
Description	<p>Defines write protection after first write. The NVRAM manager sets the write protection bit either after the NV block was written the first time or if the block was already written and it is detected as valid and consistent during a read for it.</p> <p>true: Defines write protection after first write is enabled.</p> <p>false: Defines write protection after first write is disabled.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMWriteRamBlockToNvCallback [ECUC_NvM_00520]		
Parent Container	NvMBlockDescriptor		
Description	<p>Entry address of a block specific callback routine which shall be called in order to let the application copy data from RAM block to NvM module's mirror. Implementation type: Std_ReturnType</p> <p>E_OK: copy was successful E_NOT_OK: copy was not successful, callback routine to be called again</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	NvMWriteVerification [ECUC_NvM_00534]		
Parent Container	NvMBlockDescriptor		
Description	Defines if Write Verification is enabled. false: Write verification is disabled. true: Write Verification is enabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

Name	NvMWriteVerificationDataSize [ECUC_NvM_00538]		
Parent Container	NvMBlockDescriptor		
Description	Defines the number of bytes to compare in each step when comparing the content of a RAM Block and a block read back.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

Name	NvMBlockCipherringRef [ECUC_NvM_00567]		
Parent Container	NvMBlockDescriptor		
Description	Reference to cipherring container. If configured, NvM encrypt the data before storage and decrypt the data after restoring. If empty, the NvM stores and restore the original user data. Tags: atp.Status=draft		
Multiplicity	0..1		
Type	Reference to NvMBlockCipherring		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: Key will be located in RAM if this configuration item is not present.		

Name	NvMBlockEcucPartitionRef [ECUC_NvM_00564]		
Parent Container	NvMBlockDescriptor		
Description	Maps the NV block to zero or one ECUC partition to limit the access to this NV block. The ECUC partition referenced is within the subset of the ECUC partitions where the NvM is mapped to.		
Multiplicity	0..1		
Type	Reference to EcucPartition		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
NvMInitBlockCallback	0..1	<p>The presence of this container indicates, that a block specific callback routine is called if no ROM data is available for initialization of the NVRAM block. If the container is not present, no callback routine is called for initialization of the NVRAM block with default data.</p> <p>In case the container has a NvMInitBlockCallbackFnc, the NvM will call this function.</p> <p>In case there is no NvMInitBlockCallbackFnc, the NvM will have an port PNIB_{Block}.</p>
NvMSingleBlockCallback	0..1	<p>The presence of this container indicates, that the block specific callback routine which shall be invoked on termination of each asynchronous single block request [SWS_NvM_00113] If the container is not present, no callback routine is called..</p> <p>In case the container has a NvMSingleBlockCallbackFnc, the NvM will call this function.</p> <p>In case there is no NvMSingleBlockCallbackFnc, the NvM will have an port PNJF_{Block}.</p>

NvMTargetBlock Reference	1	This parameter is just a container for the parameters for EA and FEE
--------------------------	---	--

C.4.2 NvMInitBlockCallback

SWS Item	[ECUC_NvM_00561]		
Container Name	NvMInitBlockCallback		
Parent Container	NvMBlockDescriptor		
Description	<p>The presence of this container indicates, that a block specific callback routine is called if no ROM data is available for initialization of the NVRAM block. If the container is not present, no callback routine is called for initialization of the NVRAM block with default data.</p> <p>In case the container has a NvMInitBlockCallbackFnc, the NvM will call this function.</p> <p>In case there is no NvMInitBlockCallbackFnc, the NvM will have an port PNIB_{Block}.</p>		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Configuration Parameters			

Name	NvMInitBlockCallbackFnc [ECUC_NvM_00116]		
Parent Container	NvMInitBlockCallback		
Description	<p>Entry address of a block specific callback routine which shall be called if no ROM data is available for initialization of the NVRAM block.</p> <p>If not configured, no specific callback routine shall be called for initialization of the NVRAM block with default data.</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

C.4.3 NvMSingleBlockCallback

SWS Item	[ECUC_NvM_00562]		
Container Name	NvMSingleBlockCallback		
Parent Container	NvMBlockDescriptor		
Description	<p>The presence of this container indicates, that the block specific callback routine which shall be invoked on termination of each asynchronous single block request [SWS_NvM_00113] If the container is not present, no callback routine is called..</p> <p>In case the container has a NvMSingleBlockCallbackFnc, the NvM will call this function.</p> <p>In case there is no NvMSingleBlockCallbackFnc, the NvM will have an port PNJF_{Block}.</p>		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Configuration Parameters			

Name	NvMSingleBlockCallbackFnc [ECUC_NvM_00506]		
Parent Container	NvMSingleBlockCallback		
Description	Entry address of the block specific callback routine which shall be invoked on termination of each asynchronous single block request [SWS_NvM_00113].		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

C.5 Dcm

C.5.1 DcmDspDid

SWS Item	[ECUC_Dcm_00601]
Container Name	DcmDspDid
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) of the DID.
Configuration Parameters	

Name	DcmDspDidIdentifier [ECUC_Dcm_00602]		
Parent Container	DcmDspDid		
Description	2 byte Identifier of the DID Within each DcmConfigSet all DcmDspDidIdentifier values shall be unique.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidSize [ECUC_Dcm_01099]		
Parent Container	DcmDspDid		
Description	Length of a DID in byte(s).		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidUsed [ECUC_Dcm_00805]		
Parent Container	DcmDspDid		
Description	Allow to activate or deactivate the usage of a DID, for multi purpose ECUs true = DID available false = DID not available		
Multiplicity	1		
Type	EcucBooleanParamDef		
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: ECU		

Name	DcmDspDidUsePort [ECUC_Dcm_01122]		
Parent Container	DcmDspDid		
Description	Selects application interface type for DID data elements between a single operation for all data elements or data element specific operations.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	USE_ATOMIC_BNDM	The DID shall be read/written from/to BndM.	
	USE_ATOMIC_NV_DATA_INTERFACE	A single sender receiver interface with NvData-Ports is used for all data elements of this DID.	
	USE_ATOMIC_SENDER_RECEIVER_INTERFACE	A single sender receiver interface is used to access all data elements of this DID. The sender receiver interface is typed with IsService=false.	
	USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE	A single sender receiver interface is used to access all data elements of this DID. The sender receiver interface is typed with IsService=true.	
	USE_DATA_ELEMENT_SPECIFIC_INTERFACES	The data elements of this DID are collected by using the data element specific interfaces defined by DcmDspDataUsePort.	
Default Value	USE_DATA_ELEMENT_SPECIFIC_INTERFACES		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency			

Name	DcmDspDidBndMBlockIdRef [ECUC_Dcm_01185]		
Parent Container	DcmDspDid		
Description	Associate this DID with a BndM blockId.		
Multiplicity	0..1		
Type	Symbolic name reference to BndMBlockDescriptor		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU dependency: DcmDspDidUsePort == USE_ATOMIC_BNDM		

Name	DcmDspDidInfoRef [ECUC_Dcm_00604]		
Parent Container	DcmDspDid		
Description	Reference to DcmDspDidInfo containing information on this DID.		
Multiplicity	1		
Type	Reference to DcmDspDidInfo		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRef [ECUC_Dcm_00606]		
Parent Container	DcmDspDid		
Description	Reference to DcmDspDid in case this DID refer to one or several other DID's Attributes: requiresIndex=true		
Multiplicity	0..*		
Type	Reference to DcmDspDid		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDidSignal	0..*	This container defines the reference to 1 DcmDspData container and position relevant for this DID.
DcmDspDidSupportInfo	0..1	This container defines the support information to declare the usability of the data bytes within the DIDs

C.5.2 DcmDspDidInfo

SWS Item	[ECUC_Dcm_00607]
Container Name	DcmDspDidInfo
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) of the DID's Info
Configuration Parameters	

Name	DcmDspDDDIDMaxElements [ECUC_Dcm_00970]		
Parent Container	DcmDspDidInfo		
Description	Maximum number of source elements of a DDDID.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDidDynamicallyDefined [ECUC_Dcm_00612]		
Parent Container	DcmDspDidInfo		
Description	Indicates if this DID can be dynamically defined true = DID can be dynamically defined false = DID can not be dynamically defined		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDidControl	0..1	This container contains the configuration (parameters) of the DID control.
DcmDspDidRead	0..1	This container contains the configuration (parameters) of the DID read.
DcmDspDidWrite	0..1	This container contains the configuration (parameters) of the DID write.

C.5.3 DcmDspData

SWS Item	[ECUC_Dcm_00869]
Container Name	DcmDspData
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) of a Data belonging to a DID
Configuration Parameters	

Name	DcmDspDataByteSize [ECUC_Dcm_01106]	
Parent Container	DcmDspData	
Description	Defines the array length in bytes or the the maximum array length for variable datalengths.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default Value		
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataConditionCheckReadFnc [ECUC_Dcm_00677]		
Parent Container	DcmDspData		
Description	Function name to demand application if the conditions (e.g. System state) to read the DID are correct. (ConditionCheckRead-function). Multiplicity shall be equal to parameter DcmDspDataReadFnc. This parameter is related to the interface Xxx_ConditionCheckRead.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataConditionCheckReadFncUsed [ECUC_Dcm_00955]		
Parent Container	DcmDspData		
Description	This parameter determines if a condition check function is available or not.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDataEcuSignal [ECUC_Dcm_00825]		
Parent Container	DcmDspData		
Description	Function name to control the access to a certain ECU Signal by the DCM. (IoHwAb_Dcm_<symbolic name of ECU signal>-function).		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataEndianness [ECUC_Dcm_00986]	
Parent Container	DcmDspData	
Description	Defines the endianness of the data belonging to a DID in a diagnostic request or response message.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address.
	OPAQUE	Opaque data endianness
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDataFreezeCurrentStateFnc [ECUC_Dcm_00674]		
Parent Container	DcmDspData		
Description	Function name to request to application to freeze the current state of an IOControl. (FreezeCurrentState-function). This parameter is related to the interface Xxx_FreezeCurrentState.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataGetScalingInfoFnc [ECUC_Dcm_00676]		
Parent Container	DcmDspData		
Description	Function name to request to application the scaling information of the DID. (GetScalingInformation-function). This parameter is related to the interface Xxxx_GetScalingInformation.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataReadDataLengthFnc [ECUC_Dcm_00671]		
Parent Container	DcmDspData		
Description	Function name to request from application the data length of a DID. (ReadDataLength-function). This parameter is related to the interface Xxx_ReadDataLength.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataReadEcuSignal [ECUC_Dcm_00824]		
Parent Container	DcmDspData		
Description	Function name for read access to a certain ECU Signal by the DCM. (IoHwAb_Dcm_Read<EcuSignalName>-function). Only relevant if DcmDspDataUsePort==USE_ECU_SIGNAL.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataReadFnc [ECUC_Dcm_00669]		
Parent Container	DcmDspData		
Description	Function name to request from application the data value of a DID. (ReadData-function). This parameter is related to the interface Xxx_ReadData.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDataResetToDefaultFnc [ECUC_Dcm_00673]		
Parent Container	DcmDspData		
Description	Function name to request to application to reset an IOControl to default value. (ResetToDefault-function). This parameter is related to the interface Xxx_ResetToDefault.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataReturnControlToEcuFnc [ECUC_Dcm_00672]		
Parent Container	DcmDspData		
Description	Function name to request to application to return control to ECU of an IOControl. (ReturnControlToECU-function). This parameter is related to the interface Xxx_ReturnControlToECU.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataShortTermAdjustmentFnc [ECUC_Dcm_00675]		
Parent Container	DcmDspData		
Description	Function name to request to application to adjust the IO signal. (ShortTermAdjustment-function). This parameter is related to the interface Xxx_ShortTermAdjustment.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataType [ECUC_Dcm_00985]		
Parent Container	DcmDspData		
Description	Provide the implementation data type of data belonging to a DID.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BOOLEAN	Type of the data is boolean.	
	FLOAT	Type of the data is float.	
	FLOAT_N	Type of the data is float array.	
	SINT16	Type of the data is sint16.	
	SINT16_N	Type of the data is sint16 array.	
	SINT32	Type of the data is sint32.	
	SINT32_N	Type of the data is sint32 array.	
	SINT8	Type of the data is sint8.	
	SINT8_N	Type of the data is sint8 array.	
	UINT16	Type of the data is uint16.	
	UINT16_N	Type of the data is uint16 array.	
	UINT32	Type of the data is uint32.	
	UINT32_N	Type of the data is uint32 array.	
	UINT8	Type of the data is uint8.	
	UINT8_DYN	Type of the data is uint8 array with dynamic length.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataUsePort [ECUC_Dcm_00713]		
Parent Container	DcmDspData		
Description	Defines which interface shall be used to access the data.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	USE_BLOCK_ID	The DCM will access the Data using the NVRAM Apis with the BlockId defined in DcmDspDataBlockId	

	USE_DATA_ASYNC_CLIENT_SERVER	The DCM will access the Data using an R-Port requiring a asynchronous ClientServerInterface DataServices_{Data}. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.
	USE_DATA_ASYNC_CLIENT_SERVER_ERROR	The Dcm will access the Data using an R-Port requiring a asynchronous ClientServerInterface DataServices_{Data}. The parameter ErrorCode can be returned to allow the application to trigger a negative response during the operation. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.
	USE_DATA_ASYNC_FUNC	The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return is allowed. OpStatus is existing as IN parameter.
	USE_DATA_ASYNC_FUNC_ERROR	The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return is allowed. OpStatus is existing as IN parameter. The parameter ErrorCode can be returned to allow the application to trigger a negative response during the operation.
	USE_DATA_ASYNC_FUNC_PROXY	The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (with DataLength and ErrorCode parameter). DCM_E_PENDING return is allowed. OpStatus is existing as IN parameter. Tags: atp.Status=draft

	USE_DATA_SENDER_RECEIVER	The DCM will access the Data using an Port requiring a SenderReceiverInterface (with isService=false) DataServices_{Data}. The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.	
	USE_DATA_SENDER_RECEIVER_AS_SERVICE	The DCM will access the Data using an service Port requiring a SenderReceiverInterface (with isService=true) DataServices_{Data} . The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.	
	USE_DATA_SYNC_CLIENT_SERVER	The DCM will access the Data using an R-Port requiring a synchronous ClientServerInterface DataServices_{Data}. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.	
	USE_DATA_SYNC_FUNC	The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return value is not allowed and OpStatus parameter is not existing in the prototype.	
	USE_DATA_SYNC_FUNC_PROXY	The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (with DataLength and ErrorCode parameter). DCM_E_PENDING return value is not allowed and OpStatus parameter is not existing in the prototype. Tags: atp.Status=draft	
	USE_ECU_SIGNAL	The DCM will access the Data using a direct access to IoHwAb	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	

Scope / Dependency	scope: ECU
---------------------------	------------

Name	DcmDspDataWriteFnc [ECUC_Dcm_00670]		
Parent Container	DcmDspData		
Description	Function name to request application to write the data value of a DID. (WriteData-function). This parameter is related to the interface Xxx_WriteData.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspOdxDataDescription [ECUC_Dcm_00988]		
Parent Container	DcmDspData		
Description	Defines additional description for ODX documentation		
Multiplicity	0..1		
Type	EcucAddInfoParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDataBlockIdRef [ECUC_Dcm_00809]		
Parent Container	DcmDspData		
Description	<p>NRAM blockId to access the data.</p> <p>Only relevant if DcmDspDataUsePort==USE_BLOCK_ID.</p>		
Multiplicity	0..1		
Type	Symbolic name reference to NvMBlockDescriptor		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataInfoRef [ECUC_Dcm_00811]		
Parent Container	DcmDspData		
Description	Reference to 1 DcmDspDataInfo		
Multiplicity	0..1		
Type	Reference to DcmDspDataInfo		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDiagnosisScaling	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.
DcmDspDidDataSupportInfo	0..1	This container defines the supported information.
DcmDspExternalSRDataElementClass	0..1	<p>This container defines the source of data in a provided port which shall be read respectively the target of data in a required port which shall be written.</p> <p>This container shall contain either one DcmSubElementInDataElementInstance OR DcmDataElementInstance OR DcmSubElementInImplDataElementInstance reference.</p>

C.5.4 DcmDspPidService01

SWS Item	[ECUC_Dcm_00894]
Container Name	DcmDspPidService01
Parent Container	DcmDspPidData
Description	Contains specific configuration parameter of PID for service \$01. This container exists only if DcmDspPidService is set to DCM_SERVICE_01 or DCM_SERVICE_01_02.
Configuration Parameters	

Name	DcmDspPidDataEndianness [ECUC_Dcm_01012]		
Parent Container	DcmDspPidService01		
Description	Defines the endianness of the data belonging to a PID in a diagnostic response message. If no DcmDspPidDataEndianness is defined the value of DcmDspDataDefaultEndianness is applicable.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN		Most significant byte shall be stored at the lowest address.
	LITTLE_ENDIAN		Most significant byte shall be stored at the highest address
	OPAQUE		Opaque data endianness
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidDataReadFnc [ECUC_Dcm_00629]		
Parent Container	DcmDspPidService01		
Description	Function name for reading PID data value. This is only relevant if DcmDspPidDataUsePort==USE_DATA_SYNCH_FNC. This parameter is related to the interface Xxx_ReadData.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidDataType [ECUC_Dcm_01018]		
Parent Container	DcmDspPidService01		
Description	Provide the implementation data type of data belonging to a PID.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BOOLEAN	Type of the data is boolean.	
	SINT16	Type of the data is sint16.	
	SINT16_N	Type of the data is sint16 array.	
	SINT32	Type of the data is sint32.	
	SINT32_N	Type of the data is sint32 array.	
	SINT8	Type of the data is sint8.	
	SINT8_N	Type of the data is sint8 array.	
	UINT16	Type of the data is uint16.	
	UINT16_N	Type of the data is uint16 array.	
	UINT32	Type of the data is uint32.	
	UINT32_N	Type of the data is uint32 array.	
	UINT8	Type of the data is uint8.	
	UINT8_N	Type of the data is uint8 array.	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidDataUsePort [ECUC_Dcm_00720]		
Parent Container	DcmDspPidService01		
Description	<p>If this parameter is set to USE_DATA_SYNCH_FNC, the Dcm will use the function defined in DcmDspPidDataReadFnc to get the PID data value.</p> <p>If this parameter is set to USE_DATA_SYNCH_CLIENT_SERVER, the Dcm will have an R-Port requiring the interface DataServices_{Data}.</p> <p>If this parameter is set to USE_DATA_SENDER_RECEIVER or USE_DATA_SENDER_RECEIVER_AS_SERVICE, the DCM will have an R-Port requiring a SenderReceiverInterface.</p> <p>The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspPidData.</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	USE_DATA_SENDER_RECEIVER		
	USE_DATA_SENDER_RECEIVER_AS_SERVICE		
	USE_DATA_SYNCH_CLIENT_SERVER		
	USE_DATA_SYNCH_FNC		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDiagnosis Scaling	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.
DcmDspPidService01 ExternalSRDataElement Class	0..1	<p>This container defines the source of data in a provided port which shall be read respectively the target of data in a required port which shall be written.</p> <p>This container shall contain either one DcmSubElementInDataElementInstance OR DcmDataElementInstance OR DcmSubElementInImplDataElementInstance reference.</p>

C.5.5 DcmDspDidRead

SWS Item	[ECUC_Dcm_00613]
Container Name	DcmDspDidRead
Parent Container	DcmDspDidInfo

Description	This container contains the configuration (parameters) of the DID read.
Configuration Parameters	

Name	DcmDspDidReadModeRuleRef [ECUC_Dcm_00917]		
Parent Container	DcmDspDidRead		
Description	Reference to DcmModeRule Mode rule which controls to read this DID. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidReadRoleRef [ECUC_Dcm_01141]		
Parent Container	DcmDspDidRead		
Description	Reference to DcmDspAuthenticationRow that defines a role in that this DID can be read.		
Multiplicity	0..32		
Type	Reference to DcmDspAuthenticationRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidReadSecurityLevelRef [ECUC_Dcm_00614]		
Parent Container	DcmDspDidRead		
Description	Reference to DcmDspSecurityRow Referenced security levels are allowed to read this DID. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidReadSessionRef [ECUC_Dcm_00615]		
Parent Container	DcmDspDidRead		
Description	Reference to DcmDspSessionRow Referenced sessions are allowed to read this DID. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

C.5.6 DcmDspDidWrite

SWS Item	[ECUC_Dcm_00616]
-----------------	------------------

Container Name	DcmDspDidWrite
Parent Container	DcmDspDidInfo
Description	This container contains the configuration (parameters) of the DID write.
Configuration Parameters	

Name	DcmDspDidWriteModeRuleRef [ECUC_Dcm_00922]		
Parent Container	DcmDspDidWrite		
Description	Reference to DcmModeRule Mode rule which controls to write this DID. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidWriteRoleRef [ECUC_Dcm_01142]		
Parent Container	DcmDspDidWrite		
Description	Reference to DcmDspAuthenticationRow that defines a role in that this DID can be written.		
Multiplicity	0..32		
Type	Reference to DcmDspAuthenticationRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidWriteSecurityLevelRef [ECUC_Dcm_00617]		
Parent Container	DcmDspDidWrite		
Description	Reference to DcmDspSecurityRow Referenced security levels are allowed to write this DID. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidWriteSessionRef [ECUC_Dcm_00618]		
Parent Container	DcmDspDidWrite		
Description	Reference to DcmDspSessionRow Referenced sessions are allowed to write this DID. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

C.5.7 DcmDspDidControl

SWS Item	[ECUC_Dcm_00619]
-----------------	------------------

Container Name	DcmDspDidControl
Parent Container	DcmDspDidInfo
Description	This container contains the configuration (parameters) of the DID control.
Configuration Parameters	

Name	DcmDspDidControlMask [ECUC_Dcm_01059]		
Parent Container	DcmDspDidControl		
Description	This indicates the presence of "controlEnableMask" in SWC service interfaces and defines how the Dcm treats a service request.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DCM_CONTROLMASK_EXTERNAL	The control enable mask record shall be forwarded within each interface and is handled externally.	
	DCM_CONTROLMASK_INTERNAL	The control enable mask record is handled internally and Dcm controls only the included signals.	
	DCM_CONTROLMASK_NO	No control enable mask handling.	
Default Value	DCM_CONTROLMASK_INTERNAL		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlMaskSize [ECUC_Dcm_01060]		
Parent Container	DcmDspDidControl		
Description	The value defines the size of the controlEnableMaskRecord in bytes.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967294		
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		

Name	DcmDspDidFreezeCurrentState [ECUC_Dcm_00624]		
Parent Container	DcmDspDidControl		
Description	This indicates the presence of "FreezeCurrentState".		
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	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidResetToDefault [ECUC_Dcm_00623]		
Parent Container	DcmDspDidControl		
Description	This indicates the presence of "ResetToDefault".		
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	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidShortTermAdjustment [ECUC_Dcm_00625]		
Parent Container	DcmDspDidControl		
Description	This indicates the presence of "ShortTermAdjustment".		
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	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlModeRuleRef [ECUC_Dcm_00923]		
Parent Container	DcmDspDidControl		
Description	Reference to DcmModeRule Mode rule which controls this DID. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlRoleRef [ECUC_Dcm_01143]		
Parent Container	DcmDspDidControl		
Description	Reference to DcmDspAuthenticationRow that defines a role in that this IO can be controlled.		
Multiplicity	0..32		
Type	Reference to DcmDspAuthenticationRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlSecurityLevelRef [ECUC_Dcm_00620]		
Parent Container	DcmDspDidControl		
Description	Reference to DcmDspSecurityRow Security levels allowed to control this DID. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlSessionRef [ECUC_Dcm_00621]		
Parent Container	DcmDspDidControl		
Description	Reference to DcmDspSessionRow Sessions allowed to control this DID. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDidControl EnableMask	0..32	The shortname of the container value defines the symbol of the controlMask.

C.6 Dem

C.6.1 DemEventParameter

SWS Item	[ECUC_Dem_00661]
Container Name	DemEventParameter
Parent Container	DemConfigSet
Description	This container contains the configuration (parameters) for events.
Configuration Parameters	

Name	DemCausalityDelayTime [ECUC_Dem_00921]		
Parent Container	DemEventParameter		
Description	Time to wait until the event is considered as causal. The parameter is specified in seconds.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 2.5]		
Default Value	0		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DemComponentPriority [ECUC_Dem_00909]		
Parent Container	DemEventParameter		
Description	Specifies the priority within the component. A lower value means higher priority.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DemEventAssociatedIdentification [ECUC_Dem_00969]		
Parent Container	DemEventParameter		
Description	Event associated identifier that allows to identify an event. This value can be reported as internal data element in snapshot records or extended data records.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DemEventAvailable [ECUC_Dem_00792]		
Parent Container	DemEventParameter		
Description	This parameter configures an Event as unavailable. It is treated by Dem as if it does not exist. true = Event is available false = Event is not available		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DemEventAvailableInVariant [ECUC_Dem_00968]		
Parent Container	DemEventParameter		
Description	This parameter defines the enabled or disabled status of an event. If set to FALSE, the Event is generally disabled, even though the SW-C is integrated and the Event is described in the ECUC. The Dem shall ignore all calls from the SW-C. If set to TRUE, the availability of the event depends on other availability sources.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DemEventConfirmationThreshold [ECUC_Dem_00924]		
Parent Container	DemEventParameter		
Description	Defines the operation cycle threshold of the DTC confirmation status according "Confirmation Threshold" of ISO 14229-1.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value	1		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DemEventConfirmationThresholdCounterAdaptable [ECUC_Dem_00929]		
Parent Container	DemEventParameter		
Description	Indicates whether the events confirmation Cycle threshold can be adapted by Dem_SetEventConfirmationThresholdCounter.		
Multiplicity	0..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: ECU		

Name	DemEventId [ECUC_Dem_00659]		
Parent Container	DemEventParameter		
Description	<p>Unique identifier of a diagnostic event.</p> <p>This parameter should not be changeable by user, because the Id should be generated by Dem itself to prevent gaps and multiple use of an Id. The events should be sequentially ordered beginning with 1 and no gaps in between.</p>		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	1 .. 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		

Name	DemEventKind [ECUC_Dem_00660]		
Parent Container	DemEventParameter		
Description	This parameter is used to distinguish between SW-C and BSW events.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DEM_EVENT_KIND_BSW		The event is a assigned to a BSW module
	DEM_EVENT_KIND_SWC		The event is a assigned to a SW-C
Default Value	DEM_EVENT_KIND_SWC		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	DemEventRecoverableInSameOperationCycle [ECUC_Dem_00916]		
Parent Container	DemEventParameter		
Description	If parameter is configured to FALSE, reporting of PASSED will be ignored if the event is already "testfailed this operation cycle".		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

Name	DemEventReportingType [ECUC_Dem_00970]		
Parent Container	DemEventParameter		
Description	<p>This parameter defines the way a monitor can report an event to the Dem. There are various ways to report an event and the Dem provides different APIs for it.</p> <p>Tags: atp.Status=draft</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IMMEDIATE_FAIL_AND_SYNCHRONOUS_PROCESSING	The event is reported over the API Dem_SetEventFailedWithSyncFreezeFrame.	
	STANDARD_REPORTING	The event is reported over the API Dem_SetEventStatus.	
	STANDARD_REPORTING_WITH_MONITOR_DATA	The event is reported over the API Dem_SetEventStatusWithMonitorData.	
Default Value	STANDARD_REPORTING		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DemFFPrestorageInNvm [ECUC_Dem_00948]		
Parent Container	DemEventParameter		
Description	<p>If the event uses a pre-stored freeze-frame this attribute indicates if the event requires the pre-stored data to be stored in non-volatile memory. TRUE = store the pre-stored data in non-volatile memory, FALSE = pre-stored data is not stored in non-volatile memory.</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: local
---------------------------	--------------

Name	DemFFPrestorageSupported [ECUC_Dem_00671]		
Parent Container	DemEventParameter		
Description	If this parameter is set to true, then the Prestorage of FreezeFrames is supported by the assigned event. This parameter is useful to calculate the buffer size.		
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: ECU		

Name	DemReportBehavior [ECUC_Dem_00894]		
Parent Container	DemEventParameter		
Description	Indicates the reporting behavior of the BSW Module (DemEventKind == DEM_EVENT_KIND_BSW) in order to determine the size of the reporting queue. If the parameter is not defined it means REPORT_BEFORE_INIT.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	REPORT_AFTER_INIT	Indicates that the Event will not be reported before Dem_Init().	
	REPORT_BEFORE_INIT	Indicates that the Event may be reported before Dem_Init().	
Default Value	REPORT_BEFORE_INIT		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DemComponentClassRef [ECUC_Dem_00908]		
Parent Container	DemEventParameter		
Description	Reference to the monitored component.		
Multiplicity	0..1		
Type	Reference to DemComponent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DemDTCRef [ECUC_Dem_00888]		
Parent Container	DemEventParameter		
Description	<p>This parameter defines the DTC configuration (typically Uds) associated with the diagnostic event.</p> <p>It is allowed to have events without a DTC (e.g. for ECU-internal events triggering safety reactions without being reported via diagnostic communication). The same DemDTCAttributes can be used from several events, to combine these (refer to chapter "Combination of diagnostic event").</p>		
Multiplicity	0..1		
Type	Reference to DemDTC		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	DemEnableConditionGroupRef [ECUC_Dem_00746]		
Parent Container	DemEventParameter		
Description	References an enable condition group.		
Multiplicity	0..1		
Type	Reference to DemEnableConditionGroup		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DemOBDDGroupingAssociativeEventsRef [ECUC_Dem_00839]		
Parent Container	DemEventParameter		
Description	This parameter defines a reference which points to a representative event of one group of associate events. The "reverence event" must refer to it self. Note: One event is only allowed to be reverenced to only one group of associate events.		
Multiplicity	0..1		
Type	Reference to DemEventParameter		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DemOperationCycleRef [ECUC_Dem_00702]		
Parent Container	DemEventParameter		
Description	Kind of operation cycle for the event (e.g. power cycle, driving cycle, ...)		
Multiplicity	1		
Type	Reference to DemOperationCycle		
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DemStorageConditionGroupRef [ECUC_Dem_00769]		
Parent Container	DemEventParameter		
Description	References a storage condition group.		
Multiplicity	0..1		
Type	Reference to DemStorageConditionGroup		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DemCallbackClearEvent Allowed	0..1	<p>The presence of this container indicates that the Dem has access to a "ClearEventAllowed" callback.</p> <p>In case there is a DemCallbackClearEventAllowedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackClearEventAllowedFnc, the Dem will have an R-Port requiring the interface CallbackClearEventAllowed whose name is generated by using the unique callback-prefix followed by the event name.</p>
DemCallbackEventData Changed	0..1	<p>The presence of this container indicates that the Dem has access to an "EventDataChanged" callback.</p> <p>In case there is a DemCallbackEventDataChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackEventDataChangedFnc, the Dem will have an R-Port requiring the interface CallbackEventDataChanged whose name is generated by using the unique callback-prefix followed by the event name.</p>

DemCallbackEventSyncStorageProcessed	0..1	<p>The presence of this container indicates that the Dem will call the configured callback after an event status reported with Dem_SetEventFailedWithSyncFreezeFrame is finished and the Dem is ready to process a next call for this event.</p> <p>Tags: atp.Status=draft</p>
DemCallbackEventUdsStatusChanged	0..*	<p>The presence of this container indicates, that the Dem has access to an "EventUdsStatusChanged" callback, which the Dem will call to notify other components about the change in the status of an event.</p> <p>In case there is a DemCallbackEventUdsStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackEventUdsStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackEventUdsStatusChanged, whose name is generated by using the unique callback-prefix followed by the event name.</p>
DemCallbackInitMForE	0..1	<p>The presence of this container indicates, that the Dem has access to an "InitMonitorForEvent" callback, which the Dem will call to initialize a monitor.</p> <p>In case the container has a DemCallbackInitMForEFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackInitMForEFnc, the Dem will have an R-Port requiring the interface CallbackInitMonitorForEvent, whose name is generated by using the unique callback-prefix followed by the event name.</p>
DemCallbackMonitorStatusChanged	0..*	<p>The presence of this container indicates, that the Dem has access to an "MonitorStatusChanged" callback, which the Dem will call to notify other components about the change in the status of an event.</p> <p>In case there is a DemCallbackMonitorStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackMonitorStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackMonitorStatusChanged, whose name is generated by using the unique callback-prefix followed by the event name.</p>

DemDebounceAlgorithm Class	1	Debounce algorithm class: counter based, time based, or monitor internal.
DemIndicatorAttribute	0..255	This container contains the event specific configuration of Indicators.

C.6.2 DemDataElementClass

SWS Item	[ECUC_Dem_00610]
Container Name	DemDataElementClass
Parent Container	DemGeneral
Description	This container contains the configuration (parameters) for an internal/external data element class.
Configuration Parameters	

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DemExternalCSData ElementClass	0..1	This container contains the configuration (parameters) for an external client/server based data element class. It defines, how the Dem can obtain the value of the data element from either a SW-C or another BSW module. Whether a client/server port or a C function-call is used, is defined by DemDataElementUsePort.
DemExternalSRData ElementClass	0..1	This container contains the configuration (parameters) for an external sender/receiver based data element class. It defines, how the Dem can obtain the value of the data element from a SW-C, by using a sender/receiver port.
DemInternalDataElement Class	0..1	This container contains the configuration (parameters) for an internal data element class.

C.7 Com

C.7.1 ComMainFunctionTx

SWS Item	[ECUC_Com_10014]		
Container Name	ComMainFunctionTx		
Parent Container	ComConfig		
Description	Each element of this container defines one instance of Com_MainFunctionTx.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	ComMainTxTimeBase [ECUC_Com_10010]		
Parent Container	ComMainFunctionTx		
Description	<p>The period between successive calls to according instance of Com_MainFunctionTx in seconds. This parameter may be used by the COM generator to transform the values of the reception related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.</p> <p>The COM module (generator) may rely on the fact that Com_MainFunctionTx is scheduled according to the value configured here.</p>		
Multiplicity	1		
Type	EcucFloatParamDef		
Range]0 .. INF[
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComPreparationNotification [ECUC_Com_10020]		
Parent Container	ComMainFunctionTx		
Description	<p>This callback function indicates that the signals/signal groups to be sent via a dedicated Com_MainFunctionTx instance will now be prepared for transmission.</p> <p>If this parameter is omitted no notification shall take place.</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComMainTxPartitionRef [ECUC_Com_10019]		
Parent Container	ComMainFunctionTx		
Description	Reference to EcucPartition, where the according Com_MainFunction instance is assigned to.		
Multiplicity	1		
Type	Reference to EcucPartition		
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

C.7.2 ComMainFunctionRx

SWS Item	[ECUC_Com_10011]		
Container Name	ComMainFunctionRx		
Parent Container	ComConfig		
Description	Each element of this container defines one instance of Com_MainFunctionRx.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	ComMainRxTimeBase [ECUC_Com_10018]		
Parent Container	ComMainFunctionRx		
Description	<p>The period between successive calls to according instance of Com_MainFunctionRx in seconds. This parameter may be used by the COM generator to transform the values of the reception related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.</p> <p>The COM module (generator) may rely on the fact that Com_MainFunctionRx is scheduled according to the value configured here.</p>		
Multiplicity	1		
Type	EcucFloatParamDef		
Range]0 .. INF[
Default Value			

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

Name	ComMainRxPartitionRef [ECUC_Com_10017]		
Parent Container	ComMainFunctionRx		
Description	Reference to EcucPartition, where the according Com_MainFunction instance is assigned to.		
Multiplicity	1		
Type	Reference to EcucPartition		
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

C.7.3 ComGroupSignal

SWS Item	[ECUC_Com_00520]		
Container Name	ComGroupSignal		
Parent Container	ComSignalGroup		
Description	This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.		
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			

Name	ComBitPosition [ECUC_Com_00259]		
Parent Container	ComGroupSignal		
Description	Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
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		

Name	ComBitSize [ECUC_Com_00158]		
Parent Container	ComGroupSignal		
Description	Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 64		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComHandleId [ECUC_Com_00165]		
Parent Container	ComGroupSignal		
Description	<p>The numerical value used as the ID.</p> <p>This ID identifies signals and signal groups in the COM APIs using Com_SignalIdType or Com_SignalGroupIdType parameter respectively.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	ComSignalDataInvalidValue [ECUC_Com_00391]		
Parent Container	ComGroupSignal		
Description	<p>Defines the data invalid value of the signal.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity 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 dependency: In case of UINT8_N the length of ComSignalDataInvalidValue has to be the same as ComSignalLength.		

Name	ComSignalEndianness [ECUC_Com_00157]		
Parent Container	ComGroupSignal		
Description	Defines the endianness of the signal's network representation.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN		
	LITTLE_ENDIAN		
	OPAQUE		
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		

Name	ComSignalInitValue [ECUC_Com_00170]		
Parent Container	ComGroupSignal		
Description	<p>Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value	0		
Regular Expression			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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 dependency: In case of UINT8_N the length of ComSignalInitValue has to be the same as ComSignalLength.		

Name	ComSignalLength [ECUC_Com_00437]		
Parent Container	ComGroupSignal		
Description	<p>Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.</p> <p>The supported maximum length is restricted by the used transportation system. For non TP-PDUs the maximum size of a PDU, and therefore also of any included signal, is limited by the concrete bus characteristic. For example, the limit is 8 bytes for CAN and LIN, 64 bytes for CAN FD and 254 for FlexRay.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComSignalType [ECUC_Com_00127]		
Parent Container	ComGroupSignal		
Description	The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BOOLEAN		
	FLOAT32		
	FLOAT64		
	SINT16		
	SINT32		
	SINT64		
	SINT8		
	UINT16		
	UINT32		
	UINT64		
	UINT8		

Post-Build Variant Value	UINT8_DYN		
	UINT8_N 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		

Name	ComTimeoutSubstitutionValue [ECUC_Com_10006]		
Parent Container	ComGroupSignal		
Description	<p>The signal substitution value will be used in case of a timeout and ComRxDataTimeoutAction is set to SUBSTITUTE. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00.</p> <p>In case of UINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0 (lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD

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		

Name	ComTransferProperty [ECUC_Com_00560]		
Parent Container	ComGroupSignal		
Description	Optionally defines whether this group signal shall contribute to the TRIGGERED_ON_CHANGE transfer property of the signal group. If at least one group signal of a signal group has the "ComTransferProperty" configured all other group signals of that signal group shall have the attribute configured as well.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	PENDING	A change of the value of this group signal shall not be considered in the evaluation of the signal groups ComTransferProperty.	
	TRIGGERED_ON_CHANGE	A change of the value of this group signal shall be considered in the evaluation of the signal groups ComTransferProperty.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComSystemTemplateSystemSignalRef [ECUC_Com_00002]		
Parent Container	ComGroupSignal		
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.		
Multiplicity	0..1		
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		

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

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComFilter	0..1	This container contains the configuration parameters of the AUTOSAR COM module's Filters. Note: On sender side the container is used to specify the transmission mode conditions.

C.7.4 ComIPdu

SWS Item	[ECUC_Com_00340]		
Container Name	ComIPdu		
Parent Container	ComConfig		
Description	Contains the configuration parameters of the AUTOSAR COM module's I-PDUs.		
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			

Name	ComIPduCallout [ECUC_Com_00387]
Parent Container	ComIPdu
Description	This parameter defines the existence and the name of a callout function for the corresponding I-PDU. If this parameter is omitted no I-PDU callout shall take place for the corresponding I-PDU.
Multiplicity	0..1
Type	EcucFunctionNameDef
Default Value	
Regular Expression	
Post-Build Variant Multiplicity	false
Post-Build Variant Value	false

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComIPduCancellationSupport [ECUC_Com_00709]		
Parent Container	ComIPdu		
Description	<p>Defines for I-PDUs with ComIPduType NORMAL: If the underlying IF-modul supports cancellation of transmit requests.</p> <p>Defines for I-PDUs with ComIPduType TP: If the underlying TP-module supports RX and TX cancellation of ongoing requests.</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU dependency: This parameter shall not be set to true if ComCancellationSupport is set to false		

Name	ComIPduDirection [ECUC_Com_00493]	
Parent Container	ComIPdu	
Description	The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	RECEIVE	
	SEND	
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 dependency: If configured to Sent also a ComTxIpdu container shall be included, see ECUC_Com_00496		

Name	ComIPduHandleId [ECUC_Com_00175]		
Parent Container	ComIPdu		
Description	The numerical value used as the ID of this I-PDU. The ComIPduHandleId is required by the API calls Com_RxIndication, Com_TpRxIndication, Com_StartOfReception and Com_CopyRxData to receive I-PDUs from the PduR (ComIP-duDirection: Receive), as well as the PduId passed to an Rx-I-PDU-callout. For Tx-I-PDUs (ComIPduDirection: Send), this handle Id is used for the APIs calls Com_TxConfirmation, Com_TriggerTransmit, Com_TriggerIPDUSend or Com_TriggerIPDUSendWithMetaData, Com_CopyTxData and Com_TpTxConfirmation to transmit respectively confirm transmissions of I-PDUs, as well as the PduId passed to the Tx-I-PDU-callout configured with ComIPduCallout and/or ComIPduTriggerTransmitCallout.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: ECU		

Name	ComIPduSignalProcessing [ECUC_Com_00119]	
Parent Container	ComIPdu	
Description	For the definition of the two modes Immediate and Deferred.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEFERRED	signal indication / confirmations are deferred for example to a cyclic task
	IMMEDIATE	the signal indications / confirmations are performed in Com_RxIndication/ Com_TxConfirmation

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		

Name	ComIPduTriggerTransmitCallout [ECUC_Com_00765]		
Parent Container	ComIPdu		
Description	If there is a trigger transmit callout defined for this I-PDU this parameter contains the name of the callout function.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComIPduType [ECUC_Com_00761]		
Parent Container	ComIPdu		
Description	Defines if this I-PDU is a normal I-PDU that can be sent unfragmented or if this is a large I-PDU that shall be sent via the Transport Protocol of the underlying bus.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	NORMAL	sent or received via normal L-PDU	
	TP	sent or received via TP	
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		

Name	ComIPduGroupRef [ECUC_Com_00206]		
Parent Container	ComIPdu		
Description	Reference to the I-PDU groups this I-PDU belongs to.		
Multiplicity	0..*		
Type	Reference to ComIPduGroup		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComIPduMainFunctionRef [ECUC_Com_10012]		
Parent Container	ComIPdu		
Description	Reference to the Com_MainFunctionRx/Com_MainFunctionTx this I-PDU belongs to. Mandatory, if multiple main functions of the relevant type are defined.		
Multiplicity	0..1		
Type	Choice reference to [ComMainFunctionRx, ComMainFunctionTx]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComIPduSignalGroupRef [ECUC_Com_00519]		
Parent Container	ComIPdu		
Description	References to all signal groups contained in this I-Pdu		
Multiplicity	0..*		
Type	Reference to ComSignalGroup		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		

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

Name	ComIPduSignalRef [ECUC_Com_00518]		
Parent Container	ComIPdu		
Description	References to all signals contained in this I-PDU.		
Multiplicity	0..*		
Type	Reference to ComSignal		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComMainFunctionRouteSignalsRef [ECUC_Com_10021]		
Parent Container	ComIPdu		
Description	Reference to ComMainFunctionRouteSignals which performs signal gateway related activities.		
Multiplicity	0..1		
Type	Reference to ComMainFunctionRouteSignals		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComPduIdRef [ECUC_Com_00711]		
Parent Container	ComIPdu		
Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
Multiplicity	1		
Type	Reference to Pdu		
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			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComTxIPdu	0..1	This container contains additional transmission related configuration parameters of the AUTOSAR COM module's I-PDUs.

C.7.5 ComSignal

SWS Item	[ECUC_Com_00344]		
Container Name	ComSignal		
Parent Container	ComConfig		
Description	Contains the configuration parameters of the AUTOSAR COM module's signals.		
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			

Name	ComBitPosition [ECUC_Com_00259]		
Parent Container	ComSignal		
Description	Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
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		

Name	ComBitSize [ECUC_Com_00158]		
Parent Container	ComSignal		
Description	Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 64		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComDataInvalidAction [ECUC_Com_00314]		
Parent Container	ComSignal		
Description	This parameter defines the action performed upon reception of an invalid signal. Relating to signal groups the action in case if one of the included signals is an invalid signal. If Replace is used the ComSignalInitValue will be used for the replacement.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	NOTIFY		
	REPLACE		Literal for DataInvalidAction
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComErrorNotification [ECUC_Com_00499] (Obsolete)		
Parent Container	ComSignal		
Description	Only valid on sender side: Name of Com_CbkTxErr callback function to be called. If this parameter is omitted no error notification shall take place. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComFirstTimeout [ECUC_Com_00183]		
Parent Container	ComSignal		
Description	Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 3600]		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		

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

Name	ComHandleId [ECUC_Com_00165]		
Parent Container	ComSignal		
Description	<p>The numerical value used as the ID.</p> <p>This ID identifies signals and signal groups in the COM APIs using Com_SignalIdType or Com_SignalGroupIdType parameter respectively.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	ComInitialValueOnly [ECUC_Com_00811]		
Parent Container	ComSignal		
Description	<p>This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the users (e.g. RTE, SwCluC). Thus the Com implementation does not need to expect any API calls for this signal (group).</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity 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		

Name	ComInvalidNotification [ECUC_Com_00315] (Obsolete)		
Parent Container	ComSignal		
Description	<p>Only valid on receiver side: Name of Com_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAction is configured to NOTIFY.</p> <p>Tags: atp.Status=obsolete</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComNotification [ECUC_Com_00498] (Obsolete)		
Parent Container	ComSignal		
Description	<p>On sender side: Name of Com_CbkTxAck callback function to be called. On receiver side: Name of Com_CbkRxAck callback function to be called.</p> <p>If this parameter is omitted no notification shall take place.</p> <p>Tags: atp.Status=obsolete</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComRxDataTimeoutAction [ECUC_Com_00412]		
Parent Container	ComSignal		
Description	This parameter defines the action performed upon expiration of the reception deadline monitoring timer.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	NONE	no replacement shall take place	
	REPLACE	signals shall be replaced by their ComSignalInitValue	
	SUBSTITUTE	signals shall be replaced by their ComTimeoutSubstitutionValue	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComSignalDataInvalidValue [ECUC_Com_00391]		
Parent Container	ComSignal		
Description	<p>Defines the data invalid value of the signal.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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 dependency: In case of UINT8_N the length of ComSignalDataInvalidValue has to be the same as ComSignalLength.		

Name	ComSignalEndianness [ECUC_Com_00157]		
Parent Container	ComSignal		
Description	Defines the endianness of the signal's network representation.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN		
	LITTLE_ENDIAN		
	OPAQUE		
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		

Name	ComSignalInitValue [ECUC_Com_00170]		
Parent Container	ComSignal		
Description	<p>Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value	0		
Regular Expression			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD

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 dependency: In case of UINT8_N the length of ComSignalInitValue has to be the same as ComSignalLength.		

Name	ComSignalLength [ECUC_Com_00437]		
Parent Container	ComSignal		
Description	<p>Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.</p> <p>The supported maximum length is restricted by the used transportation system. For non TP-PDUs the maximum size of a PDU, and therefore also of any included signal, is limited by the concrete bus characteristic. For example, the limit is 8 bytes for CAN and LIN, 64 bytes for CAN FD and 254 for FlexRay.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComSignalType [ECUC_Com_00127]		
Parent Container	ComSignal		
Description	The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BOOLEAN		
	FLOAT32		

Post-Build Variant Value	FLOAT64		
	SINT16		
	SINT32		
	SINT64		
	SINT8		
	UINT16		
	UINT32		
	UINT64		
	UINT8		
	UINT8_DYN		
	UINT8_N 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		

Name	ComTimeout [ECUC_Com_00263]		
Parent Container	ComSignal		
Description	Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by ECUC_Com_00183.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 3600]		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComTimeoutNotification [ECUC_Com_00552] (Obsolete)		
Parent Container	ComSignal		
Description	<p>On sender side: Name of Com_CbkTxTOut callback function to be called. On receiver side: Name of Com_CbkRxTOut callback function to be called.</p> <p>Tags: atp.Status=obsolete</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComTimeoutSubstitutionValue [ECUC_Com_10006]		
Parent Container	ComSignal		
Description	<p>The signal substitution value will be used in case of a timeout and ComRxDataTimeoutAction is set to SUBSTITUTE. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00.</p> <p>In case ofUINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComTransferProperty [ECUC_Com_00232]		
Parent Container	ComSignal		
Description	Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	PENDING	A write access to this signal never triggers the transmission of the corresponding I-PDU.	
	TRIGGERED	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.	
	TRIGGERED_ON_CHANGE	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.	
	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.	
	TRIGGERED_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComUpdateBitPosition [ECUC_Com_00257]		
Parent Container	ComSignal		
Description	<p>Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.</p> <p>Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComSystemTemplateSystemSignalRef [ECUC_Com_00002]		
Parent Container	ComSignal		
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.		
Multiplicity	0..1		
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComFilter	0..1	This container contains the configuration parameters of the AUTOSAR COM module's Filters. Note: On sender side the container is used to specify the transmission mode conditions.

C.7.6 ComUserModule

SWS Item	[ECUC_Com_10031]		
Container Name	ComUserModule		
Parent Container	ComConfig		
Description	Contains the configuration parameters of the Com user modules. Tags: atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	ComUserModuleCnfRef [ECUC_Com_10029]		
Parent Container	ComUserModule		
Description	Reference to the Com user module configuration. Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to destinationUri [ComUserUriDefSet/ComUser]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

C.7.7 ComUserUriDefSet

SWS Item	ECUC_Com_10039
EcucDestinationUriDefSet Name	ComUserUriDefSet
Description	Defines the set of DestinationUriDefs for the Com module.
Included EcucDestinationUriDefs	
Name	Description
ComUser	Defines the configuration container content of the Com user modules relevant settings.

SWS Item	[ECUC_Com_10040]
EcucDestinationUriDef Name	ComUser
Destination Uri Definition Set	ComUserUriDefSet
Description	Defines the configuration container content of the Com user modules relevant settings.
destinationUriNesting Contract	vertexOfTargetContainer
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComUserModuleCnf	0..1	Contains the configuration parameters of the Com user module. Tags: atp.Status=draft

C.7.8 ComSignalGroup

SWS Item	[ECUC_Com_00345]		
Container Name	ComSignalGroup		
Parent Container	ComConfig		
Description	Contains the configuration parameters of the AUTOSAR COM module's signal groups.		
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			

Name	ComDataInvalidAction [ECUC_Com_00314]		
Parent Container	ComSignalGroup		
Description	This parameter defines the action performed upon reception of an invalid signal. Relating to signal groups the action in case if one of the included signals is an invalid signal. If Replace is used the ComSignalNitValue will be used for the replacement.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	NOTIFY		
Post-Build Variant Multiplicity	REPLACE false		Literal for DataInvalidAction
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComErrorNotification [ECUC_Com_00499] (Obsolete)		
Parent Container	ComSignalGroup		
Description	Only valid on sender side: Name of Com_CbkTxErr callback function to be called. If this parameter is omitted no error notification shall take place. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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
---------------------------	--------------

Name	ComFirstTimeout [ECUC_Com_00183]		
Parent Container	ComSignalGroup		
Description	Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 3600]		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComHandleId [ECUC_Com_00165]		
Parent Container	ComSignalGroup		
Description	The numerical value used as the ID. This ID identifies signals and signal groups in the COM APIs using Com_SignalIdType or Com_SignalGroupIdType parameter respectively.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

Name	ComInitialValueOnly [ECUC_Com_00811]		
Parent Container	ComSignalGroup		
Description	This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the users (e.g. RTE, SwCluC). Thus the Com implementation does not need to expect any API calls for this signal (group).		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComInvalidNotification [ECUC_Com_00315] (Obsolete)		
Parent Container	ComSignalGroup		
Description	Only valid on receiver side: Name of Com_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAction is configured to NOTIFY. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComNotification [ECUC_Com_00498] (Obsolete)		
Parent Container	ComSignalGroup		
Description	<p>On sender side: Name of Com_CbkTxAck callback function to be called. On receiver side: Name of Com_CbkRxAck callback function to be called.</p> <p>If this parameter is omitted no notification shall take place.</p> <p>Tags: atp.Status=obsolete</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComRxDataTimeoutAction [ECUC_Com_00412]	
Parent Container	ComSignalGroup	
Description	This parameter defines the action performed upon expiration of the reception deadline monitoring timer.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	NONE	no replacement shall take place
	REPLACE	signals shall be replaced by their ComSignalInitValue
	SUBSTITUTE	signals shall be replaced by their ComTimeoutSubstitutionValue
Post-Build Variant Multiplicity	false	

Post-Build Variant Value	false		
Multiplicity 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		

Name	ComSignalGroupArrayAccess [ECUC_Com_10003]		
Parent Container	ComSignalGroup		
Description	Defines whether the uint8-array based access shall be used for this ComSignalGroup.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	ComTimeout [ECUC_Com_00263]		
Parent Container	ComSignalGroup		
Description	Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by ECUC_Com_00183.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 3600]		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		

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

Name	ComTimeoutNotification [ECUC_Com_00552] (Obsolete)		
Parent Container	ComSignalGroup		
Description	On sender side: Name of Com_CbkTxTOut callback function to be called. On receiver side: Name of Com_CbkRxTOut callback function to be called. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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		

Name	ComTransferProperty [ECUC_Com_00232]		
Parent Container	ComSignalGroup		
Description	Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	PENDING	A write access to this signal never triggers the transmission of the corresponding I-PDU.	

	TRIGGERED	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.	
	TRIGGERED_ON_CHANNEL	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.	
	TRIGGERED_ON_CHANNEL_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.	
	TRIGGERED_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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		

Name	ComUpdateBitPosition [ECUC_Com_00257]	
Parent Container	ComSignalGroup	
Description	Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 4294967295	
Default Value		
Post-Build Variant Multiplicity	true	
Post-Build Variant Value	true	

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

Name	ComSystemTemplateSignalGroupRef [ECUC_Com_00001]		
Parent Container	ComSignalGroup		
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this ComSignalGroup represents.		
Multiplicity	0..1		
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComGroupSignal	0..*	This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.

C.8 LdCom

Module SWS Item	ECUC_LdCom_00001		
Module Name	LdCom		
Module Description	Configuration of the AUTOSAR LdCom module.		
Post-Build Variant Support	true		
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE		
Included Containers			
Container Name	Multiplicity	Scope / Dependency	

Container Name	Multiplicity	Scope / Dependency
LdComConfig	1	This container contains the configuration parameters and sub containers of the AUTOSAR LdCom module.
LdComGeneral	1	Contains the general configuration parameters of the LdCom module.

C.8.1 LdComConfig

SWS Item	[ECUC_LdCom_00003]
Container Name	LdComConfig
Parent Container	LdCom
Description	This container contains the configuration parameters and sub containers of the AUTOSAR LdCom module.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
LdComIPdu	0..*	Contains the configuration parameters of the LdCom's signal (IPdu) inside LdCom.
LdComUserModule	0..*	Contains the configuration parameters of the LdCom user modules. Tags: atp.Status=draft

C.8.2 LdComIPdu

SWS Item	[ECUC_LdCom_00006]		
Container Name	LdComIPdu		
Parent Container	LdComConfig		
Description	Contains the configuration parameters of the LdCom's signal (IPdu) inside LdCom.		
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			

Name	LdComApiType [ECUC_LdCom_00002]		
Parent Container	LdComIPdu		
Description	<p>Defines if this I-PDU is a normal I-PDU that shall be sent unfragmented or if this is a large I-PDU that shall be sent via the Transport Protocol of the underlying bus.</p> <p>This setting is used by RTE to invoke the proper API.</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	LDCOM_IF	sent or received via interface API.	
	LDCOM_TP	sent or received via transport protocol API.	
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: ECU		

Name	LdComHandleId [ECUC_LdCom_00005]		
Parent Container	LdComIPdu		
Description	<p>This is the ID used by the LdCom users (e.g. RTE) to invoke LdCom. A corresponding shortName is created, which is used for the invocations of the users (e.g. RTE). The same ID is used for invocations by PduR.</p>		
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		

Name	LdComIPduDirection [ECUC_LdCom_00007]		
Parent Container	LdComIPdu		
Description	<p>The direction defines if this IPdu, and therefore the contributing signal, shall be sent or received.</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	LDCOM_RECEIVE		
	LDCOM_SEND		
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		

Name	LdComRxCopyRxData [ECUC_LdCom_00013] (Obsolete)		
Parent Container	LdComIPdu		
Description	Only on receiver side: Name of Rte_LdComCbkJCopyRxData callback function to be called. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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: ECU		

Name	LdComRxIndication [ECUC_LdCom_00014] (Obsolete)		
Parent Container	LdComIPdu		
Description	Only on receiver side: Name of Rte_LdComCbkJRxIndication callback function to be called. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity 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: ECU		

Name	LdComRxStartOfReception [ECUC_LdCom_00015] (Obsolete)		
Parent Container	LdComIPdu		
Description	Only on receiver side: Name of Rte_LdComCbKStartOfReception callback function to be called. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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: ECU		

Name	LdComTpRxIndication [ECUC_LdCom_00016] (Obsolete)		
Parent Container	LdComIPdu		
Description	Only on receiver side: Name of Rte_LdComCbKTpRxIndication callback function to be called. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			

Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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: ECU		

Name	LdComTpTxConfirmation [ECUC_LdCom_00017] (Obsolete)		
Parent Container	LdComIPdu		
Description	Only on sender side: Name of Rte_LdComCbKpTxConfirmation callback function to be called. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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: ECU		

Name	LdComTxConfirmation [ECUC_LdCom_00021] (Obsolete)		
Parent Container	LdComIPdu		
Description	<p>Only on sender side: Name of Rte_LdComCbktxConfirmation callback function to be called.</p> <p>Tags: atp.Status=obsolete</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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: ECU		

Name	LdComTxCopyTxData [ECUC_LdCom_00018] (Obsolete)		
Parent Container	LdComIPdu		
Description	<p>Only on sender side: Name of Rte_LdComCbkcpxTxData callback function to be called.</p> <p>Tags: atp.Status=obsolete</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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: ECU		

Name	LdComTxTriggerTransmit [ECUC_LdCom_00019] (Obsolete)		
Parent Container	LdComIPdu		
Description	Only on sender side: Name of Rte_LdComCbKTriggerTransmit callback function to be called. If defined TriggerTransmit has to be supported for this signal. Tags: atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity 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: ECU		

Name	LdComPduRef [ECUC_LdCom_00010]		
Parent Container	LdComIPdu		
Description	Reference to the global Pdu.		
Multiplicity	1		
Type	Reference to Pdu		
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: ECU		

Name	LdComSystemTemplateSignalRef [ECUC_LdCom_00011]		
Parent Container	LdComIPdu		
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template).		
Multiplicity	0..1		
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
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: ECU		

No Included Containers

C.8.3 LdComUserModule

SWS Item	[ECUC_LdCom_00029]		
Container Name	LdComUserModule		
Parent Container	LdComConfig		
Description	Contains the configuration parameters of the LdCom user modules. Tags: atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	LdComUserModuleCnfRef [ECUC_LdCom_00032]		
Parent Container	LdComUserModule		
Description	Reference to the LdCom user module configuration. Tags: atp.Status=draft		
Multiplicity	1		
Type	Reference to destinationUri [LdComUserUriDefSet/LdComUser]		
Post-Build Variant Multiplicity	false		

Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

C.8.4 LdComUserUriDefSet

SWS Item	ECUC_LdCom_00034
EcucDestinationUriDefSet Name	LdComUserUriDefSet
Description	Defines the set of DestinationUriDefs for the LdCom module.
Included EcucDestinationUriDefs	
Name	Description
LdComUser	Defines the configuration container content of the LdCom user modules relevant settings.

SWS Item	[ECUC_LdCom_00035]
EcucDestinationUriDef Name	LdComUser
Destination Uri Definition Set	LdComUserUriDefSet
Description	Defines the configuration container content of the LdCom user modules relevant settings.
destinationUriNesting Contract	vertexOfTargetContainer
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
LdComUserModuleCnf	0..1	Contains the configuration parameters of the LdCom user module. Tags: atp.Status=draft

D Referenced C-API

D.1 RTE

D.1.1 Rte_Rips_DatalsUpdated

Service Name	Rte_Rips_<PlugIn>_DatalsUpdated_<SwcBswI>_<CGI> (draft)	
Syntax	<pre>boolean Rte_Rips_<PlugIn>_DataIsUpdated_<SwcBswI>_<CGI> (void)</pre>	
Service ID [hex]	0xB4	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	boolean	The return value is used to indicate if the data has been updated or not.
Description	The Rte_Rips_DatalsUpdated API provides access to the update flag for an explicit receiver Tags: atp.Status=draft	
Available via	Rte_Rips_<PlugIn>_<SwcBswI>.h	

D.1.2 Rte_Rips_DRead

Service Name	Rte_Rips_<PlugIn>_DRead_<SwcBswI>[Partition][_<ExE>]_<CGI>(draft)	
Syntax	<pre><return> Rte_Rips_<PlugIn>_DRead_<SwcBswI> [Partition] [_<ExE>]_<CGI> ([Std_TransformerError transformerError])</pre>	
Service ID [hex]	0xFF	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	transformerError	The OUT parameter transformerError contains the transformer error which occurred during execution of the transformer chain.
Return value	<return>	Return value provides access to the data value of the Variable DataPrototype. For details of the <return> value definition see section 5.2.6.6.
Description	Rte_Rips_DRead Performs an "explicit" read on a sender-receiver communication data element typed by a primitive data type.	
Available via	Rte_Rips_<PlugIn>_<SwcBswI>.h	

D.1.3 Rte_Rips_DatalsUpdated_EventActivation

Service Name	Rte_Rips_<PlugIn>_DatalsUpdatedEventActivation_<SwcBswI>_<DR>_<CGI> (draft)	
Syntax	<pre>boolean Rte_Rips_<PlugIn>_DataIsUpdatedEventActivation_<SwcBswI>_<DR>_ <CGI> (void)</pre>	
Service ID [hex]	0xB5	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	boolean	The return value is used to indicate if the Runnable shall be activated or not.
Description	<p>The Rte_Rips_DatalsUpdated_EventActivation API indicates the RTE to activate the Runnable Entity triggered by DataReceivedEvent when the related VariableDataPrototype has been updated</p> <p>Tags: atp.Status=draft</p>	
Available via	Rte_Rips_<PlugIn>.h	

D.1.4 Rte_Rips_Feedback

Service Name	Rte_Rips_<PlugIn>_Feedback_ [<SwcBswI>][Partition]_<CGI> (draft)	
Syntax	<pre>Std_ReturnType Rte_Rips_<PlugIn>_Feedback_ [<SwcBswI>] [Partition]_<CGI> (void)</pre>	
Service ID [hex]	0xB6	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	The return value is used to pass error notifications.
Description	<p>The Rte_Rips_Feedback API provides access to acknowledgment notifications for explicit and implicit sender-receiver communication and to pass error notification to senders in a Software Cluster</p> <p>Tags: atp.Status=draft</p>	
Available via	Rte_Rips_<PlugIn>_<SwcBswI>.h	

D.1.5 Rte_Rips_Invoke

Service Name	Rte_Rips_<PlugIn>_Invoke_<SwcBswI>_<CGI>(draft)	
Syntax	<pre>Std_ReturnType Rte_Rips_<PlugIn>_Invoke_<SwcBswI>_<CGI> ([IN IN/OUT OUT] <data_1>, [IN IN/OUT OUT] ..., [IN IN/OUT OUT] <data_n>, [Std_TransformerError transformerError])</pre>	
Service ID [hex]	0xEC	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	<data_1>	The Rte_Rips_Invoke API includes zero or more IN, IN/OUT and OUT parameters according SWS_Rte_01102 and none in case of triggers
Parameters (inout)	...	The Rte_Rips_Invoke API includes zero or more IN, IN/OUT and OUT parameters according SWS_Rte_01102 and none in case of triggers
Parameters (out)	<data_n>	The Rte_Rips_Invoke API includes zero or more IN, IN/OUT and OUT parameters according SWS_Rte_01102 and none in case of triggers
	transformerError	The OUT parameter transformerError contains the transformer error which occurred during execution of the transformer chain.
Return value	Std_ReturnType	The return value is used to indicate communication errors.
Description	Rte_Rips_Invoke performs a transformer or cross cluster invocation for clients or trigger sources.	
Available via	Rte_Rips_<PlugIn>_<SwcBswI>.h	

D.1.6 Rte_Rips_Prm

Service Name	Rte_Rips_<PlugIn>_Prm_<CGI> (draft)	
Syntax	<pre><return> Rte_Rips_<PlugIn>_Prm_<CGI> (void)</pre>	
Service ID [hex]	0x100	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	<return>	For primitive data types, the Rte_Rips_Prm API returns the parameter value. For composite data types, the Rte_Rips_Prm API returns a reference (in C, a pointer) to the constant parameter.
Description	The Rte_Rips_Prm API provides access to a parameter provided by another Software Cluster Tags: atp.Status=draft	
Available via	Rte_Rips_<PlugIn>_<SwcBswI>.h	

D.1.7 Rte_Rips_Read

Service Name	Rte_Rips_<PlugIn>_Read_<SwcBswI>[[Partition]][_<ExE>]_<CGI>(draft)	
Syntax	<pre>Std_ReturnType Rte_Rips_<PlugIn>_Read_<SwcBswI> [[Partition]][_<ExE>]_<CGI> (OUT <data>, [Std_TransformerError transformerError])</pre>	
Service ID [hex]	0xEA	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	<data>	The OUT parameter <data> pass back the received data.
	transformerError	The OUT parameter transformerError contains the transformer error which occurred during execution of the transformer chain.
Return value	Std_ReturnType	The return value is used to indicate communication errors.
Description	Rte_Rips_Read Performs an "explicit" read on a sender-receiver communication data element.	
Available via	Rte_Rips_<PlugIn>_<SwcBswI>.h	

D.1.8 Rte_Rips_ReturnResult

Service Name	Rte_Rips_<PlugIn>_ReturnResult_<SwcBswI>_<CGI>(draft)	
Syntax	<pre>Std_ReturnType Rte_Rips_<PlugIn>_ReturnResult_<SwcBswI>_<CGI> ([IN/OUT OUT] <param_1>, [IN/OUT OUT] <param_n>, [Std_TransformerError transformerError])</pre>	
Service ID [hex]	0xED	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	<param_1>	The Rte_Rips_ReturnResult API includes zero or more IN/OUT and OUT parameters according SWS_Rte_01111.
Parameters (out)	<param_n>	The Rte_Rips_ReturnResult API includes zero or more IN/OUT and OUT parameters according SWS_Rte_01111.
	transformerError	The OUT parameter transformerError contains the transformer error which occurred during execution of the transformer chain.
Return value	Std_ReturnType	The return value is used to indicate communication errors
Description	Rte_Rips_ReturnResult get the server results of a performed a transformer or cross cluster invocation for clients.	
Available via	Rte_Rips_<PlugIn>_<SwcBswI>.h	

D.1.9 Rte_Rips_Start

Service Name	Rte_Rips_<PlugIn>_Rte_Start
Syntax	<pre>void Rte_Rips_<PlugIn>_Rte_Start (void)</pre>
Service ID [hex]	0xF1
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Rte_Rips_Rte_Start initializes those RTE Implementation Plug-In parts which are relevant for the RTE related operation.
Available via	Rte_Rips_<PlugIn>.h

D.1.10 Rte_Rips_Stop

Service Name	Rte_Rips_<PlugIn>_Rte_Stop
Syntax	<pre>void Rte_Rips_<PlugIn>_Rte_Stop (void)</pre>
Service ID [hex]	0xF2
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Rte_Rips_Rte_Stop deinitializes those RTE Implementation Plug-In parts which are relevant for the RTE related operation.
Available via	Rte_Rips_<PlugIn>.h

D.1.11 Rte_Rips_SchM_Deinit

Service Name	Rte_Rips_SchM_Deinit
Syntax	<pre>void Rte_Rips_SchM_Deinit (void)</pre>
Service ID [hex]	0xF3
Sync/Async	Synchronous





Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Rte_Rips_SchM_Deinit deinitializes those RTE Implementation Plug-In parts which are relevant for the SchM related operations.
Available via	Rte_Rips_<PlugIn>.h

D.1.12 Rte_Rips_SchM_Init

Service Name	Rte_Rips_<PlugIn>_SchM_Init
Syntax	<pre>void Rte_Rips_<PlugIn>_SchM_Init (void)</pre>
Service ID [hex]	0xF0
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Rte_Rips_SchM_Init initializes those RTE Implementation Plug-In parts which are relevant for the SchM related operations.
Available via	Rte_Rips_<PlugIn>.h

D.1.13 Rte_Rips_SwitchNotificationStatusType

Name	Rte_Rips_SwitchNotificationStatusType (draft)		
Kind	Type		
Derived from	uint8		
Range	RTE_SWITCH_NOTIFICATION_SKIP	0x00	mode switch notification cannot be dequeued
	RTE_SWITCH_NOTIFICATION_ENQUEUED_FIRST	0x01	mode switch notification is enqueued into an empty mode queue
	RTE_SWITCH_NOTIFICATION_ENQUEUED_NOT_FIRST	0x02	mode switch notification is enqueued into a non empty mode queue
	RTE_SWITCH_NOTIFICATION_ENQUEUE_FAILED	0x03	enqueue operation into a non empty mode queue failed
	RTE_SWITCH_NOTIFICATION_DEQUEUED_LAST	0x04	last mode switch notification was enqueued from mode queue





	RTE_SWITCH_NOTIFICATION_DEQUEUED_NOT_LAST	0x05	mode switch notification was enqueued from mode queue, further mode switch notifications are in the queue
Description	Status of the en- and dequeue operation on a mode queue Tags: atp.Status=draft		
Available via	Rte_Type.h		

D.1.14 Rte_Rips_Switch

Service Name	Rte_Rips_<PlugIn>_Switch_<BswSwcI>_<MMI> (draft)	
Syntax	<pre>void Rte_Rips_<PlugIn>_Switch_<BswSwcI>_<MMI> (Rte_Rips_SwitchNotificationStatusType switchNotificationStatus, uint32 previousmode, uint32 nextmode)</pre>	
Service ID [hex]	0xB0	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	switchNotificationStatus	Status of the enqueue operation
	previousmode	The value of the ModeDeclaration of the mode being left
	nextmode	The value of the ModeDeclaration of the mode being entered
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Rte_Rips_StartModeSwitch notifies the RTE Implementation Plug-In about an enqueue operation in a mode queue. Tags: atp.Status=draft	
Available via	Rte_Rips_<PlugIn>.h	

D.1.15 Rte_Rips_DequeueModeSwitch

Service Name	Rte_Rips_<PlugIn>_DequeueModeSwitch_<MMI>_<OsTask> (draft)	
Syntax	<pre>Rte_Rips_SwitchNotificationStatusType Rte_Rips_<PlugIn>_DequeueModeSwitch_<MMI>_<OsTask> (void)</pre>	
Service ID [hex]	0xB1	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	





Return value	Rte_Rips_Switch NotificationStatusType	The return value is used indicate the status of the dequeue operation in a mode queue
Description	Rte_Rips_DequeueModeSwitch dequeues a mode switch notification from the mode queue when it is called after the last on-entry ExecutableEntity terminated. Tags: atp.Status=draft	
Available via	Rte_Buffers.h	

D.1.16 Rte_Rips_Trigger

Service Name	Rte_Rips_<PlugIn>_Trigger_<BswSwcI>_<MMI> (draft)	
Syntax	<pre>void Rte_Rips_<PlugIn>_Trigger_<BswSwcI>_<MMI> (void)</pre>	
Service ID [hex]	0xB2	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Rte_Rips_Trigger notifies the RTE Implementation Plug-In about a raised trigger. Tags: atp.Status=draft	
Available via	Rte_Rips_<PlugIn>.h	

D.1.17 Rte_Rips_Write

Service Name	Rte_Rips_<PlugIn>_Write_<SwcBswI>[Partition]_<ExE>_<CGI>(draft)	
Syntax	<pre>Std_ReturnType Rte_Rips_<PlugIn>_Write_<SwcBswI> [Partition] [<ExE>]_ <CGI> (IN <data>, [Std_TransformerError transformerError])</pre>	
Service ID [hex]	0xEB	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	<data>	The IN parameter <data> pass the received data.
Parameters (inout)	None	
Parameters (out)	transformerError	The OUT parameter transformerError contains the transformer error which occurred during execution of the transformer chain.
Return value	Std_ReturnType	The return value is used to indicate communication errors.
Description	Rte_Rips_Write Performs an "explicit" write on a sender-receiver communication data element.	
Available via	Rte_Rips_<PlugIn>_<SwcBswI>.h	

D.2 OS

See document [12] and [17] as reference for Os.

- TASK
- ActivateTask
- ChainTask
- TerminateTask
- GetResource
- ReleaseResource
- SuspendOSInterrupts
- ResumeOSInterrupts
- GetSpinlock
- ReleaseSpinlock
- GetApplicationID
- StartScheduleTable
- StopScheduleTable
- Schedule
- DisableAllInterrupts
- EnableAllInterrupts
- SuspendAllInterrupts
- ResumeAllInterrupts
- GetResource
- ReleaseResource
- SuspendOSInterrupts
- ResumeOSInterrupts
- ReleaseResource
- GetSpinlock
- ReleaseSpinlock
- GetApplicationID

D.3 NvM

D.3.1 NvM_CancelJobs

Service Name	NvM_CancelJobs	
Syntax	<pre>Std_ReturnType NvM_CancelJobs (NvM_BlockIdType BlockId)</pre>	
Service ID [hex]	0x10	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The job was successfully removed from queue. E_NOT_OK: The job could not be found in the queue.
Description	Service to cancel all jobs pending for a NV block.	
Available via	NvM.h	

D.3.2 NvM_EraseNvBlock

Service Name	NvM_EraseNvBlock	
Syntax	<pre>Std_ReturnType NvM_EraseNvBlock (NvM_BlockIdType BlockId)</pre>	
Service ID [hex]	0x09	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: request has been accepted E_NOT_OK: request has not been accepted
Description	Service to erase a NV block.	
Available via	NvM.h	

D.3.3 NvM_GetDataIndex

Service Name	NvM_GetDataIndex	
Syntax	<pre>Std_ReturnType NvM_GetDataIndex (NvM_BlockIdType BlockId, uint8* DataIndexPtr)</pre>	
Service ID [hex]	0x02	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	DataIndexPtr	Pointer to where to store the current dataset index (0..255)
Return value	Std_ReturnType	E_OK: The index position has been retrieved successfully. E_NOT_OK: An error occurred.
Description	Service for getting the currently set DataIndex of a dataset NVRAM block	
Available via	NvM.h	

D.3.4 NvM_GetErrorStatus

Service Name	NvM_GetErrorStatus	
Syntax	<pre>Std_ReturnType NvM_GetErrorStatus (NvM_BlockIdType BlockId, NvM_RequestResultType* RequestResultPtr)</pre>	
Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	RequestResultPtr	Pointer to where to store the request result. See NvM_RequestResultType .
Return value	Std_ReturnType	E_OK: The block dependent error/status information was read successfully. E_NOT_OK: An error occurred.
Description	Service to read the block dependent error/status information.	
Available via	NvM.h	

D.3.5 NvM_InvalidateNvBlock

Service Name	NvM_InvalidateNvBlock	
Syntax	Std_ReturnType NvM_InvalidateNvBlock (NvM_BlockIdType BlockId)	
Service ID [hex]	0x0b	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: request has been accepted E_NOT_OK: request has not been accepted
Description	Service to invalidate a NV block.	
Available via	NvM.h	

D.3.6 NvM_ReadBlock

Service Name	NvM_ReadBlock	
Syntax	Std_ReturnType NvM_ReadBlock (NvM_BlockIdType BlockId, void* NvM_DstPtr)	
Service ID [hex]	0x06	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	NvM_DstPtr	Pointer to the RAM data block.
Return value	Std_ReturnType	E_OK: request has been accepted E_NOT_OK: request has not been accepted
Description	Service to copy the data of the NV block to its corresponding RAM block.	
Available via	NvM.h	

D.3.7 NvM_ReadPRAMBlock

Service Name	NvM_ReadPRAMBlock
---------------------	-------------------





Syntax	Std_ReturnType NvM_ReadPRAMBlock (NvM_BlockIdType BlockId)	
Service ID [hex]	0x16	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: request has been accepted E_NOT_OK: request has not been accepted
Description	Service to copy the data of the NV block to its corresponding permanent RAM block.	
Available via	NvM.h	

D.3.8 NvM_RestoreBlockDefaults

Service Name	NvM_RestoreBlockDefaults	
Syntax	Std_ReturnType NvM_RestoreBlockDefaults (NvM_BlockIdType BlockId, void* NvM_DestPtr)	
Service ID [hex]	0x08	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	NvM_DestPtr	Pointer to the RAM data block.
Return value	Std_ReturnType	E_OK: request has been accepted E_NOT_OK: request has not been accepted
Description	Service to restore the default data to its corresponding RAM block.	
Available via	NvM.h	

D.3.9 NvM_RestorePRAMBlockDefaults

Service Name	NvM_RestorePRAMBlockDefaults	
Syntax	Std_ReturnType NvM_RestorePRAMBlockDefaults (NvM_BlockIdType BlockId)	
Service ID [hex]	0x18	
Sync/Async	Asynchronous	





Reentrancy	Non Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: request has been accepted E_NOT_OK: request has not been accepted
Description	Service to restore the default data to its corresponding permanent RAM block.	
Available via	NvM.h	

D.3.10 NvM_SetBlockLockStatus

Service Name	NvM_SetBlockLockStatus	
Syntax	<pre>void NvM_SetBlockLockStatus (NvM_BlockIdType BlockId, boolean BlockLocked)</pre>	
Service ID [hex]	0x13	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
	BlockLocked	TRUE: Mark the RAM.block as locked FALSE: Mark the RAM.block as unlocked
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Service for setting the lock status of a permanent RAM block or of the explicit synchronization of a NVRAM block.	
Available via	NvM.h	

D.3.11 NvM_SetBlockProtection

Service Name	NvM_SetBlockProtection	
Syntax	<pre>Std_ReturnType NvM_SetBlockProtection (NvM_BlockIdType BlockId, boolean ProtectionEnabled)</pre>	
Service ID [hex]	0x03	
Sync/Async	Synchronous	
Reentrancy	Reentrant	





Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
	ProtectionEnabled	TRUE: Write protection shall be enabled FALSE: Write protection shall be disabled
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The block was enabled/disabled as requested E_NOT_OK: An error occurred.
Description	Service for setting/resetting the write protection for a NV block.	
Available via	NvM.h	

D.3.12 NvM_SetDataIndex

Service Name	NvM_SetDataIndex	
Syntax	<pre>Std_ReturnType NvM_SetDataIndex (NvM_BlockIdType BlockId, uint8 DataIndex)</pre>	
Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
	DataIndex	Index position (association) of a NV/ROM block.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The index position was set successfully. E_NOT_OK: An error occurred.
Description	Service for setting the DataIndex of a dataset NVRAM block.	
Available via	NvM.h	

D.3.13 NvM_SetRamBlockStatus

Service Name	NvM_SetRamBlockStatus	
Syntax	<pre>Std_ReturnType NvM_SetRamBlockStatus (NvM_BlockIdType BlockId, boolean BlockChanged)</pre>	
Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant	





Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
	BlockChanged	TRUE: Validate the permanent RAM block or the explicit synchronization and mark block as changed. FALSE: Invalidate the permanent RAM block or the explicit synchronization and mark block as unchanged.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The status of the permanent RAM block or the explicit synchronization was changed as requested. E_NOT_OK: An error occurred.
Description	Service for setting the RAM block status of a permanent RAM block or the status of the explicit synchronization of a NVRAM block.	
Available via	NvM.h	

D.3.14 NvM_WriteBlock

Service Name	NvM_WriteBlock	
Syntax	<pre>Std_ReturnType NvM_WriteBlock (NvM_BlockIdType BlockId, const void* NvM_SrcPtr)</pre>	
Service ID [hex]	0x07	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
	NvM_SrcPtr	Pointer to the RAM data block.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: request has been accepted E_NOT_OK: request has not been accepted
Description	Service to copy the data of the RAM block to its corresponding NV block.	
Available via	NvM.h	

D.3.15 NvM_WritePRAMBlock

Service Name	NvM_WritePRAMBlock	
Syntax	<pre>Std_ReturnType NvM_WritePRAMBlock (NvM_BlockIdType BlockId)</pre>	
Service ID [hex]	0x17	
Sync/Async	Asynchronous	





Reentrancy	Reentrant	
Parameters (in)	BlockId	The block identifier uniquely identifies one NVRAM block descriptor. A NVRAM block descriptor contains all needed information about a single NVRAM block.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: request has been accepted E_NOT_OK: request has not been accepted
Description	Service to copy the data of the permanent RAM block to its corresponding NV block.	
Available via	NvM.h	

D.3.16 NvM_SingleBlockCallbackFunction

Service Name	NvM_SingleBlockCallbackFunction	
Syntax	<pre>Std_ReturnType NvM_SingleBlockCallbackFunction (NvM_BlockRequestType BlockRequest, NvM_RequestResultType JobResult)</pre>	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	BlockRequest	The request type (read, write, ... etc.) of the previous processed block job
	JobResult	The request result of the previous processed block job.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: callback function has been processed successfully any other: callback function has been processed unsuccessfully
Description	Per block callback routine to notify the upper layer that an asynchronous single block request has been finished.	
Available via	NvM_Externals.h	

D.3.17 NvM_InitBlockCallbackFunction

Service Name	NvM_InitBlockCallbackFunction	
Syntax	<pre>Std_ReturnType NvM_InitBlockCallbackFunction (NvM_InitBlockRequestType InitBlockRequest)</pre>	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	InitBlockRequest	The request type (read, restore, ... etc.) of the currently processed block
Parameters (inout)	None	
Parameters (out)	None	





Return value	Std_ReturnType	E_OK: callback function has been processed successfully any other: callback function has been processed unsuccessfully
Description	Per block callback routine which shall be called by the NvM module when default data needs to be restored in RAM, and a ROM block is not configured.	
Available via	NvM_Externals.h	

D.3.18 NvM_ReadRamBlockFromNvm

Service Name	NvM_ReadRamBlockFromNvm	
Syntax	Std_ReturnType NvM_ReadRamBlockFromNvm (const void* NvMBuffer)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	NvMBuffer	the address of the buffer where the data can be read from
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: callback function has been processed successfully any other: callback function has been processed unsuccessfully
Description	Block specific callback routine which shall be called in order to let the application copy data from NvM module's mirror to RAM block.	
Available via	NvM_Externals.h	

D.3.19 NvM_WriteRamBlockToNvm

Service Name	NvM_WriteRamBlockToNvm	
Syntax	Std_ReturnType NvM_WriteRamBlockToNvm (void* NvMBuffer)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	NvMBuffer	the address of the buffer where the data shall be written to
Return value	Std_ReturnType	E_OK: callback function has been processed successfully any other: callback function has been processed unsuccessfully
Description	Block specific callback routine which shall be called in order to let the application copy data from RAM block to NvM module's mirror.	
Available via	NvM_Externals.h	

D.4 Com

D.4.1 Com_SendSignal

Service Name	Com_SendSignal	
Syntax	<pre>uint8 Com_SendSignal (Com_SignalIdType SignalId, const void* SignalDataPtr)</pre>	
Service ID [hex]	0x0a	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be sent.
	SignalDataPtr	Reference to the signal data to be transmitted.
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling
Description	The service Com_SendSignal updates the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter.	
Available via	Com.h	

D.4.2 Com_SendDynSignal

Service Name	Com_SendDynSignal	
Syntax	<pre>uint8 Com_SendDynSignal (Com_SignalIdType SignalId, const void* SignalDataPtr, uint16 Length)</pre>	
Service ID [hex]	0x21	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be sent.
	SignalDataPtr	Reference to the signal data to be transmitted.
	Length	Length of the dynamic length signal
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted E_NOT_OK: in case the Length is greater than the configured ComSignalLength of this sent signal COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked





Description	The service Com_SendDynSignal updates the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter.
Available via	Com.h

D.4.3 Com_SendSignalGroup

Service Name	Com_SendSignalGroup	
Syntax	<pre>uint8 Com_SendSignalGroup (Com_SignalGroupIdType SignalGroupId)</pre>	
Service ID [hex]	0x0d	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be sent.
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling
Description	The service Com_SendSignalGroup copies the content of the associated shadow buffer to the associated I-PDU.	
Available via	Com.h	

D.4.4 Com_SendSignalGroupArray

Service Name	Com_SendSignalGroupArray	
Syntax	<pre>uint8 Com_SendSignalGroupArray (Com_SignalGroupIdType SignalGroupId, const uint8* SignalGroupArrayPtr)</pre>	
Service ID [hex]	0x23	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be sent.
	SignalGroupArrayPtr	Reference to the signal group array.
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling





Description	The service Com_SendSignalGroupArray copies the content of the provided SignalGroupArray Ptr to the associated I-PDU. The provided data shall correspond to the array representation of the signal group.
Available via	Com.h

D.4.5 Com_SendSignalWithMetaData

Service Name	Com_SendSignalWithMetaData	
Syntax	<pre>uint8 Com_SendSignalWithMetaData (Com_SignalIdType SignalId, const void* SignalDataPtr, const uint8* MetaDataPtr)</pre>	
Service ID [hex]	0x4d	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be sent.
	SignalDataPtr	Reference to the signal data to be transmitted.
	MetaDataPtr	Pointer to the meta data of the signal
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling
Description	The service Com_SendSignalWithMetaData updates the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter. The meta data is provided via the MetaDataPtr parameter.	
Available via	Com.h	

D.4.6 Com_SendDynSignalWithMetaData

Service Name	Com_SendDynSignalWithMetaData	
Syntax	<pre>uint8 Com_SendDynSignalWithMetaData (Com_SignalIdType SignalId, const void* SignalDataPtr, uint16 Length, const uint8* MetaDataPtr)</pre>	
Service ID [hex]	0x4e	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be sent.
	SignalDataPtr	Reference to the signal data to be transmitted.





	Length	Length of the dynamic length signal
	MetaDataPtr	Pointer to the meta data of the signal
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted E_NOT_OK: in case the Length is greater than the configured ComSignalLength of this sent signal COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked
Description	The service Com_SendDynSignalWithMetaData updates the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter. The meta data is provided via the MetaDataPtr parameter.	
Available via	Com.h	

D.4.7 Com_SendSignalGroupWithMetaData

Service Name	Com_SendSignalGroupWithMetaData	
Syntax	<pre>uint8 Com_SendSignalGroupWithMetaData (Com_SignalGroupIdType SignalGroupId, const uint8* MetaDataPtr)</pre>	
Service ID [hex]	0x4f	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be sent.
	MetaDataPtr	Pointer to the meta data of the signal group
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling
Description	The service Com_SendSignalGroupWithMetaData copies the content of the associated shadow buffer to the associated I-PDU. The meta data is provided via the MetaDataPtr parameter.	
Available via	Com.h	

D.4.8 Com_SendSignalGroupArrayWithMetaData

Service Name	Com_SendSignalGroupArrayWithMetaData	
Syntax	<pre>uint8 Com_SendSignalGroupArrayWithMetaData (Com_SignalGroupIdType SignalGroupId, const uint8* SignalGroupArrayPtr, const uint8* MetaDataPtr)</pre>	





Service ID [hex]	0x50	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be sent.
	SignalGroupArrayPtr	Reference to the signal group array.
	MetaDataPtr	Pointer to the meta data of the signal group.
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling
Description	The service Com_SendSignalGroupArrayWithMetaData copies the content of the provided SignalGroupArrayPtr to the associated I-PDU. The provided data shall correspond to the array representation of the signal group. The meta data is provided via the MetaDataPtr parameter.	
Available via	Com.h	

D.4.9 Com_ReceiveSignal

Service Name	Com_ReceiveSignal	
Syntax	<pre>uint8 Com_ReceiveSignal (Com_SignalIdType SignalId, void* SignalDataPtr)</pre>	
Service ID [hex]	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be received.
Parameters (inout)	None	
Parameters (out)	SignalDataPtr	Reference to the location where the received signal data shall be stored
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling
Description	Com_ReceiveSignal copies the data of the signal identified by SignalId to the location specified by SignalDataPtr.	
Available via	Com.h	

D.4.10 Com_ReceiveDynSignal

Service Name	Com_ReceiveDynSignal
---------------------	----------------------





Syntax	<pre>uint8 Com_ReceiveDynSignal (Com_SignalIdType SignalId, void* SignalDataPtr, uint16* Length)</pre>	
Service ID [hex]	0x22	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be received.
Parameters (inout)	Length	in: maximum length that could be received out: length of the dynamic length signal
Parameters (out)	SignalDataPtr	reference to the location where the received signal data shall be stored
Return value	uint8	E_OK: service has been accepted E_NOT_OK: in case the Length (as in-parameter) is smaller than the received length of the dynamic length signal COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked
Description	Com_ReceiveDynSignal copies the data of the signal identified by SignalId to the location specified by SignalDataPtr and stores the length of the dynamical length signal at the position given by the Length parameter.	
Available via	Com.h	

D.4.11 Com_ReceiveSignalGroup

Service Name	Com_ReceiveSignalGroup	
Syntax	<pre>uint8 Com_ReceiveSignalGroup (Com_SignalGroupIdType SignalGroupId)</pre>	
Service ID [hex]	0x0e	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be received.
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling
Description	The service Com_ReceiveSignalGroup copies the received signal group from the I-PDU to the shadow buffer.	
Available via	Com.h	

D.4.12 Com_ReceiveSignalGroupArray

Service Name	Com_ReceiveSignalGroupArray	
Syntax	<pre>uint8 Com_ReceiveSignalGroupArray (Com_SignalGroupIdType SignalGroupId, uint8* SignalGroupArrayPtr)</pre>	
Service ID [hex]	0x24	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be received.
Parameters (inout)	None	
Parameters (out)	SignalGroupArrayPtr	reference to the location where the received signal group array shall be stored
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling
Description	The service Com_ReceiveSignalGroupArray copies the received signal group array representation from the PDU to the location designated by SignalGroupArrayPtr.	
Available via	Com.h	

D.4.13 Com_ReceiveSignalWithMetaData

Service Name	Com_ReceiveSignalWithMetaData	
Syntax	<pre>uint8 Com_ReceiveSignalWithMetaData (Com_SignalIdType SignalId, void* SignalDataPtr, uint8* MetaDataPtr)</pre>	
Service ID [hex]	0x49	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be received.
Parameters (inout)	None	
Parameters (out)	SignalDataPtr	Reference to the location where the received signal data shall be stored
	MetaDataPtr	Pointer to the meta data of the signal
Return value	uint8	–
Description	Com_ReceiveSignalWithMetaData copies the data of the signal identified by SignalId to the location specified by SignalDataPtr. The received meta data is stored at the position given by the MetaDataPtr parameter.	
Available via	Com.h	

D.4.14 Com_ReceiveDynSignalWithMetaData

Service Name	Com_ReceiveDynSignalWithMetaData	
Syntax	<pre>uint8 Com_ReceiveDynSignalWithMetaData (Com_SignalIdType SignalId, void* SignalDataPtr, uint16* Length, uint8* MetaDataPtr)</pre>	
Service ID [hex]	0x4a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be received.
Parameters (inout)	None	
Parameters (out)	SignalDataPtr	reference to the location where the received signal data shall be stored
	Length	–
	MetaDataPtr	Pointer to the meta data of the signal
Return value	uint8	E_OK: service has been accepted E_NOT_OK: in case the Length (as in-parameter) is smaller than the received length of the dynamic length signal COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked
Description	Com_ReceiveDynSignalWithMetaData copies the data of the signal identified by SignalId to the location specified by SignalDataPtr and stores the length of the dynamical length signal at the position given by the Length parameter. The received meta data is stored at the position given by the MetaDataPtr parameter.	
Available via	Com.h	

D.4.15 Com_ReceiveSignalGroupWithMetaData

Service Name	Com_ReceiveSignalGroupWithMetaData	
Syntax	<pre>uint8 Com_ReceiveSignalGroupWithMetaData (Com_SignalGroupIdType SignalGroupId, uint8* MetaDataPtr)</pre>	
Service ID [hex]	0x4b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be received.
Parameters (inout)	None	
Parameters (out)	MetaDataPtr	Pointer to the meta data of the signal group
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped COM_BUSY: in case the TP-Buffer is locked for large data types handling





Description	The service Com_ReceiveSignalGroupWithMetaData copies the received signal group from the I-PDU to the shadow buffer. The received meta data is stored at the position given by the MetaDataPtr parameter.
Available via	Com.h

D.4.16 Com_ReceiveSignalGroupArrayWithMetaData

Service Name	Com_ReceiveSignalGroupArrayWithMetaData	
Syntax	<pre>void Com_ReceiveSignalGroupArrayWithMetaData (Com_SignalGroupIdType SignalGroupId, uint8* SignalGroupArrayPtr, uint8* MetaDataPtr)</pre>	
Service ID [hex]	0x4c	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be received.
Parameters (inout)	None	
Parameters (out)	SignalGroupArrayPtr	reference to the location where the received signal group array shall be stored
	MetaDataPtr	Pointer to the meta data of the signal group.
Return value	None	
Description	The service Com_ReceiveSignalGroupArrayWithMetaData copies the received signal group array representation from the PDU to the location designated by SignalGroupArrayPtr. The received meta data is stored at the position given by the MetaDataPtr parameter.	
Available via	Com.h	

D.4.17 Com_InvalidateSignal

Service Name	Com_InvalidateSignal	
Syntax	<pre>uint8 Com_InvalidateSignal (Com_SignalIdType SignalId)</pre>	
Service ID [hex]	0x10	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalId	Id of signal to be invalidated.
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group is stopped, no ComSignalDataInvalidValue is configured for the given signalId COM_BUSY: in case the TP-Buffer is locked
Description	The service Com_InvalidateSignal invalidates the signal with the given SignalId by setting its value to its configured ComSignalDataInvalidValue.	





Available via	Com.h
----------------------	-------

D.4.18 Com_InvalidateSignalGroup

Service Name	Com_InvalidateSignalGroup	
Syntax	<pre>uint8 Com_InvalidateSignalGroup (Com_SignalGroupIdType SignalGroupId)</pre>	
Service ID [hex]	0x1b	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SignalGroupId	Id of signal group to be invalidated.
Parameters (inout)	None	
Parameters (out)	None	
Return value	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group is stopped, no ComSignalDataInvalidValue is configured for any of the group signals COM_BUSY: in case the TP-Buffer is locked
Description	The service Com_InvalidateSignalGroup invalidates all group signals of the signal group with the given SignalGroupId by setting their values to their configured ComSignalDataInvalidValues.	
Available via	Com.h	

D.5 LdCom

D.5.1 LdCom_Transmit

Service Name	LdCom_Transmit	
Syntax	<pre>Std_ReturnType LdCom_Transmit (PduIdType Id, const PduInfoType* InfoPtr)</pre>	
Service ID [hex]	0x49	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Ids. Non reentrant for the same Id.	
Parameters (in)	Id	Identifier of the signal to be transmitted.
	InfoPtr	Length of and pointer to the signal data and pointer to MetaData.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Transmit request has been accepted. E_NOT_OK: Transmit request has not been accepted.
Description	Requests transmission of a signal.	
Available via	LdCom.h	

D.6 Dcm

D.6.1 DataServices

D.6.1.1 ReadData

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ReadData (draft)	
Syntax	<pre>Std_ReturnType Xxx_ReadData (uint8* Data, uint16 DataLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x68	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DataLength	Length in byte of the data to be read
Parameters (inout)	None	
Parameters (out)	Data	Buffer where the requested data shall be copied to
	ErrorCode	If the operation <code>Xxx_ReadData</code> returns value <code>E_NOT_OK</code> , the Dcm module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application a data value of a DID/PID if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_SYNCH_FNC_PROXY</code> . Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ReadData (draft)	
Syntax	<pre>Std_ReturnType Xxx_ReadData (Dcm_OpStatusType OpStatus, uint8* Data, uint16 DataLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x69	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
	DataLength	Length in byte of the data to be read
Parameters (inout)	None	
Parameters (out)	Data	Buffer where the requested data shall be copied to





	ErrorCode	If the operation Xxx_ReadData returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application a data value of a DID/PID if DcmDspDataUsePort is set to USE_DATA_ASYNCH_FNC_PROXY.. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

D.6.1.2 WriteData

If `DcmDspDataType` is set to `USE_DATA_SYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_WriteData (draft)	
Syntax	<pre>Std_ReturnType Xxx_WriteData (const uint8* Data, uint16 DataLength, Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x70	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Data	Buffer containing the data to be written
	DataLength	Length in byte of the data to be written
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_WriteData returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests the application to write a data value of a DID if DcmDspDataUsePort is set to USE_DATA_SYNCH_FNC_PROXY.. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

If `DcmDspDataType` is set to `USE_DATA_ASYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_WriteData (draft)
---------------------	-----------------------





Syntax	<pre>Std_ReturnType Xxx_WriteData (const uint8* Data, uint16 DataLength, Dcm_OpStatusType OpStatus, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x71	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Data	Buffer containing the data to be written
	DataLength	Length in byte of the data to be written
	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation <code>Xxx_WriteData</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	<p><code>E_OK</code>: Request was successful.</p> <p><code>E_NOT_OK</code>: Request was not successful.</p> <p><code>DCM_E_PENDING</code>: Request is not yet finished. Further call(s) required to finish.</p>
Description	<p>This function requests the application to write a data value of a DID if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNC_FNC_PROXY</code>.</p> <p>Tags: atp.Status=draft</p>	
Available via	Dcm_Externals.h	

D.6.1.3 ConditionCheckRead

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_FNC`, or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR`, `USE_DATA_ASYNC_FNC_ERROR` or `USE_DATA_ASYNC_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ConditionCheckRead	
Syntax	<pre>Std_ReturnType Xxx_ConditionCheckRead (Dcm_OpStatusType OpStatus, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x37	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation <code>Xxx_ConditionCheckRead</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.





Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application if the conditions to read the Data are correct.	
Available via	Dcm_Externals.h	

D.6.1.4 ReadDataLength

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_CLIENT_SERVER`, or `USE_DATA_SYNCH_FNC`, or `USE_DATA_SYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ReadDataLength (draft)	
Syntax	Std_ReturnType Xxx_ReadDataLength (uint16* DataLength)	
Service ID [hex]	0x36	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	DataLength	Length in byte of the data to be read
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests the application to return the data length in byte of a Data. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_FNC_ERROR`, or `USE_DATA_SYNCH_FNC_PROXY`, or `USE_DATA_ASYNC_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ReadDataLength (draft)	
Syntax	Std_ReturnType Xxx_ReadDataLength (Dcm_OpStatusType OpStatus, uint16* DataLength)	
Service ID [hex]	0x4c	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	None	





Parameters (out)	DataLength	Length in byte of the data to be read
Return value	Std_ReturnType	E_OK: this value is always returned. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests the application to return the data length in byte of a Data. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

D.6.1.5 GetScalingInformation

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_GetScalingInformation (draft)	
Syntax	<pre>Std_ReturnType Xxx_GetScalingInformation (uint8* ScalingInfo, uint8 ScalingInfoLength, Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x72	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Variation	$\{ \{ \text{ecuc}(\text{Dcm}/\text{DcmConfigSet}/\text{DcmDsp}/\text{DcmDspData}.\text{DcmDspDataUsePort}) \} == (\text{USE_DATA_SYNCH_FNC_PROXY}) \} \&\& \{ \{ \text{ecuc}(\text{Dcm}/\text{DcmConfigSet}/\text{DcmDsp}/\text{DcmDspData}/\text{DcmDspDataInfoRef} \rightarrow \text{DcmDspDataScalingInfoSize}) \} != \text{NULL} \}$	
Parameters (in)	ScalingInfoLength	Length in byte of the scaling information to be read
Parameters (inout)	None	
Parameters (out)	ScalingInfo	Scaling information (scalingByte and scalingByteExtension)
	ErrorCode	If the operation <code>Xxx_GetScalingInformation</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application for the scaling information of a Data. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC_PROXY`, the following definition is used:

Service Name	Xxx_GetScalingInformation (draft)
---------------------	-----------------------------------





Syntax	<pre>Std_ReturnType Xxx_GetScalingInformation (Dcm_OpStatusType OpStatus, uint8* ScalingInfo, uint8 ScalingInfoLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x73	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)}== (USE_DATA_ ASYNCH_FNC_PROXY) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspData InfoRef->DcmDspDataScalingInfoSize)} != NULL))	
Parameters (in)	OpStatus	Status of the current operation
	ScalingInfoLength	Length in byte of the scaling information to be read
Parameters (inout)	None	
Parameters (out)	ScalingInfo	Scaling information (scalingByte and scalingByteExtension)
	ErrorCode	If the operation Xxx_GetScalingInformation returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application for the scaling information of a Data. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

D.6.1.6 ReturnControlToECU

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC_PROXY` or `USE_DATA_ ASYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ReturnControlToECU (draft)	
Syntax	<pre>Std_ReturnType Xxx_ReturnControlToECU (uint8* controlMask, uint8 controlMaskLength, Dcm_NegativeResponseCodeType ErrorCode)</pre>	
Service ID [hex]	0x74	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)}== (USE_DATA_ SYNCH_FNC_PROXY USE_DATA_ ASYNCH_FNC_PROXY)) && ({ecuc(Dcm/DcmConfig Set/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl)} != NULL)	
Parameters (in)	controlMask	Control enable mask
	controlMaskLength	Control enable mask length in byte
Parameters (inout)	None	





Parameters (out)	ErrorCode	If the operation Xxx_ReturnControlToECU returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application to return control to ECU of an IOControl. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

D.6.1.7 ResetToDefault

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ResetToDefault (draft)	
Syntax	<pre>Std_ReturnType Xxx_ResetToDefault (uint8* controlMask, uint8 controlMaskLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x75	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Variation	<code>{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_SYNCH_FNC_PROXY} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)) == TRUE}</code>	
Parameters (in)	controlMask	Control enable mask
	controlMaskLength	Control enable mask length in byte
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_ResetToDefault returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application to reset an IOControl to default value. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC_ERROR`, the following definition is used:

Service Name	Xxx_ResetToDefault
---------------------	--------------------





Syntax	<pre>Std_ReturnType Xxx_ResetToDefault (Dcm_OpStatusType OpStatus, [Dcm_ControlMask_{DID}]Type controlMask], [uint8* controlMask], Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x3c	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
	controlMask	–
	Type	Dcm_ControlMask_{DID}Type
	Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == (USE_DATA_ASYNC_FNC USE_DATA_ASYNC_FNC_ERROR))&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) <= 0x04}
	controlMask	–
	Type	uint8*
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == (USE_DATA_ASYNC_FNC USE_DATA_ASYNC_FNC_ERROR))&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) => 0x05}	
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation <code>Xxx_ResetToDefault</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	<code>E_OK</code> : Request was successful. <code>E_NOT_OK</code> : Request was not successful. <code>DCM_E_PENDING</code> : Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application to reset an IOControl to default value.	
Available via	Dcm_Externals.h	

Note: Square brackets [] indicate that an argument is optional.

D.6.1.8 FreezeCurrentState

If `DcmDspDataUsePort` is set to `USE_DATA_SYNC_FNC_PROXY`, the following definition is used:

Service Name	Xxx_FreezeCurrentState (draft)	
Syntax	<pre>Std_ReturnType Xxx_FreezeCurrentState (uint8* controlMask, uint8 controlMaskLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x77	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)}== USE_DATA_SYNCH_FNC_PROXY) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreezeCurrentState)} == TRUE)	
Parameters (in)	controlMask	Control enable mask
	controlMaskLength	Control enable mask length in byte
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_FreezeCurrentState returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application to freeze the current state of an IOControl. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC_PROXY`, the following definition is used:

Service Name	Xxx_FreezeCurrentState (draft)	
Syntax	<pre>Std_ReturnType Xxx_FreezeCurrentState (Dcm_OpStatusType OpStatus, uint8* controlMask, uint8 controlMaskLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x78	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)}== USE_DATA_ASYNC_FNC_PROXY) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreezeCurrentState)} == TRUE)	
Parameters (in)	OpStatus	Status of the current operation
	controlMask	Control enable mask
	controlMaskLength	Control enable mask length in byte
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_FreezeCurrentState returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.





Description	This function requests to the application to freeze the current state of an IOControl. Tags: atp.Status=draft
Available via	Dcm_Externals.h

D.6.1.9 ShortTermAdjustment

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ShortTermAdjustment (draft)	
Syntax	<pre>Std_ReturnType Xxx_ShortTermAdjustment (const uint8* ControlStateInfo, uint16 DataLength, uint8* controlMask, uint8 controlMaskLength, Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x79	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Variation	<code>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_SYNCH_FNC_PROXY) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)) == TRUE)</code>	
Parameters (in)	ControlStateInfo	ControlState information contained in the ControlOptionRecord parameter of the InputOutputControlByIdentifier diagnostic request
	DataLength	ControlState information length in byte
	controlMask	Control enable mask
	controlMaskLength	Control enable mask length in byte
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation <code>Xxx_ShortTermAdjustment</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	<code>E_OK</code> : Request was successful. <code>E_NOT_OK</code> : Request was not successful.
Description	This function requests to the application to adjust the IO signal. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC_PROXY`, the following definition is used:

Service Name	Xxx_ShortTermAdjustment (draft)
---------------------	---------------------------------





Syntax	<pre>Std_ReturnType Xxx_ShortTermAdjustment (const uint8* ControlStateInfo, uint16 DataLength, Dcm_OpStatusType OpStatus, uint8* controlMask, uint8 controlMaskLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x83	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)}== USE_DATA_ ASYNCH_FNC_PROXY) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == TRUE)	
Parameters (in)	ControlStateInfo	ControlState information contained in the ControlOptionRecord parameter of the InputOutputControlByIdentifier diagnostic request
	DataLength	ControlState information length in byte
	OpStatus	Status of the current operation
	controlMask	Control enable mask
	controlMaskLength	Control enable mask length in byte
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_ShortTermAdjustment returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application to adjust the IO signal. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

D.6.2 RoutineServices

D.6.2.1 Xxx_Start Operation

Service Name	Xxx_Start (draft)
Syntax	<pre>Std_ReturnType Xxx_Start (Dcm_OpStatusType OpStatus, uint8* dataInOut, uint16* currentDataLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>
Service ID [hex]	0x80
Sync/Async	Asynchronous
Reentrancy	Non Reentrant





Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	dataInOut	Input and output data in the routine control request / response
	currentDataLength	This parameter contains the length in bytes of the dataInOut array. It include fixed length and variable length data.
Parameters (out)	ErrorCode	If the operation Xxx_Start returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. DCM_E_FORCE_RCRRP: application requests the transmission of a response Pending (NRC 0x78)
Description	This function requests to the application to start the execution of a routine. Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

D.6.2.2 Xxx_StartConfirmation Operation

If DcmDspRoutineFncSignature is set to ROUTINE_FNC_NORMAL or ROUTINE_FNC_PROXY, the following definition is used:

Service Name	Xxx_StartConfirmation	
Syntax	Std_ReturnType Xxx_StartConfirmation (Dcm_ConfirmationStatusType ConfirmationStatus)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfirmationStatus	Confirmation status of a StartRoutine request
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function indicates the transmission of a response to a StartRoutine request	
Available via	Dcm_Externals.h	

D.6.2.3 Xxx_Stop Operation

If DcmDspRoutineFncSignature is set to ROUTINE_FNC_PROXY, the following definition is used:

Service Name	Xxx_Stop (draft)
---------------------	------------------





Syntax	<pre>Std_ReturnType Xxx_Stop (Dcm_OpStatusType OpStatus, uint8* dataInOut, uint16* currentDataLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x81	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	dataInOut	Input and output data in the routine control request / response
	currentDataLength	This parameter contains the length in bytes of the dataInOut array. It include fixed length and variable length data.
Parameters (out)	ErrorCode	If the operation Xxx_Stop returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish DCM_E_FORCE_RCRRP: application requests the transmission of a response Pending (NRC 0x78)
Description	This function requests to the application to stop the execution of a routine Tags: atp.Status=draft	
Available via	Dcm_Externals.h	

D.6.2.4 Xxx_StopConfirmation Operation

If `DcmDspRoutineFncSignature` is set to `ROUTINE_FNC_NORMAL` or `ROUTINE_FNC_PROXY`, the following definition is used:

Service Name	Xxx_StopConfirmation	
Syntax	<pre>Std_ReturnType Xxx_StopConfirmation (Dcm_ConfirmationStatusType ConfirmationStatus)</pre>	
Service ID [hex]	0x69	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfirmationStatus	Dcm_ConfirmationStatus Confirmation status of a StopRoutine request
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function indicates the transmission of a response to a StopRoutine request	
Available via	Dcm_Externals.h	

D.6.2.5 Xxx_RequestResults Operation

If `DcmDspRoutineFncSignature` is set to `ROUTINE_FNC_PROXY`, the following definition is used:

Service Name	Xxx_RequestResults (draft)	
Syntax	<pre>Std_ReturnType Xxx_RequestResults (Dcm_OpStatusType OpStatus, uint8* dataInOut, uint16* currentDataLength, Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x82	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	dataInOut	Input and output data in the routine control request / response. If variable length routine input data is used, this parameter contains the length in bytes of the dataInVar array. If variable length routine output data is used, this parameter contains the length in bytes of the dataOutVar parameter.
	currentDataLength	This parameter contains the length in bytes of the dataInOut array. It includes fixed length and variable length data.
Parameters (out)	ErrorCode	If the operation <code>Xxx_RequestResults</code> returns value <code>E_NOT_OK</code> , the Dcm module shall send a negative response with N.
Return value	Std_ReturnType	<p><code>E_OK</code>: Request was successful.</p> <p><code>E_NOT_OK</code>: Request was not successful.</p> <p><code>DCM_E_PENDING</code>: Request is not yet finished. Further call(s) required to finish.</p> <p><code>DCM_E_FORCE_RCRRP</code>: application requests the transmission of a response Pending (NRC 0x78).</p>
Description	<p>This function requests to the application the result of a routine execution.</p> <p>Tags: atp.Status=draft</p>	
Available via	Dcm_Externals.h	

D.6.2.6 Xxx_RequestResultsConfirmation Operation

If `DcmDspRoutineFncSignature` is set to `ROUTINE_FNC_NORMAL` or `ROUTINE_FNC_PROXY`, the following definition is used:

Service Name	Xxx_RequestResultsConfirmation	
Syntax	<pre>Std_ReturnType Xxx_RequestResultsConfirmation (Dcm_ConfirmationStatusType ConfirmationStatus)</pre>	
Service ID [hex]	0x70	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfirmationStatus	Confirmation status of a RequestRoutineResults request
Parameters (inout)	None	





Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function indicates the transmission of a response to a RequestRoutineResults request	
Available via	Dcm_Externals.h	

D.6.3 ServiceRequestNotification

From the point of view of the DCM, the operations has the following signatures:

D.6.3.1 Indication

Service Name	Xxx_Indication	
Syntax	<pre>Std_ReturnType Xxx_Indication (uint8 SID, const uint8* RequestData, uint32 DataSize, uint8 ReqType, uint16 ConnectionId, Dcm_NegativeResponseCodeType* ErrorCode, Dcm_ProtocolType ProtocolType, uint16 TesterSourceAddress)</pre>	
Service ID [hex]	0x65	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SID	–
	RequestData	Complete request data (diagnostic buffer), except the service ID
	DataSize	Number of valid bytes in the RequestData parameter
	ReqType	Addressing type of the request(0=physical request,1=functional request)
	ConnectionId	Unique connection identifier
	ProtocolType	Type of the protocol to be indicated
	TesterSourceAddress	source address of the tester
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_Indication re- turns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. E_REQUEST_NOT_ACCEPTED : Request not accepted
Description	This function indicates to the application that a service is about to be executed and allows the application to reject the execution of the service request	
Available via	Dcm_Externals.h	

D.6.3.2 Confirmation

Service Name	Xxx_Confirmation	
Syntax	<pre>Std_ReturnType Xxx_Confirmation (uint8 SID, uint8 ReqType, uint16 ConnectionId, Dcm_ConfirmationStatusType ConfirmationStatus, Dcm_ProtocolType ProtocolType, uint16 TesterSourceAddress)</pre>	
Service ID [hex]	0x66	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SID	Value of service identifier
	ReqType	Addressing type of the request(0=physical request,1=functional request)
	ConnectionId	Unique connection identifier
	ConfirmationStatus	Confirmation of a successful transmission or a transmission error of a diagnostic service.
	ProtocolType	Type of Dcm Protocol
	TesterSourceAddress	source address of the tester
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function confirms to the application the successful transmission or a transmission error of a diagnostic service.	
Available via	Dcm_Externals.h	

D.7 Dem

D.7.1 Dem_ClearPrestoredFreezeFrame

Service Name	Dem_ClearPrestoredFreezeFrame	
Syntax	<pre>Std_ReturnType Dem_ClearPrestoredFreezeFrame (Dem_EventIdType EventId)</pre>	
Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Clear prestored freeze frame was successful E_NOT_OK: Clear prestored freeze frame failed





Description	<p>Clears a prestored freeze frame of a specific event. This API can only be used through the RTE and therefore no declaration is exported via Dem.h.</p> <p>API Availability: This API will be available only if $\{\{ecuc(Dem/DemConfigSet/DemEventParameter.DemFFPrestorageSupported)\} == true\}$</p>
Available via	Dem.h

D.7.2 Dem_PrestoreFreezeFrame

Service Name	Dem_PrestoreFreezeFrame	
Syntax	<pre>Std_ReturnType Dem_PrestoreFreezeFrame (Dem_EventIdType EventId)</pre>	
Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK Freeze frame prestorage was successful E_NOT_OK Freeze frame prestorage failed
Description	<p>Captures the freeze frame data for a specific event. This API can only be used through the RTE and therefore no declaration is exported via Dem.h.</p> <p>API Availability: This API will be available only if $\{\{ecuc(Dem/DemConfigSet/DemEventParameter.DemFFPrestorageSupported)\} == true\}$</p>	
Available via	Dem.h	

D.7.3 Dem_ResetEventDebounceStatus

Service Name	Dem_ResetEventDebounceStatus	
Syntax	<pre>Std_ReturnType Dem_ResetEventDebounceStatus (Dem_EventIdType EventId, Dem_DebounceResetStatusType DebounceResetStatus)</pre>	
Service ID [hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	DebounceResetStatus	Freeze or reset the internal debounce counter/timer of the specified event.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Only on development error





Description	Control the internal debounce counter/timer by BSW modules and SW-Cs. The event qualification will not be affected by these debounce state changes. This API is available for BSW modules as soon as Dem_Prenit has been completed (refer to SWS_Dem_00438 and SWS_Dem_00167).
Available via	Dem.h

D.7.4 Dem_ResetEventStatus

Service Name	Dem_ResetEventStatus	
Syntax	Std_ReturnType Dem_ResetEventStatus (Dem_EventIdType EventId)	
Service ID [hex]	0x05	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request to reset the event status was successful accepted. E_NOT_OK: Request to reset the event status failed or is not allowed, because the event is already tested in this operation cycle.
Description	Resets the event failed status. This API can only be used through the RTE and therefore no declaration is exported via Dem.h.	
Available via	Dem.h	

D.7.5 Dem_SetEventDisabled

Service Name	Dem_SetEventDisabled	
Syntax	Std_ReturnType Dem_SetEventDisabled (Dem_EventIdType EventId)	
Service ID [hex]	0x51	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK set of event to disabled was successfull. E_NOT_OK set of event disabled failed
Description	Service for reporting the event as disabled to the Dem for the PID \$41 computation. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	





Available via	Dem.h
----------------------	-------

D.7.6 Dem_SetEventStatus

Service Name	Dem_SetEventStatus	
Syntax	<pre>Std_ReturnType Dem_SetEventStatus (Dem_EventIdType EventId, Dem_EventStatusType EventStatus)</pre>	
Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	EventStatus	Monitor test result
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: set of event status was successful E_NOT_OK: Event status setting or processing failed or could not be accepted.
Description	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value. This API will be available only if ({Dem/DemConfigSet/DemEventParameter/DemEventReportingType} == STANDARD_REPORTING)	
Available via	Dem.h	

D.7.7 Dem_GetDTCOfEvent

Service Name	Dem_GetDTCOfEvent	
Syntax	<pre>Std_ReturnType Dem_GetDTCOfEvent (Dem_EventIdType EventId, Dem_DTCFormatType DTCFormat, uint32* DTCOfEvent)</pre>	
Service ID [hex]	0x0d	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	DTCFormat	Defines the output-format of the requested DTC value.
Parameters (inout)	None	
Parameters (out)	DTCOfEvent	Receives the DTC value in respective format returned by this function. If the return value of the function is other than E_OK this parameter does not contain valid data.
Return value	Std_ReturnType	E_OK: get of DTC was successful E_NOT_OK: the call was not successful DEM_E_NO_DTC_AVAILABLE: there is no DTC configured in the requested format





Description	Gets the DTC of an event.
Available via	Dem.h

D.7.8 Dem_GetDebouncingOfEvent

Service Name	Dem_GetDebouncingOfEvent	
Syntax	<pre>Std_ReturnType Dem_GetDebouncingOfEvent (Dem_EventIdType EventId, Dem_DebouncingStateType* DebouncingState)</pre>	
Service ID [hex]	0x9f	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	DebouncingState	Bit 0 Temporarily Defective (corresponds to 0 < FDC < 127) Bit 1 finally Defective (corresponds to FDC = 127) Bit 2 temporarily healed (corresponds to -128 < FDC < 0) Bit 3 Test complete (corresponds to FDC = -128 or FDC = 127) Bit 4 DTR Update (= Test complete && Debouncing complete && enable conditions / storage conditions fulfilled)
Return value	Std_ReturnType	E_OK: get of debouncing status per event state successful E_NOT_OK: get of debouncing per event state failed
Description	Gets the debouncing status of an event. This function shall not be used for EventId with native debouncing within their functions. It is rather for EventIds using debouncing within the Dem.	
Available via	Dem.h	

D.7.9 Dem_GetEventAvailable

Service Name	Dem_GetEventAvailable	
Syntax	<pre>Std_ReturnType Dem_GetEventAvailable (Dem_EventIdType EventId, boolean* AvailableStatus)</pre>	
Service ID [hex]	0xbe	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	AvailableStatus	TRUE if the event is available. FALSE if the event is not available.
Return value	Std_ReturnType	E_OK : Event availability has been obtained. E_NOT_OK : Event availability cannot be obtained.
Description	Get the Event's availability.	
Available via	Dem.h	

D.7.10 Dem_GetEventExtendedDataRecordEx

Service Name	Dem_GetEventExtendedDataRecordEx	
Syntax	<pre>Std_ReturnType Dem_GetEventExtendedDataRecordEx (Dem_EventIdType EventId, uint8 RecordNumber, uint8* DestBuffer, uint16* BufSize)</pre>	
Service ID [hex]	0x6d	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	RecordNumber	Identification of requested Extended data record. Valid values are between 0x01 and 0xEF as defined in ISO14229-1.
Parameters (inout)	BufSize	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer. The function returns the actual number of written data bytes in this parameter.
Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the extended data shall be written to. The format is raw hexadecimal values and contains no header-information.
Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation could not be performed DEM_NO_SUCH_ELEMENT: The requested event data is not currently stored (but the request was valid) OR the requested record number is not supported by the event. DEM_BUFFER_TOO_SMALL: The provided buffer size is too small.
Description	Gets the data of an extended data record by event.	
Available via	Dem.h	

D.7.11 Dem_GetEventFreezeFrameDataEx

Service Name	Dem_GetEventFreezeFrameDataEx	
Syntax	<pre>Std_ReturnType Dem_GetEventFreezeFrameDataEx (Dem_EventIdType EventId, uint8 RecordNumber, uint16 DataId, uint8* DestBuffer, uint16* BufSize)</pre>	
Service ID [hex]	0x6e	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	RecordNumber	This parameter is a unique identifier for a freeze frame record as defined in ISO14229-1. 0xFF means most recent freeze frame record is returned. 0x00 is only supported if the Dem module supports WWH-OBd (refer to DemOBDSupport)





	DataId	This parameter specifies the DID (ISO14229-1) that shall be copied to the destination buffer.
Parameters (inout)	BufSize	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer. The function returns the actual number of written data bytes in this parameter.
Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the freeze frame data record shall be written to. The format is raw hexadecimal values and contains no header-information.
Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation could not be performed DEM_NO_SUCH_ELEMENT: The requested event data is not currently stored (but the request was valid) OR The requested record number is not supported by the event OR The requested DID is not supported by the freeze frame. DEM_BUFFER_TOO_SMALL: The provided buffer size is too small.
Description	Gets the data of a freeze frame by event.	
Available via	Dem.h	

D.7.12 Dem_GetEventUdsStatus

Service Name	Dem_GetEventUdsStatus	
Syntax	<pre>Std_ReturnType Dem_GetEventUdsStatus (Dem_EventIdType EventId, Dem_UdsStatusByteType* UDSStatusByte)</pre>	
Service ID [hex]	0xb6	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	UDSStatusByte	UDS DTC status byte of the requested event (refer to chapter "Status bit support"). If the return value of the function call is E_NOT_OK, this parameter does not contain valid data.
Return value	Std_ReturnType	E_OK: get of event status was successful E_NOT_OK: get of event status failed
Description	Gets the current UDS status byte assigned to the DTC for the event	
Available via	Dem.h	

D.7.13 Dem_GetFaultDetectionCounter

Service Name	Dem_GetFaultDetectionCounter	
Syntax	<pre>Std_ReturnType Dem_GetFaultDetectionCounter (Dem_EventIdType EventId, sint8* FaultDetectionCounter)</pre>	





Service ID [hex]	0x3e	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	FaultDetectionCounter	This parameter receives the Fault Detection Counter information of the requested EventId. If the return value of the function call is other than E_OK this parameter does not contain valid data. -128dec...127dec PASSED...FAILED according to ISO 14229-1
Return value	Std_ReturnType	E_OK: request was successful E_NOT_OK: request failed DEM_E_NO_FDC_AVAILABLE: there is no fault detection counter available for the requested event
Description	Gets the fault detection counter of an event. This API can only be used through the RTE, and therefore no declaration is exported via Dem.h.	
Available via	Dem.h	

D.7.14 Dem_GetMonitorStatus

Service Name	Dem_GetMonitorStatus	
Syntax	<pre>Std_ReturnType Dem_GetMonitorStatus (Dem_EventIdType EventID, Dem_MonitorStatusType* MonitorStatus)</pre>	
Service ID [hex]	0xb5	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventID	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	MonitorStatus	Monitor status byte of the requested event. If the return value of the function call is E_NOT_OK, this parameter does not contain valid data.
Return value	Std_ReturnType	E_OK: Get monitor status was successful, E_NOT_OK: getting the monitor status failed (e.g. an invalid event id was provided).
Description	Gets the current monitor status for an event.	
Available via	Dem.h	

D.7.15 Dem_SetEventStatusWithMonitorData

Service Name	Dem_SetEventStatusWithMonitorData	
---------------------	-----------------------------------	--





Syntax	Std_ReturnType Dem_SetEventStatusWithMonitorData (Dem_EventIdType EventId, Dem_EventStatusType EventStatus, Dem_MonitorDataType monitorData0, Dem_MonitorDataType monitorData1)	
Service ID [hex]	0xbd	
Sync/Async	Synchronous/Asynchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	EventStatus	Monitor test result
	monitorData0	–
	monitorData1	–
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: set of event status was successful E_NOT_OK: Event status setting or processing failed or could not be accepted.
Description	This API will be available only if ((Dem/DemConfigSet/DemEventParameter/DemEventReportingType) == STANDARD_REPORTING_WITH_MONITOR_DATA)	
Available via	Dem.h	

D.7.16 Dem_SetWIRStatus

Service Name	Dem_SetWIRStatus	
Syntax	Std_ReturnType Dem_SetWIRStatus (Dem_EventIdType EventId, boolean WIRStatus)	
Service ID [hex]	0x7a	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId. The Event Number is configured in the DEM. Min.: 1 (0: Indication of no Event or Failure) Max.:Result of configuration of Event Numbers in DEM (Max is either 255 or 65535)
	WIRStatus	Requested status of event related WIR-bit (regarding to the current status of function inhibition) WIRStatus = TRUE -> WIR-bit shall be set to "1" WIRStatus = FALSE -> WIR-bit shall be set to "0"
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request to set the WIR status was successful. E_NOT_OK: Request to set the WIR status was not accepted (e.g. disabled controlDTCSetting) and should be repeated.
Description	Sets the WIR status bit via failsafe SW-Cs. This API can only be used through the RTE and therefore no declaration is exported via Dem.h.	
Available via	Dem.h	

D.7.17 <Module>_DemGetFaultDetectionCounter<ForEvent>

Service Name	<Module>_DemGetFaultDetectionCounter<ForEvent>	
Syntax	Std_ReturnType <Module>_DemGetFaultDetectionCounter<ForEvent> (sint8* FaultDetectionCounter)	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	FaultDetectionCounter	This parameter receives the fault detection counter information of the requested EventId. If the return value of the function call is other than E_OK this parameter does not contain valid data. -128dec...127dec PASSED...FAILED according to ISO 14229-1
Return value	Std_ReturnType	E_OK: request was successful E_NOT_OK: request failed
Description	Gets the current fault detection counter value. There is one c-callback per event using monitor-internal debouncing, if no port interface is provided by the Dem.	
Available via	Dem_Externals.h	

D.7.18 <Module>_DemInitMonitorFor<EventName>

Service Name	<Module>_DemInitMonitorFor<EventName>	
Syntax	Std_ReturnType <Module>_DemInitMonitorFor<EventName> (Dem_InitMonitorReasonType InitMonitorReason)	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	InitMonitorReason	Specific (re-)initialization reason evaluated from the monitor to identify the initialization kind to be performed.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
Description	Inits the diagnostic monitor of a specific event. There is one separate callback per event (if configured), if no port interface is provided by the Dem.	
Available via	Dem_Externals.h	

D.7.19 <Module>_DemTriggerOnMonitorStatus

Service Name	<Module>_DemTriggerOnMonitorStatus	
Syntax	Std_ReturnType <Module>_DemTriggerOnMonitorStatus (void)	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this function always returns E_OK for compatibility reasons with RTE
Description	Triggers on changes of the monitor status. Called synchronously in context of event status reporting.	
Available via	Dem_Externals.h	

D.7.20 <Module>_DemClearEventAllowed<ForCondition>

Service Name	<Module>_DemClearEventAllowed<ForCondition>	
Syntax	Std_ReturnType <Module>_DemClearEventAllowed<ForCondition> (boolean* Allowed)	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	Allowed	True - clearance of event is allowed False - clearance of event is not allowed
Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
Description	Triggers on DTC-deletion, which is not allowed if the out-parameter returns False. There is one separate callback per condition, which can be assigned to one or several events, if no port interface is provided by the Dem. Parameter "Allowed" will be unchanged in case E_NOT_OK is returned.	
Available via	Dem_Externals.h	

D.7.21 <Module>_DemTriggerOnEventData

Service Name	<Module>_DemTriggerOnEventData	
Syntax	Std_ReturnType <Module>_DemTriggerOnEventData (Dem_EventIdType EventId)	
Sync/Async	Synchronous	





Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this function always returns E_OK for compatibility reasons with RTE.
Description	Triggers on changes of the event related data in the event memory.	
Available via	Dem_Externals.h	

D.7.22 <Module>_DemTriggerOnEventUdsStatus

Service Name	<Module>_DemTriggerOnEventUdsStatus	
Syntax	<pre>Std_ReturnType <Module>_DemTriggerOnEventUdsStatus (Dem_UdsStatusByteType EventStatusByteOld, Dem_UdsStatusByteType EventStatusByteNew)</pre>	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventStatusByteOld	UDS DTC status byte of event before change (refer to chapter "Status bit support").
	EventStatusByteNew	UDS DTC status byte of event after change (refer to chapter "Status bit support").
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this function always returns E_OK for compatibility reasons with RTE
Description	Triggers on changes of the UDS DTC status byte.	
Available via	Dem_Externals.h	

D.8 FiM

D.8.1 FiM_GetFunctionPermission

Service Name	FiM_GetFunctionPermission	
Syntax	<pre>Std_ReturnType FiM_GetFunctionPermission (FiM_FunctionIdType FID, boolean* Permission)</pre>	
Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Reentrant	





Parameters (in)	FID	Identification of a functionality by assigned FID. The FunctionId is configured in the FIM. Min.: 1 (0: Indication of no functionality) Max.: Result of configuration of FIDs in FIM (Max is either 255 or 65535)
Parameters (inout)	None	
Parameters (out)	Permission	TRUE: FID has permission to run FALSE: FID has no permission to run, i.e. shall not be executed
Return value	Std_ReturnType	E_OK: The request is accepted E_NOT_OK: The request is not accepted, ie. initialization of FIM not completed
Description	This service reports the permission state to the functionality.	
Available via	FiM.h	

D.8.2 FiM_SetFunctionAvailable

Service Name	FiM_SetFunctionAvailable	
Syntax	<pre>Std_ReturnType FiM_SetFunctionAvailable (FiM_FunctionIdType FID, boolean Availability)</pre>	
Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	FID	Identification of a functionality by assigned FID.
	Availability	The permission of the requested FID: TRUE: Function is available. FALSE: Function is not available.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The request is accepted E_NOT_OK: Request is not accepted (e.g. invalid FID is given)
Description	This service sets the availability of a function. The function is only available if FiMAvailability Support is configured as True.	
Available via	FiM.h	

E Referenced Service Interfaces

E.1 Os

E.2 NvM

E.2.1 NvM_BlockIdType

Name	NvM_BlockIdType		
Kind	Type		
Derived from	uint16		
Range	0..2 ^(16- NvMDatasetSelection Bits) -1	–	–
Description	Identification of a NVRAM block via a unique block identifier. Reserved NVRAM block IDs: 0 -> to derive multi block request results via NvM_GetErrorStatus 1 -> redundant NVRAM block which holds the configuration ID		
Variation	–		
Available via	Rte_NvM_Type.h		

E.2.2 NvM_BlockRequestType

Name	NvM_BlockRequestType		
Kind	Type		
Derived from	uint8		
Range	NVM_READ_BLOCK	0x00	NvM_ReadBlock/ NvM_Read PRAMBlock was performed on the block
	NVM_WRITE_BLOCK	0x01	NvM_WriteBlock/ NvM_Write PRAMBlock was performed on the block
	NVM_RESTORE_BLOCK_DEFAULTS	0x02	NvM_RestoreBlockDefaults/ NvM_RestorePRAMBlockDefaults was performed on the block
	NVM_ERASE_NV_BLOCK	0x03	NvM_EraseNvBlock was performed on the block
	NVM_INVALIDATE_NV_BLOCK	0x04	NvM_InvalidateNvBlock was performed on the block
	NVM_READ_ALL_BLOCK	0x05	NvM_ReadAll has finished processing this block
Description	Identifies the type of request performed on a block when signaled via the callback function		
Variation	–		
Available via	Rte_NvM_Type.h		

E.2.3 NvM_InitBlockRequestType

Name	NvM_InitBlockRequestType		
Kind	Type		
Derived from	uint8		
Range	NVM_INIT_READ_BLOCK	0x00	NvM_ReadBlock/ NvM_ReadPRAMBlock is requested on the block
	NVM_INIT_RESTORE_BLOCK_DEFAULTS	0x01	NvM_RestoreBlockDefaults/ NvM_RestorePRAMBlockDefaults is requested on the block
	NVM_INIT_READ_ALL_BLOCK	0x02	NvM_ReadAll is processing this block
	NVM_INIT_FIRST_INIT_ALL	0x03	NvM_FirstInitAll is processing this block
Description	Identifies the type of request performed on a block when signaled via the callback function		
Variation	-		
Available via	Rte_NvM_Type.h		

E.2.4 NvM_RequestResultType

Name	NvM_RequestResultType		
Kind	Type		
Derived from	uint8		
Range	NVM_REQ_OK	0x00	The last asynchronous request has been finished successfully. This shall be the default value after reset. This status shall have the value 0.
	NVM_REQ_NOT_OK	0x01	The last asynchronous read/write/control request has been finished unsuccessfully.
	NVM_REQ_PENDING	0x02	An asynchronous read/write/control request is currently pending.
	NVM_REQ_INTEGRITY_FAILED	0x03	The result of the last asynchronous request NvM_ReadBlock or NvM_ReadAll is a data integrity failure. Note: In case of NvM_ReadBlock the content of the RAM block has changed but has become invalid. The application is responsible to renew and validate the RAM block content.





	NVM_REQ_BLOCK_SKIPPED	0x04	The referenced block was skipped during execution of NvM_ReadAll or NvM_WriteAll, e.g. Dataset NVRAM blocks (NvM_ReadAll) or NVRAM blocks without a permanently configured RAM block.
	NVM_REQ_NV_INVALIDATED	0x05	The referenced NV block is invalidated.
	NVM_REQ_CANCELED	0x06	The multi block request NvM_WriteAll was canceled by calling NvM_CancelWriteAll. Or Any single block job request (NvM_ReadBlock, NvM_WriteBlock, NvM_EraseNvBlock, NvM_InvalidateNvBlock and NvM_RestoreBlock Defaults) was canceled by calling NvM_CancelJobs.
	NVM_REQ_RESTORED_DEFAULTS	0x08	The referenced NV block had the default values copied to the RAM image.
Description	This is an asynchronous request result returned by the API service NvM_GetErrorStatus. The availability of an asynchronous request result can be additionally signaled via a callback function.		
Variation	-		
Available via	Rte_NvM_Type.h		

E.2.5 NvMService

Name	NvMService		
Comment	-		
IsService	true		
Variation	-		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	EraseBlock
Comment	Service to erase a NV block.
Variation	<pre> FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass3; </pre>
Possible Errors	E_OK E_NOT_OK

Operation	GetDataIndex
Comment	Service for getting the currently set DataIndex of a dataset NVRAM block





Variation	<pre>FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass2 = configClass.value() == "NVM_API_CONFIG_CLASS_2"; WHERE isConfigClass2;</pre>	
Parameters	DataIndex	
	Type	uint8
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetErrorStatus	
Comment	Service to read the block dependent error/status information.	
Variation	–	
Parameters	RequestResult	
	Type	NvM_RequestResultType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	InvalidateNvBlock	
Comment	Service to invalidate a NV block.	
Variation	<pre>FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass3;</pre>	
Possible Errors	E_OK E_NOT_OK	

Operation	ReadBlock	
Comment	Service to copy the data of the NV block to its corresponding RAM block.	
Variation	<pre>FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass2 = configClass.value() == "NVM_API_CONFIG_CLASS_2"; isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass2 OR isConfigClass3;</pre>	
Parameters	DstPtr	
	Type	VoidPtr
	Direction	IN





	Comment	The parameter "DstPtr" shall be typed by an ImplementationDataType of category DATA_REFERENCE with the pointer target void to pass an address (pointer) to the RAM Block.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	ReadPRAMBlock
Comment	–
Variation	<pre>FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass2 = configClass.value() == "NVM_API_CONFIG_CLASS_2"; isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass2 OR isConfigClass3;</pre>
Possible Errors	E_OK E_NOT_OK

Operation	RestoreBlockDefaults										
Comment	Service to restore the default data to its corresponding RAM block.										
Variation	<pre>FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass2 = configClass.value() == "NVM_API_CONFIG_CLASS_2"; isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass2 OR isConfigClass3;</pre>										
Parameters	<table border="1"> <tr> <td colspan="2">DstPtr</td> </tr> <tr> <td>Type</td> <td>VoidPtr</td> </tr> <tr> <td>Direction</td> <td>IN</td> </tr> <tr> <td>Comment</td> <td>The parameter "DstPtr" shall be typed by an ImplementationDataType of category DATA_REFERENCE with the pointer target void to pass an address (pointer) to the RAM Block.</td> </tr> <tr> <td>Variation</td> <td>–</td> </tr> </table>	DstPtr		Type	VoidPtr	Direction	IN	Comment	The parameter "DstPtr" shall be typed by an ImplementationDataType of category DATA_REFERENCE with the pointer target void to pass an address (pointer) to the RAM Block.	Variation	–
DstPtr											
Type	VoidPtr										
Direction	IN										
Comment	The parameter "DstPtr" shall be typed by an ImplementationDataType of category DATA_REFERENCE with the pointer target void to pass an address (pointer) to the RAM Block.										
Variation	–										
Possible Errors	E_OK E_NOT_OK										

Operation	RestorePRAMBlockDefaults
Comment	–
Variation	<pre>FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass2 = configClass.value() == "NVM_API_CONFIG_CLASS_2"; isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass2 OR isConfigClass3;</pre>
Possible Errors	E_OK E_NOT_OK

Operation	SetDataIndex	
Comment	Service for setting the DataIndex of a dataset NVRAM block.	
Variation	<pre> FOR configClass : ECV.subEltList("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass2 = configClass.value() == "NVM_API_CONFIG_CLASS_2"; isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; blockMgmTypes = ECV.subEltList("NvM/NvMBlockDescriptor/ NvMBlockManagementType"); isMgd(mgmtType) = mgmtType.value() == "NVM_BLOCK_DATASET"; datasetMgdCount = blockMgmTypes.filter(isMgd).count(); WHERE (isConfigClass2 OR isConfigClass3) AND (datasetMgdCount GT 0); </pre>	
Parameters	DataIndex	
	Type	uint8
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	SetRamBlockStatus	
Comment	Service for setting the RAM block status of an NVRAM block.	
Variation	<pre> LET nvmBlockUseSetRamBlockStatus = ECV.subEltList("NvM/ NvMBlockDescriptor/NvMBlockUseSetRamBlockStatus"); useSetRamBlockStatus(useApi) = useApi.value() == true; useSetRamBlockStatusCount = nvmBlockUseSetRamBlockStatus.filter(useSetRamBlockStatus).count(); WHERE (useSetRamBlockStatusCount GT 0); </pre>	
Parameters	BlockChanged	
	Type	boolean
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	WriteBlock	
Comment	Service to copy the data of the RAM block to its corresponding NV block.	
Variation	<pre> FOR configClass : ECV.subEltList("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass2 = configClass.value() == "NVM_API_CONFIG_CLASS_2"; isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass2 OR isConfigClass3; </pre>	
Parameters	SrcPtr	
	Type	ConstVoidPtr
	Direction	IN





	Comment	The parameter "SrcPtr" shall be typed by an ImplementationDataType of category DATA_REFERENCE with the pointer target void to pass an address (pointer) to the RAM Block.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	WritePRAMBlock
Comment	–
Variation	<pre>FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass2 = configClass.value() == "NVM_API_CONFIG_CLASS_2"; isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass2 OR isConfigClass3;</pre>
Possible Errors	E_OK E_NOT_OK

E.2.6 NvMAdmin

Name	NvMAdmin		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetBlockProtection	
Comment	Service for setting/resetting the write protection for a NV block.	
Variation	<pre>FOR configClass : ECV.subEltList ("NvM/NvMCommon/NvMApiConfigClass"); LET isConfigClass3 = configClass.value() == "NVM_API_CONFIG_CLASS_3"; WHERE isConfigClass3;</pre>	
Parameters	ProtectionEnabled	
	Type	boolean
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

E.2.7 NvMNotifyJobFinished

Name	NvMNotifyJobFinished		
Comment	Callback that is called when a job has finished		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	JobFinished		
Comment	Callback that gets called if a job has finished		
Variation	–		
Parameters	BlockRequest		
	Type	NvM_BlockRequestType	
	Direction	IN	
	Comment	–	
	Variation	–	
	JobResult		
	Type	NvM_RequestResultType	
	Direction	IN	
Comment	–		
Variation	–		
Possible Errors	E_OK		

E.2.8 NvMNotifyInitBlock

Name	NvMNotifyInitBlock		
Comment	Callback that is called by the NvM module when default data needs to be restored to the RAM image		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	RAM block content was updated
	1	RTE_E_RAM_UNCHANGED	RAM block content was not changed

Operation	InitBlock		
Comment	This callback is called if the initialization of a block has completed.		
Variation	–		
Parameters	InitBlockRequest		
	Type	NvM_InitBlockRequestType	
	Direction	IN	
	Comment	–	
	Variation	–	
Possible Errors	–		

E.2.9 NvMMirror

Name	NvMMirror		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	ReadRamBlockFromNvM		
Comment	Block specific callback routine which shall be called in order to let the application copy data from NvM module's mirror to RAM block.		
Variation	–		
Parameters	SrcPtr		
	Type	ConstVoidPtr	
	Direction	IN	
	Comment	The parameter "SrcPtr" shall be typed by an ImplementationDataType of category DATA_REFERENCE with the pointer target void to pass an address (pointer) to the RAM Block.	
Variation	–		
Possible Errors	E_OK E_NOT_OK		

Operation	WriteRamBlockToNvM		
Comment	Block specific callback routine which shall be called in order to let the application copy data from RAM block to NvM module's mirror.		
Variation	–		
Parameters	DstPtr		
	Type	VoidPtr	
	Direction	IN	
	Comment	The parameter "DstPtr" shall be typed by an ImplementationDataType of category DATA_REFERENCE with the pointer target void to pass an address (pointer) to the RAM Block.	
Variation	–		
Possible Errors	E_OK E_NOT_OK		

E.2.10 PS_{Block}

Name	PS_{Block}		
Kind	ProvidedPort	Interface	NvMService
Description	–		
Port Defined Argument Value(s)	Type	NvM_BlockIdType	





	Value	FOR nvBlockDescriptor : ECV.subEltList("NvM/ NvMBlockDescriptor"); LET Block = nvBlockDescriptor.shortname(); BlockId = nvBlockDescriptor.subElt("NvMNvramBlockIdentifier").value();
Variation		FOR nvBlockDescriptor : ECV.subEltList("NvM/NvMBlockDescriptor"); LET Block = nvBlockDescriptor.shortname(); UsePort = nvBlockDescriptor.subElt("NvMBlockUsePort").value() == true; WHERE UsePort;

E.2.11 PAdmin_{Block}

Name	PAdmin_{Block}		
Kind	ProvidedPort	Interface	NvMAdmin
Description	-		
Port Defined Argument Value(s)	Type	NvM_BlockIdType	
	Value	FOR nvBlockDescriptor : ECV.subEltList("NvM/ NvMBlockDescriptor"); LET Block = nvBlockDescriptor.shortname(); BlockId = nvBlockDescriptor.subElt("NvMNvramBlockIdentifier").value();	
Variation	FOR nvBlockDescriptor : ECV.subEltList("NvM/NvMBlockDescriptor"); LET Block = nvBlockDescriptor.shortname(); UsePort = nvBlockDescriptor.subElt("NvMBlockUsePort").value() == true; WHERE UsePort;		

E.2.12 PNJF_{Block}

Name	PNJF_{Block}		
Kind	RequiredPort	Interface	NvMNotifyJobFinished
Description	-		





Variation	<pre> FOR nvBlockDescriptor : ECV.subEltList("NvM/NvMBlockDescriptor"); LET Block = nvBlockDescriptor.shortname(); UsePort = nvBlockDescriptor.subElt("NvMBlockUsePort"). value() == true; SingleBlockCallbackDef = nvBlockDescriptor.subElt(" NvMSingleBlockCallback").isDefined(); SingleBlockCallbackFncDef = nvBlockDescriptor.subElt(" NvMSingleBlockCallback/NvMSingleBlockCallbackFnc").isDefined(); WHERE UsePort AND SingleBlockCallbackDef AND NOT SingleBlockCallbackFncDef; </pre>
------------------	---

E.2.13 PNIB_{Block}

Name	PNIB_{Block}		
Kind	RequiredPort	Interface	NvMNotifyInitBlock
Description	-		
Variation	<pre> FOR nvBlockDescriptor : ECV.subEltList("NvM/NvMBlockDescriptor"); LET Block = nvBlockDescriptor.shortname(); UsePort = nvBlockDescriptor.subElt("NvMBlockUsePort"). value() == true; InitBlockCallbackDef = nvBlockDescriptor.subElt(" NvMInitBlockCallback").isDefined(); InitBlockCallbackFncDef = nvBlockDescriptor.subElt(" NvMInitBlockCallback/NvMInitBlockCallbackFnc").isDefined(); WHERE UsePort AND InitBlockCallbackDef AND NOT InitBlockCallbackFncDef; </pre>		

E.2.14 PM_{Block}

Name	PM_{Block}		
Kind	RequiredPort	Interface	NvMMirror
Description	-		
Variation	<pre> FOR nvBlockDescriptor : ECV.subEltList("NvM/NvMBlockDescriptor"); LET Block = nvBlockDescriptor.shortname(); UsePort = nvBlockDescriptor.subElt("NvMBlockUsePort").value() == true; UsePortSyncMech = nvBlockDescriptor.subElt("NvMBlockUseSyncMechanism") .value() == true; WHERE UsePort AND UsePortSyncMech; </pre>		

E.3 Dcm

E.3.1 Implementation Data Types

E.3.1.1 Dcm_OpStatusType

Name	Dcm_OpStatusType		
Kind	Type		
Derived from	uint8		
Range	DCM_INITIAL	0x00	Indicates the initial call to the operation
	DCM_PENDING	0x01	Indicates that a pending return has been done on the previous call of the operation
	DCM_CANCEL	0x02	Indicates that the DCM requests to cancel the pending operation
	DCM_FORCE_RCRRP_OK	0x03	Confirm a response pending transmission
Description	–		
Variation	–		
Available via	Rte_Dcm_Type.h		

E.3.1.2 Dcm_ConfirmationStatusType

Name	Dcm_ConfirmationStatusType		
Kind	Type		
Derived from	uint8		
Range	DCM_RES_POS_OK	0x00	–
	DCM_RES_POS_NOT_OK	0x01	–
	DCM_RES_NEG_OK	0x02	–
	DCM_RES_NEG_NOT_OK	0x03	–
Description	–		
Variation	–		
Available via	Rte_Dcm_Type.h		

Name	Dcm_ProtocolType		
Kind	Type		
Derived from	uint8		
Range	DCM_OBD_ON_CAN	0x00	OBD on CAN (ISO15765-4; ISO15031-5)
	DCM_OBD_ON_FLEXRAY	0x01	(OBD on Flexray (Manufacturer specific; ISO15031-5))





DCM_OBD_ON_IP	0x02	(OBD on Internet Protocol (Manufacturer specific; ISO15031-5))
DCM_UDS_ON_CAN	0x03	UDS on CAN (ISO15765-3; ISO14229-1)
DCM_UDS_ON_FLEXRAY	0x04	UDS on FlexRay (Manufacturer specific; ISO14229-1)
DCM_UDS_ON_IP	0x05	(UDS on Internet Protocol (Manufacturer specific; ISO14229-1))
DCM_ROE_ON_CAN	0x06	Response On Event on CAN
DCM_ROE_ON_FLEXRAY	0x07	Response On Event on FlexRay
DCM_ROE_ON_IP	0x08	(Response on Event on Internet Protocol)
DCM_PERIODICTRANS_ON_CAN	0x09	Periodic Transmission on CAN
DCM_PERIODICTRANS_ON_FLEXRAY	0x0A	Periodic Transmission on FlexRay
DCM_PERIODICTRANS_ON_IP	0x0B	(Periodic Transmission on Internet Protocol)
DCM_NO_ACTIVE_PROTOCOL	0x0C	No protocol has been started
DCM_UDS_ON_LIN	0x0D	UDS on LIN (ISO14229-1; ISO14229-7)
Reserved for further AUTOSAR implementation	0x0E..0xEF	–
DCM_SUPPLIER_1	0xF0	Reserved for SW supplier specific.
DCM_SUPPLIER_2	0xF1	Reserved for SW supplier specific.
DCM_SUPPLIER_3	0xF2	Reserved for SW supplier specific.
DCM_SUPPLIER_4	0xF3	Reserved for SW supplier specific.
DCM_SUPPLIER_5	0xF4	Reserved for SW supplier specific.
DCM_SUPPLIER_6	0xF5	Reserved for SW supplier specific.
DCM_SUPPLIER_7	0xF6	Reserved for SW supplier specific.
DCM_SUPPLIER_8	0xF7	Reserved for SW supplier specific.
DCM_SUPPLIER_9	0xF8	Reserved for SW supplier specific.
DCM_SUPPLIER_10	0xF9	Reserved for SW supplier specific.
DCM_SUPPLIER_11	0xFA	Reserved for SW supplier specific.
DCM_SUPPLIER_12	0xFB	Reserved for SW supplier specific.
DCM_SUPPLIER_13	0xFC	Reserved for SW supplier specific.





	DCM_SUPPLIER_14	0xFD	Reserved for SW supplier specific.
	DCM_SUPPLIER_15	0xFE	Reserved for SW supplier specific.
Description	Protocol type definition		
Variation	–		
Available via	Rte_Dcm_Type.h		

E.3.1.3 Dcm_NegativeResponseCodeType

Name	Dcm_NegativeResponseCodeType		
Kind	Type		
Derived from	uint8		
Range	DCM_POS_RESP	0x00	PR
	range of values 0x01..0x0F reserved by ISO 14229	0x01..0x0F	ISOSAERESRVD
	DCM_E_GENERALREJECT	0x10	GR
	DCM_E_SERVICENOTSUPPORTED	0x11	SNS
	DCM_E_SUBFUNCTIONNOT-SUPPORTED	0x12	SFNS
	DCM_E_INCORRECTMESSAGE-LENGTHORINVALIDFORMAT	0x13	IMLOIF
	DCM_E_RESPONSETOOLONG	0x14	RTL
	range of values 0x15..0x20 reserved by ISO 14229	0x15..0x20	ISOSAERESRVD
	DCM_E_BUSYREPEATREQUEST	0x21	BRR
	DCM_E_CONDITIONSNOTCORRECT	0x22	CNC
	value 0x23 reserved by ISO 14229	0x23	ISOSAERESRVD
	DCM_E_REQUESTSEQUENCEERROR	0x24	RSE
	DCM_E_NORESPONSEFROM-SUBNETCOMPONENT	0x25	NRFSC
	DCM_E_FAILUREPREVENTSEXCUTIONOFREQUESTEDACTION	0x26	FPEORA
	range of values 0x27..0x30 reserved by ISO 14229	0x27..0x30	ISOSAERESRVD
	DCM_E_REQUESTOUTOFRANGE	0x31	ROOR
	value 0x32 reserved by ISO 14229	0x32	ISOSAERESRVD
	DCM_E_SECURITYACCESSDENIED	0x33	SAD
value 0x34 reserved by ISO 14229	0x34	ISOSAERESRVD	





DCM_E_INVALIDKEY	0x35	IK
DCM_E_EXCEEDNUMBEROFATTEMPTS	0x36	ENOA
DCM_E_REQUIREDTIMEDELAYNOTEXPIRED	0x37	RTDNE
range of values 0x38..0x4F reserved by ISO 15764	0x38..0x4F	RBEDLSD
range of values 0x50..0x6F reserved by ISO 14229	0x50..0x6F	ISOSAERESRVD
DCM_E_UPLOADDOWNLOADNOTACCEPTED	0x70	UDNA
DCM_E_TRANSFERDATASUSPENDED	0x71	TDS
DCM_E_GENERALPROGRAMMINGFAILURE	0x72	GPF
DCM_E_WRONGBLOCKSEQUENCECOUNTER	0x73	WBSC
range of values 0x74..0x77 reserved by ISO 14229	0x74..0x77	ISOSAERESRVD
range of values 0x79..0x7D reserved by ISO 14229	0x79..0x7D	ISOSAERESRVD
DCM_E_SUBFUNCTIONNOTSUPPORTEDINACTIVESSESSION	0x7E	SFNSIAS
DCM_E_SERVICENOTSUPPORTEDINACTIVESSESSION	0x7F	SNSIAS
value 0x80 reserved by ISO 14229	0x80	ISOSAERESRVD
DCM_E_RPMTOOHIGH	0x81	RPMTL
DCM_E_RPMTOOLOW	0x82	RPMTL
DCM_E_ENGINEISRUNNING	0x83	EIR
DCM_E_ENGINEISNOTRUNNING	0x84	EINR
DCM_E_ENGINERUNTIMETOLOW	0x85	ERTTL
DCM_E_TEMPERATURETOOHIGH	0x86	TEMPH
DCM_E_TEMPERATURETOOLOW	0x87	TEMPH
DCM_E_VEHICLESPEEDTOOHIGH	0x88	VSTH
DCM_E_VEHICLESPEEDTOOLOW	0x89	VSTL
DCM_E_THROTTLE_PEDALTOOHIGH	0x8A	TPTH
DCM_E_THROTTLE_PEDALTOOLOW	0x8B	TPTL
DCM_E_TRANSMISSION-RANGENOTINNEUTRAL	0x8C	TRNIN
DCM_E_TRANSMISSION-RANGENOTINGEAR	0x8D	TRNIG
value 0x8E reserved by ISO 14229	0x8E	ISOSAERESRVD





	DCM_E_BRAKESWITCH_NOTCLOSED	0x8F	BSNC
	DCM_E_SHIFTERLEVERNOTINPARK	0x90	SLNIP
	DCM_E_TORQUECONVERTER-CLUTCHLOCKED	0x91	TCCL
	DCM_E_VOLTAGETOOHIGH	0x92	VTH
	DCM_E_VOLTAGETOLOW	0x93	VTL
	range of values 0x94..0xEF reserved by ISO 14229	0x94..0xEF	RFSCNC
	DCM_E_VMSCNC_0	0xF0	VMSCNC
	DCM_E_VMSCNC_1	0xF1	VMSCNC1
	DCM_E_VMSCNC_2	0xF2	VMSCNC2
	DCM_E_VMSCNC_3	0xF3	VMSCNC3
	DCM_E_VMSCNC_4	0xF4	VMSCNC4
	DCM_E_VMSCNC_5	0xF5	VMSCNC5
	DCM_E_VMSCNC_6	0xF6	VMSCNC6
	DCM_E_VMSCNC_7	0xF7	VMSCNC7
	DCM_E_VMSCNC_8	0xF8	VMSCNC8
	DCM_E_VMSCNC_9	0xF9	VMSCNC9
	DCM_E_VMSCNC_A	0xFA	VMSCNCA
	DCM_E_VMSCNC_B	0xFB	VMSCNCB
	DCM_E_VMSCNC_C	0xFC	VMSCNCC
	DCM_E_VMSCNC_D	0xFD	VMSCNCD
	DCM_E_VMSCNC_E	0xFE	VMSCNCE
	value 0xFF reserved by ISO 14229	0xFF	ISOSAERESRVD
Description	This Table of available Negative Response Codes represents the allowed Response Codes an AUTOSAR SW Component shall return after a function call. For the allowed NRC of the executed Service ID please refer to the specification of the service in ISO14229-1 (UDS) and ISO15031-5 (OBD/CARB) (see chapter 4.2.4 Response code parameter definition Table 12).		
Variation	-		
Available via	Rte_Dcm_Type.h		

E.3.1.4 Dcm_DataElementType_{Data}Type

Name	Dcm_DataElement_{Data}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_DataElement_{Data}_Array Type	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == ([S U]INT[8 16 32] FLOAT)_N) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == ([S U]INT[8 16 32] FLOAT)_N))





	Dcm_DataElement_{Data}_PrimitiveType	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == (BOOLEAN [S U]INT[8 16 32] FLOAT)) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == (BOOLEAN [S U]INT[8 16 32] FLOAT)))
Description	Common description for S/R and C/S data elements.	
Variation	<pre>((Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort == USE_DATA_ELEMENT_SPECIFIC_INTERFACES) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == USE_DATA_SENDER_RECEIVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == USE_DATA_SENDER_RECEIVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNCH_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNCH_CLIENT_SERVER_ERROR) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER))) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})</pre>	
Available via	Rte_Dcm_Type.h	

E.3.1.5 Dcm_DataElementType_{Data}ArrayType

Name	Dcm_DataElement_{Data}_ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == FLOAT_N) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == FLOAT_N))
	sint16	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == SINT16_N) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == SINT16_N))
	sint32	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == SINT32_N) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == SINT32_N))
	sint8	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == SINT8_N) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == SINT8_N))
	uint16	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == UINT16_N) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == UINT16_N))
uint32	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == UINT32_N) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == UINT32_N))	





	uint8	((ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == UINT8_N) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == UINT8_N))
Size	–	
Description	–	
Variation	(((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) == ([S U]INT[8 16 32] FLOAT)_N) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataType) == ([S U]INT[8 16 32] FLOAT)_N) && ((Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort == USE_DATA_ELEMENT_SPECIFIC_INTERFACES) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort) == USE_DATA_SENDER_RECEIVER) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort) == USE_DATA_SENDER_RECEIVER) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort) == USE_DATA_SENDER_RECEIVER_AS_SERVICE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort) == USE_DATA_SENDER_RECEIVER_AS_SERVICE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_SYNCH_CLIENT_SERVER) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_ASYNC_CLIENT_SERVER) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_ASYNC_CLIENT_SERVER_ERROR) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataUsePort) == USE_DATA_SYNCH_CLIENT_SERVER))) Data = (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)))	
Available via	Rte_Dcm_Type.h	

E.3.1.6 Dcm_DataElementType_{Data}PrimitiveType

Name	Dcm_DataElement_{Data}_PrimitiveType	
Kind	Type	
Derived from	Basetype	Variation
	boolean	((ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == BOOLEAN) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == BOOLEAN))
	float32	((ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == FLOAT) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == FLOAT))
	sint16	((ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == SINT16) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == SINT16))
	sint32	((ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == SINT32) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == SINT32))
	sint8	((ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == SINT8) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == SINT8))





	uint16	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == UINT16) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == UINT16))
	uint32	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == UINT32) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == UINT32))
	uint8	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == UINT8) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == UINT8))
Description	–	
Variation	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} == (BOOLEAN [S U]INT[8 16 32] FLOAT)) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataType)} == (BOOLEAN [S U]INT[8 16 32] FLOAT))) && ({Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort == USE_DATA_ELEMENT_SPECIFIC_INTERFACES}) && (({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == USE_DATA_SENDER_RECEIVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == USE_DATA_SENDER_RECEIVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNC_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNC_CLIENT_SERVER_ERROR) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER))) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})	
Available via	Rte_Dcm_Type.h	

E.3.1.7 Dcm_DataArrayTypeUint8_{Data}Type

Name	Dcm_DataArrayTypeUint8_{Data}Type		
Kind	Array	Element type	uint8
Size	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataByteSize)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.DcmDspPidDataByteSize)})) Elements		
Description	–		
Variation	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES) && (({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} == UINT8_N) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} == UINT8_DYN) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataType)} == UINT8_N)) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})		
Available via	Rte_Dcm_Type.h		

E.3.1.8 {DID}_Struct_DataType

This data type has a different modeling as other data types. The {DID}_Struct_DataType datatype is modeled as prose text only. At the time this specification was created there are no means to visualize this datatype with existing

AUTOSAR tooling as table as all the other data types. Still AUTOSAR allows modeling such data types. Simply that they cannot be shown here as table.

Name	{DID}_Struct_DataType
Kind	Structure
Description	The elements of this structure data type is a composition of all DcmDspDataElement of the DcmDspDid. Example: A DID with the 3 data elements uint32 data1, sint8 data2 and sint16 data 3, has a structure definition of struct { uint32 data1, sint8 data2, sint16 data 3}.
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)) == ((USE_ATOMIC_SENDER_RECEIVER_INTERFACE USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE) USE_ATOMIC_NV_DATA_INTERFACE)) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

E.3.1.9 Dcm_RangeArray_{Range}Type

Name	Dcm_RangeArray_{Range}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRangeMaxDataLength)} Elements		
Description	-		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.DcmDspDidRangeUsePort)} == TRUE Range = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

E.3.1.10 Dcm_ScalingInfoArray_{Data}Type

Name	Dcm_ScalingInfoArray_{Data}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDataInfo.DcmDspDataScalingInfoSize)} Elements		
Description	-		
Variation	(((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_ASYNC_CLIENT_SERVER) ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_SYNC_CLIENT_SERVER) ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_ASYNC_CLIENT_SERVER) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData->DcmDspDataInfoRef.DcmDspDataScalingInfoSize)) != NULL)) != NULL Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

E.3.1.11 Dcm_RequestDataOut_{Routine}_{Signal}PrimitivType

Name	Dcm_RequestDataOut_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT8
Description	-	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32] FLOAT) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.12 Dcm_RequestDataIn_{Routine}_{Signal}PrimitiveType

Name	Dcm_RequestDataIn_{Routine}_{Signal}PrimitiveType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == FLOAT





	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT8
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32]FLOAT) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)})	
Available via	Rte_Dcm_Type.h	

E.3.1.13 Dcm_RequestDataOut_{Routine}_{Signal}Type

Name	Dcm_RequestDataOut_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_RequestDataOut_{Routine}_{Signal}PrimitiveType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32]FLOAT)
	Dcm_RequestDataOut_{Routine}_{Signal}ArrayType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32]FLOAT)_N
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.14 Dcm_RequestDataIn_{Routine}_{Signal}Type

Name	Dcm_RequestDataIn_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_RequestDataIn_{Routine}_{Signal}PrimitiveType	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32 FLOAT])}
	Dcm_RequestDataIn_{Routine}_{Signal}ArrayType	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32 FLOAT]_N)}
Description	-	
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE} Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.15 Dcm_RequestDataOut_{Routine}_{Signal}ArrayType

Name	Dcm_RequestDataOut_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == FLOAT_N}
	sint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT16_N}
	sint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT32_N}
	sint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT8_N}
	uint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT16_N}
	uint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT32_N}
	uint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT8_N}
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineParameterSize)} Elements	
Description	-	





Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

E.3.1.16 Dcm_RequestDataIn_{Routine}_{Signal}ArrayType

Name	Dcm_RequestDataIn_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == FLOAT_N
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT16_N
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT32_N
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT8_N
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT16_N
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT32_N
uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT8_N	
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineParameterSize)} Elements	
Description	-	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.17 Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type

Name	Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineParameterSize) Elements		
Description	–		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

E.3.1.18 Dcm_RequestFlexibleInArrayData_{Routine}_{Signal}Type

Name	Dcm_RequestFlexibleInArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineParameterSize) Elements		
Description	–		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

E.3.1.19 Dcm_StartDataIn_{Routine}_{Signal}PrimitivType

Name	Dcm_StartDataIn_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT32





	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT8
Description	-	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32][FLOAT]) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.20 Dcm_StartDataIn_{Routine}_{Signal}Type

Name	Dcm_StartDataIn_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_StartDataIn_{Routine}_{Signal}PrimitivType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32][FLOAT])
	Dcm_StartDataIn_{Routine}_{Signal}ArrayType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32][FLOAT])_N
Description	-	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.21 Dcm_StartDataIn_{Routine}_{Signal}ArrayType

Name	Dcm_StartDataIn_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation





	float32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == FLOAT_N}
	sint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT16_N}
	sint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT32_N}
	sint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT8_N}
	uint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT16_N}
	uint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT32_N}
	uint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT8_N}
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineParameterSize)} Elements	
Description	-	
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N} && {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE} Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.22 Dcm_StartDataOut_{Routine}_{Signal}PrimitivType

Name	Dcm_StartDataOut_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT16





	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT8
Description	-	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32][FLOAT]) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.23 Dcm_StartDataOut_{Routine}_{Signal}Type

Name	Dcm_StartDataOut_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_StartDataOut_{Routine}_{Signal}PrimitivType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32][FLOAT])
	Dcm_StartDataOut_{Routine}_{Signal}ArrayType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32][FLOAT]_N)
Description	-	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.24 Dcm_StartDataOut_{Routine}_{Signal}ArrayType

Name	Dcm_StartDataOut_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == FLOAT_N
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT16_N





	sint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT32_N)
	sint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT8_N)
	uint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT16_N)
	uint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT32_N)
	uint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT8_N)
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineParameterSize)} Elements	
Description	-	
Variation	({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N) && ({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.25 Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type

Name	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineParameterSize)} Elements		
Description	-		
Variation	({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

E.3.1.26 Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type

Name	Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineParameterSize)} Elements		
Description	-		





Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

E.3.1.27 Dcm_StopDataIn_{Routine}_{Signal}PrimitivType

Name	Dcm_StopDataIn_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT8
Description	-	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.28 Dcm_StopDataIn_{Routine}_{Signal}Type

Name	Dcm_StopDataIn_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation





	Dcm_StopDataIn_{Routine}_{Signal}PrimitivType	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)}
	Dcm_StopDataIn_{Routine}_{Signal}ArrayType	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N}
Description	–	
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.29 Dcm_StopDataIn_{Routine}_{Signal}ArrayType

Name	Dcm_StopDataIn_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == FLOAT_N}
	sint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT16_N}
	sint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT32_N}
	sint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT8_N}
	uint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT16_N}
	uint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT32_N}
	uint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT8_N}
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineParameterSize)} Elements	
Description	–	
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N && {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.30 Dcm_StopDataOut_{Routine}_{Signal}PrimitivType

Name	Dcm_StopDataOut_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT8
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32] FLOAT) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.31 Dcm_StopDataOut_{Routine}_{Signal}Type

Name	Dcm_StopDataOut_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_StopDataOut_{Routine}_{Signal}PrimitivType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)
	Dcm_StopDataOut_{Routine}_{Signal}ArrayType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N
Description	–	





Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

E.3.1.32 Dcm_StopDataOut_{Routine}_{Signal}ArrayType

Name	Dcm_StopDataOut_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == FLOAT_N
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT16_N
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT32_N
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT8_N
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT16_N
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT32_N
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT8_N
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineParameterSize)} Elements	
Description	-	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

E.3.1.33 Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type

Name	Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8





Size	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineParameterSize) Elements
Description	–
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)) == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

E.3.1.34 Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type

Name	Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineParameterSize) Elements		
Description	–		
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)) == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

E.3.1.35 Dcm_KeyArray_{SecurityLevel}Type

Name	Dcm_KeyArray_{SecurityLevel}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityKeySize)} Elements		
Description	–		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityUsePort)} == USE_ASYNC_CLIENT_SERVER SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

E.3.1.36 Dcm_SeedArray_{SecurityLevel}Type

Name	Dcm_SeedArray_{SecurityLevel}Type		
Kind	Array	Element type	uint8





Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecuritySeed Size)} Elements
Description	–
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityUse Port)} == USE_ASYNC_CLIENT_SERVER SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurity Row.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

E.3.1.37 Dcm_SecurityAccessDataRecordArray_{SecurityLevel}Type

Name	Dcm_SecurityAccessDataRecordArray_{SecurityLevel}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDsp SecurityADRSIZE)} Elements		
Description	–		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityUse Port)} == USE_ASYNC_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDsp Security/DcmDspSecurityRow.DcmDspSecurityADRSIZE)} != NULL) SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurity Row.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

E.3.1.38 Dcm_RequestDataArrayType

Name	Dcm_RequestDataArrayType		
Kind	Array	Element type	uint8
Size	(MAX({ecuc(Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocol RxBufferID->DcmDslBuffer.DcmDslBufferSize)}) - 1) Elements		
Description	–		
Variation	–		
Available via	Rte_Dcm_Type.h		

E.3.1.39 Dcm_ControlMask_{DID}Type

Name	Dcm_ControlMask_{DID}Type		
Kind	Type		
Derived from	Basetype	Variation	





	Dcm_ControlMask_{Data}Array Type	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} >= 0x05) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)))
	Dcm_ControlMask_{Data}_PrimitiveType	{ecuc(Dcm/DcmConfigSet/ DcmDsp/DcmDspDid/ DcmDspDidInfoRef->DcmDspDidInfo/ DcmDspDidControl/ DcmDspDidControlMaskSize)} <= 0x04)
Description	-	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlEnableMask)} == NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})	
Available via	Rte_Dcm_Type.h	

E.3.1.40 Dcm_inputOutputControlParameterType

Name	Dcm_InputOutputControlParameterType		
Kind	Type		
Derived from	uint8		
Range	DCM_RETURN_CONTROL_TO_ECU	0x00	returnControlToECU
	DCM_RESET_TO_DEFAULT	0x01	resetToDefault
	DCM_FREEZE_CURRENT_STATE	0x02	freezeCurrentState
	DCM_SHORT_TERM_ADJUSTMENT	0x03	shortTermAdjustment
	DCM_IDLE	0xff	Idle state, no request in processing (initial value)
Description	-		
Variation	-		
Available via	Rte_Dcm_Type.h		

E.3.1.41 Dcm_IOOperationRequest_{DID}Type

Name	Dcm_IOOperationRequest_{DID}Type	
Kind	Structure	
Elements	inputOutputControlParameter	
	Type	Dcm_InputOutputControlParameterType
	Comment	-
	controlEnableMask	
Type	Dcm_Cemr_{DID}Type	





	Comment	-
Description	-	
Variation	<pre> ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE)) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE))&& ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl)} != NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}) </pre>	
Available via	Rte_Dcm_Type.h	

E.3.1.42 Dcm_IOOperationResponseType

Name	Dcm_IOOperationResponseType		
Kind	Type		
Derived from	uint8		
Range	DCM_POSITIVE_RESPONSE	0x00	positive response (similar to E_OK)
	DCM_GENERAL_REJECT	0x10	NRC generalReject
	DCM_BUSY_REPEAT_REQUEST	0x21	NRC busyRepeatRequest
	DCM_CONDITIONS_NOT_CORRECT	0x22	NRC conditionsNotCorrect
	DCM_FAILURE_PREVENTS_EXECUTION	0x26	NRC FailurePreventsExecutionOf RequestedAction
	DCM_REQUEST_OUT_OF_RANGE	0x31	NRC requestOutOfRange
	DCM_RESPONSE_PENDING	0x78	ResponsePending (similar to E_PENDING)
Description	-		
Variation	-		
Available via	Rte_Dcm_Type.h		

E.3.1.43 Dcm_ResponseDataArrayType

Name	Dcm_ResponseDataArrayType		
Kind	Array	Element type	uint8
Size	(MAX({ecuc(Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocol TxBufferRef->DcmDslBuffer.DcmDslBufferSize)}) -1) Elements		
Description	-		
Variation	<pre> ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestFileTransfer/DcmRequestFileTransferUsePort)} == TRUE) </pre>		
Available via	Rte_Dcm_Type.h		

E.3.1.44 Dcm_ControlMask_{Data}ArrayType

Name	Dcm_ControlMask_{Data}ArrayType		
Kind	Array	Element type	uint8
Size	({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)}} Elements		
Description	The current DcmDspData is referenced by the DcmDspDID.		
Variation	(({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)}} >= 0x05) && ({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)}} == (USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER_ERROR))) Data = ({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME}})		
Available via	Rte_Dcm_Type.h		

E.3.1.45 Dcm_Cemr_{DID}Type

Name	Dcm_Cemr_{DID}Type		
Kind	Bitfield		
Derived from	Dcm_ControlMask_{DID}Type		
Elements	Variable bit		
	Kind	bit	
	Name	({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidControl.DcmDspDidControlEnableMask.SHORT-NAME}})	
	Mask	2 ² ({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)}} * 8 - 1) - {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlEnableMask/DcmDspDidControlMaskBitPosition}})	
	Description	Bitmask of data element in control enable mask record.	
Description	-		
Variation	({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidControl/DcmDspDidControlEnableMask)}} != NULL) DID = ({{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME}})		
Available via	Rte_Dcm_Type.h		

E.3.2 Sender-Receiver-Interfaces

Using the concepts of the SW-C template, the interface is defined as follows if Sender-Receiver interface is used ([DcmDspDataUsePort](#) set to [USE_DATA_SENDER_RECEIVER || USE_DATA_SENDER_RECEIVER_AS_SERVICE](#)):

E.3.2.1 DataServices_{DID}

Name	DataServices_{DID}	
Comment	–	
IsService	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE)	
Variation	(((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE)) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE))) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}	
Data Elements	data	
	Type	{DID}_Struct_DataType
	Variation	DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}

E.3.2.2 DataServices_{Data}

Name	DataServices_{Data}	
Comment	–	
IsService	(((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE))	
Variation	(((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES)&&(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE))) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE)) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)} (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME))	
Data Elements	data	
	Type	Dcm_DataElement_{Data}Type
	Variation	Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)} (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME))

E.3.2.3 IOControlRequest_{DID}

Name	IOControlRequest_{DID}
Comment	Attention: controlState is only valid in case of IOOperationRequest is set to shortTerm Adjustment. The DCM provides a byte stream which could be transformed via transformer into an complex type.
IsService	true





Variation	<pre>((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE) {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/ DcmDspDidControl)} != NULL)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL))) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>	
Data Elements	underControl	
	Type	Dcm_Cemr_{DID}Type
	Variation	–
	IOOperationRequest	
	Type	Dcm_IOOperationRequest_{DID}Type
	Variation	DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})
	controlState	
	Type	{DID}_Struct_DataType
Variation	<pre>{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == True) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>	

E.3.2.4 IOControlResponse_{DID}

Name	IOControlResponse_{DID}	
Comment	–	
IsService	true	
Variation	<pre>{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE)) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl)} != NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>	
Data Elements	IOOperationResponse	
	Type	Dcm_IOOperationResponseType
	Variation	–

E.3.3 NvDataInterface

E.3.3.1 DataServices_{DID}

Name	DataServices_{DID}	
Comment	–	
IsService	false	
Variation	<pre>{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_ATOMIC_NV_DATA_INTERFACE) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}</pre>	
Data Elements	data	





	Type	{DID}_Struct_DataType
	Variation	DID = {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}}

E.3.4 Client-Server-Interfaces

E.3.4.1 DataServices_{Data}

E.3.4.2 RoutineServices_{RoutineName}

The following interface defines operations needed for the UDS Service RoutineControl (0x31).

Name	RoutineServices_{RoutineName}		
Comment	–		
IsService	true		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.DcmDspRoutineUsePort)} == TRUE RoutineName = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DCM_E_PENDING	Request is not yet finished. Further call(s) required to finish.
	12	DCM_E_FORCE_RCRRP	application request the transmission of a response Response Pending (NRC 0x78)

Operation	RequestResults	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_RequestDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_RequestDataOut_{Routine}_{Signal}Type
	Direction	OUT





	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRFP	

Operation	RequestResults	
Comment	–	
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH}	
Parameters	DataIn_{Signal}	
	Type	Dcm_RequestDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_RequestDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
DataOut_{Signal}		
Type	Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type	
Direction	OUT	





Possible Errors	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength	
	Type	uint16
	Direction	OUT
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–

Operation	RequestResults	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_RequestDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataIn_{Signal}	
	Type	Dcm_RequestFlexibleInArrayData_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–





	DataOut_{Signal}
Type	Dcm_RequestDataOut_{Routine}_{Signal}Type
Direction	OUT
Comment	–
Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	ErrorCode
Type	Dcm_NegativeResponseCodeType
Direction	OUT
Comment	–
Variation	–
Possible Errors	–

Operation	RequestResults	
Comment	–	
Variation	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH))	
Parameters	DataIn_{Signal}	
	Type	Dcm_RequestDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
		DataIn_{Signal}
	Type	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	–
		OpStatus
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
Type	Dcm_RequestDataOut_{Routine}_{Signal}Type	
Direction	OUT	
Comment	–	





	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}	
	Type	Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength	
	Type	uint16
	Direction	OUT
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	RequestResultsConfirmation
Comment	This operation indicates the transmission of a response to a RequestResultsRoutine request
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsConfirmationEnabled))==TRUE)
Parameters	ConfirmationStatus
	Type Dcm_ConfirmationStatusType
	Direction IN
	Comment Confirmation status of a RequestResultsRoutine request
	Variation –
Possible Errors	E_OK E_NOT_OK

Operation	Start
Comment	–





Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH)		
Parameters	DataIn_{Signal}		
	Type	Dcm_StartDataIn_{Routine}_{Signal}Type	
	Direction	IN	
	Comment	–	
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
	OpStatus		
	Type	Dcm_OpStatusType	
	Direction	IN	
	Comment	–	
	Variation	–	
	DataOut_{Signal}		
	Type	Dcm_StartDataOut_{Routine}_{Signal}Type	
	Direction	OUT	
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
ErrorCode			
Type	Dcm_NegativeResponseCodeType		
Direction	OUT		
Comment	–		
Variation	–		
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP		

Operation	Start	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_StartDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–





	Variation	Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)} Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StartDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}	
	Type	Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	–
	currentDataLength	
	Type	uint16
	Direction	OUT
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	Start
Comment	–
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH)
Parameters	DataIn_{Signal}
	Type Dcm_StartDataIn_{Routine}_{Signal}Type
	Direction IN
	Comment –





	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataIn_{Signal}	
	Type	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	–
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StartDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	Start	
Comment	–	
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_StartDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–





Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
DataIn_{Signal}	
Type	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type
Direction	IN
Comment	–
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
OpStatus	
Type	Dcm_OpStatusType
Direction	IN
Comment	–
Variation	–
DataOut_{Signal}	
Type	Dcm_StartDataOut_{Routine}_{Signal}Type
Direction	OUT
Comment	–
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
DataOut_{Signal}	
Type	Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type
Direction	OUT
Comment	–
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
currentDataLength	
Type	uint16
Direction	INOUT
Comment	–
Variation	–





	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	StartConfirmation	
Comment	This operation indicates the transmission of a response to a StartRoutine request	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineConfirmationEnabled)}==TRUE	
Parameters	ConfirmationStatus	
	Type	Dcm_ConfirmationStatusType
	Direction	IN
	Comment	Confirmation status of a StartRoutine request
Possible Errors	E_OK E_NOT_OK	

Operation	Stop		
Comment	–		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH)		
Parameters	DataIn_{Signal}		
	Type	Dcm_StopDataIn_{Routine}_{Signal}Type	
	Direction	IN	
	Comment	–	
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
	OpStatus		
	Type	Dcm_OpStatusType	
	Direction	IN	
	Comment	–	
	Variation	–	
	DataOut_{Signal}		
	Type	Dcm_StopDataOut_{Routine}_{Signal}Type	
	Direction	OUT	
Comment	–		





	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	Stop	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_StopDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StopDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}	
Type	Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type	
Direction	OUT	
Comment	–	
Variation	–	
currentDataLength		





	Type	uint16
	Direction	OUT
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRFP	

Operation	Stop	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_StopDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataIn_{Signal}	
	Type	Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	–
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StopDataOut_{Routine}_{Signal}Type
	Direction	OUT
Comment	–	
Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
currentDataLength		





	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRFP	

Operation	Stop	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_StopDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataIn_{Signal}	
	Type	Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
Variation	–	
DataOut_{Signal}		
Type	Dcm_StopDataOut_{Routine}_{Signal}Type	
Direction	OUT	





	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}	
	Type	Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength	
	Type	uint16
	Direction	INOUT
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	StopConfirmation
Comment	This operation indicates the transmission of a response to a StopRoutine request
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineConfirmationEnabled))==TRUE)
Parameters	ConfirmationStatus
	Type Dcm_ConfirmationStatusType
	Direction IN
	Comment Confirmation status of a StopRoutine request
	Variation –
Possible Errors	E_OK E_NOT_OK

E.3.4.3 ServiceRequestNotification

The following interface indicates to the Application that a service is about to be executed and allows the Application to reject the execution of the service request.

Name	ServiceRequestNotification		
Comment	–		
IsService	true		
Variation	({{ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestManufacturerNotification)}} != NULL) ({{ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestSupplierNotification)}} != NULL)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	8	E_REQUEST_NOT_ACCEPTED	no response will be sent

Operation	Confirmation	
Comment	–	
Variation	–	
Parameters	SID	
	Type	uint8
	Direction	IN
	Comment	Value of service identifier
	Variation	–
	ReqType	
	Type	uint8
	Direction	IN
	Comment	Addressing type of the request(0=physical request, 1=functional request)
	Variation	–
	ConnectionId	
	Type	uint16
	Direction	IN
	Comment	Unique connection identifier
	Variation	–
	ConfirmationStatus	
	Type	Dcm_ConfirmationStatusType
	Direction	IN
	Comment	Confirmation of a successful transmission or a transmission error of a diagnostic service.
	Variation	–
	ProtocolType	
	Type	Dcm_ProtocolType
	Direction	IN
	Comment	–
	Variation	–
	TesterSourceAddress	
	Type	uint16
Direction	IN	
Comment	–	





	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	Indication		
Comment	–		
Variation	–		
Parameters	SID		
	Type	uint8	
	Direction	IN	
	Comment	Value of service identifier	
	Variation	–	
	RequestData		
	Type	Dcm_RequestDataArrayType	
	Direction	IN	
	Comment	This parameter contains the complete request data (diagnostic buffer), except the service ID	
	Variation	–	
	DataSize		
	Type	uint16	
	Direction	IN	
	Comment	This parameter defines how many bytes in the RequestData parameter are valid	
	Variation	–	
	ReqType		
	Type	uint8	
	Direction	IN	
	Comment	Addressing type of the request(0=physical request, 1=functional request)	
	Variation	–	
	ConnectionId		
	Type	uint16	
	Direction	IN	
	Comment	Unique connection identifier	
	Variation	–	
	ErrorCode		
	Type	Dcm_NegativeResponseCodeType	
	Direction	OUT	
	Comment	–	
	Variation	–	
	ProtocolType		
	Type	Dcm_ProtocolType	
Direction	IN		
Comment	–		
Variation	–		
TesterSourceAddress			
Type	uint16		
Direction	IN		
Comment	–		





	Variation	-
Possible Errors	E_OK E_NOT_OK E_REQUEST_NOT_ACCEPTED	

E.3.5 Ports

E.3.5.1 DataServices_{DID} RPortPrototype

Name	DataServices_{DID}		
Kind	RequiredPort	Interface	DataServices_{DID} , DataServices_{DID}
Description	-		
Variation	<pre> ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)) == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)) == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)) == (USE_ATOMIC_NV_DATA_INTERFACE)) && (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidWrite) == NULL) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidRead) != NULL) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl) !=NULL)) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)} </pre>		

E.3.5.2 DataServices_{DID} PPortPrototype

Name	DataServices_{DID}		
Kind	ProvidedPort	Interface	DataServices_{DID} , DataServices_{DID}
Description	-		
Variation	<pre> ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)) == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)) == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)) == (USE_ATOMIC_NV_DATA_INTERFACE)) && (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidWrite) != NULL) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidRead) == NULL) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl) !=NULL)) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)} </pre>		

E.3.5.3 DataServices_{DID} PRPortPrototype

Name	DataServices_{DID}		
Kind	Provided RequiredPort	Interface	DataServices_{DID} , DataServices_{DID}
Description	-		





Variation	<pre>(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_NV_DATA_INTERFACE)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidWrite)} != NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidRead)} != NULL)) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}</pre>
------------------	---

E.3.5.4 DataServices_{Data} RPortPrototype

Name	DataServices_{Data}		
Kind	RequiredPort	Interface	DataServices_{Data}, DataServices_{Data}
Description	–		
Variation	<pre>({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == (USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE)) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)} {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)}</pre>		

E.3.5.5 DataServices_{Data} RPortPrototype

Name	DataServices_{Data}		
Kind	RequiredPort	Interface	DataServices_{Data}, DataServices_{Data}
Description	–		
Variation	<pre>((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidWrite)} == NULL) && (({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidRead)} != NULL))) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}</pre>		

E.3.5.6 DataServices_{Data} PPortPrototype

Name	DataServices_{Data}		
Kind	ProvidedPort	Interface	DataServices_{Data}, DataServices_{Data}
Description	–		
Variation	<pre>((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidWrite)} != NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidRead)} == NULL)) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}</pre>		

E.3.5.7 DataServices_{Data} PRPortPrototype

Name	DataServices_{Data}		
Kind	Provided RequiredPort	Interface	DataServices_{Data}
Description	-		
Variation	<pre>((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE))) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidWrite)} != NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidRead)} != NULL) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})</pre>		

E.3.5.8 IOControlRequest_{DID}

Name	IOControlRequest_{DID}		
Kind	Provided RequiredPort	Interface	IOControlRequest_{DID}
Description	-		
Variation	<pre>(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE))) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl)} != NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>		

E.3.5.9 IOControlResponse_{DID}

Name	IOControlResponse_{DID}		
Kind	RequiredPort	Interface	IOControlResponse_{DID}
Description	-		
Variation	<pre>(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE))) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl)} != NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>		

E.3.5.10 ServiceRequestManufacturerNotification_{Name}

Name	ServiceRequestManufacturerNotification_{Name}		
Kind	RequiredPort	Interface	ServiceRequestNotification
Description	-		





Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestManufacturerNotification)) != NULL) Name = {ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestManufacturerNotification.SHORT-NAME)}
------------------	--

E.3.5.11 ServiceRequestSupplierNotification_{Name}

Name	ServiceRequestSupplierNotification_{Name}		
Kind	RequiredPort	Interface	ServiceRequestNotification
Description	–		
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestSupplierNotification)) != NULL) Name = {ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestSupplierNotification.SHORT-NAME)}		

E.3.5.12 RoutineServices_{RoutineName}

Name	RoutineServices_{RoutineName}		
Kind	RequiredPort	Interface	RoutineServices_{RoutineName}
Description	–		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.DcmDspRoutineUsePort)} == TRUE RoutineName = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		

E.4 Dem

E.4.1 Implementation Data Types

E.4.1.1 Dem_DebouncingStateType

Name	Dem_DebouncingStateType			
Kind	Bitfield			
Derived from	uint8			
Elements	Kind	Name	Mask	Description
	bit	DEM_TEMPORARILY_DEFECTIVE	0x01	Bit 0: Temporarily Defective (corresponds to 0 < FDC < 127)
	bit	DEM_FINALLY_DEFECTIVE	0x02	Bit 1: finally Defective (corresponds to FDC = 127)
	bit	DEM_TEMPORARILY_HEALED	0x04	Bit 2: temporarily healed (corresponds to -128 < FDC < 0)
	bit	DEM_TEST_COMPLETE	0x08	Bit 3: Test complete (corresponds to FDC = -128 or FDC = 127)





	bit	DEM_DTR_UPDATE	0x10	Bit 4: DTR Update (= Test complete && Debouncing complete && enable conditions / storage conditions fulfilled)
Description	-			
Variation	-			
Available via	Rte_Dem_Type.h			

E.4.1.2 Dem_DTCTOriginType

Name	Dem_DTCTOriginType			
Kind	Type			
Derived from	uint16			
Range	DEM_DTC_ORIGIN_PRIMARY_MEMORY	0x0001	Event information located in the primary memory	
	DEM_DTC_ORIGIN_PERMANENT_MEMORY	0x0003	The Event information is located in the permanent memory	
	DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY	0x0004	OBD event information located in the primary memory.	
	DEM_DTC_ORIGIN_USERDEFINED_MEMORY_<Name>	0x01XX	Event information located in the user defined memory, where XX is the configured DemUserDefinedMemoryIdentifier in hexadecimal and <Name> is the Short-Name of the DemUserDefinedMemory.	
Description	This enum is used to define the location of the events. The definition and use of the different memory types is OEM-specific.			
Variation	-			
Available via	Rte_Dem_Type.h			

E.4.1.3 Dem_EventIdType

Name	Dem_EventIdType			
Kind	Type			
Derived from	uint16			
Range	1..65535	-	Internal identifier of a diagnostic event Remark: 0 is not a valid value	
Description	Identification of an event by assigned EventId. The EventId is assigned by the Dem. Example: 1 refers to monitor x, 2 refers to monitor y, etc.			
Variation	-			
Available via	Rte_Dem_Type.h			

E.4.1.4 Dem_EventStatusType

Name	Dem_EventStatusType		
Kind	Type		
Derived from	uint8		
Range	DEM_EVENT_STATUS_PASSED	0x00	Monitor reports qualified test result passed.
	DEM_EVENT_STATUS_FAILED	0x01	Monitor reports qualified test result failed.
	DEM_EVENT_STATUS_PREPASSED	0x02	Monitor reports non-qualified test result pre-passed (debounced Dem-internally).
	DEM_EVENT_STATUS_PREFAILED	0x03	Monitor reports non-qualified test result pre-failed (debounced Dem-internally).
	DEM_EVENT_STATUS_FDC_THRESHOLD_REACHED	0x04	Monitor triggers the storage of ExtendedDataRecords and FreezeFrames ON_FDC_THRESHOLD.
		0x05 - 0xFF	reserved
Description	This type contains all monitor test result values, which can be reported via Dem_SetEventStatus().		
Variation	–		
Available via	Rte_Dem_Type.h		

E.4.1.5 Dem_DebounceResetStatusType

Name	Dem_DebounceResetStatusType		
Kind	Type		
Derived from	uint8		
Range	DEM_DEBOUNCE_STATUS_FREEZE	0x00	Freeze the internal debounce counter/timer.
	DEM_DEBOUNCE_STATUS_RESET	0x01	Reset the internal debounce counter/timer.
			0x02 - 0xFF reserved
Description	This type contains all definitions to control an internal debounce counter/timer via the function Dem_ResetEventDebounceStatus().		
Variation	–		
Available via	Rte_Dem_Type.h		

E.4.1.6 Dem_DTCTFormatType

Name	Dem_DTCTFormatType		
Kind	Type		
Derived from	uint8		





Range	DEM_DTC_FORMAT_OBD	0	selects the 2-byte OBD DTC format (refer to configuration parameter DemObdDTC)
	DEM_DTC_FORMAT_UDS	1	selects the 3-byte UDS DTC format (refer to configuration parameter DemUdsDTC)
	DEM_DTC_FORMAT_J1939	2	selects the merged SPN + FMI to 3-byte J1939 DTC format (refer to DemJ1939DTC)
	DEM_DTC_FORMAT_OBD_3BYTE	3	Selects the 3-byte OBD DTC defined by DemDtcValue3Byte in case that SAE J2012 and UDS DTC format separation is used.
Description	This type is used to select the format of the DTC value.		
Variation	–		
Available via	Rte_Dem_Type.h		

E.4.1.7 Dem_InitMonitorReasonType

Name	Dem_InitMonitorReasonType		
Kind	Type		
Derived from	uint8		
Range	DEM_INIT_MONITOR_CLEAR	0x01	Event was cleared and all internal values and states are reset.
	DEM_INIT_MONITOR_RESTART	0x02	Operation cycle of the event was restarted.
	DEM_INIT_MONITOR_REENABLED	0x03	Enable conditions or DTC settings re-enabled.
	DEM_INIT_MONITOR_STORAGE_REENABLED	0x04	Storage condition reenabled.
Description	(Re-)Initialization reason returned by the callback <Module>_DemInitMonitorFor<EventName>().		
Variation	–		
Available via	Rte_Dem_Type.h		

E.4.1.8 Dem_MaxDataValueType

Name	Dem_MaxDataValueType		
Kind	Array	Element type	uint8
Size	size of largest Extended data class / Freeze frame record Elements		
Description	–		
Variation	–		
Available via	Rte_Dem_Type.h		

E.4.1.9 Dem_MonitorDataType

Name	Dem_MonitorDataType
Kind	Type
Derived from	uint32
Description	This type is used to pass monitoring data to the Dem.
Variation	–
Available via	Rte_Dem_Type.h

E.4.1.10 Dem_MonitorStatusType

Name	Dem_MonitorStatusType			
Kind	Bitfield			
Derived from	uint8			
Elements	Kind	Name	Mask	Description
	bit	DEM_MONITOR_STATUS_TF	0x01	Bit0: TestFailed
	bit	DEM_MONITOR_STATUS_TNCTOC	0x02	Bit1: TestNotCompletedThisOperationCycle
Description	This type contains possible monitor status values.			
Variation	–			
Available via	Rte_Dem_Type.h			

E.4.1.11 Dem_UdsStatusByteType

Name	Dem_UdsStatusByteType			
Kind	Bitfield			
Derived from	uint8			
Lower limit	0x00			
Upper limit	0xFF			
Elements	Kind	Name	Mask	Description
	bit	DEM_UDS_STATUS_TF	0x01	bit 0: TestFailed
	bit	DEM_UDS_STATUS_TFTOC	0x02	bit 1: TestFailedThisOperationCycle
	bit	DEM_UDS_STATUS_PDTC	0x04	bit 2: PendingDTC
	bit	DEM_UDS_STATUS_CDTC	0x08	bit 3: ConfirmedDTC
	bit	DEM_UDS_STATUS_TNCSLC	0x10	bit 4: TestNotCompletedSinceLastClear
	bit	DEM_UDS_STATUS_TFSLC	0x20	bit 5: TestFailedSinceLastClear
	bit	DEM_UDS_STATUS_TNCTOC	0x40	bit 6: TestNotCompletedThisOperationCycle
bit	DEM_UDS_STATUS_WIR	0x80	bit 7: WarningIndicatorRequested	





Description	In this data-type each bit has an individual meaning. The bit is set to 1 when the condition holds. For example, if the 2nd bit (0x02) is set to 1, this means that the test failed this operation cycle. If the bit is set to 0, it has not yet failed this cycle.
Variation	–
Available via	Rte_Dem_Type.h

E.4.2 Client-Server-Interfaces

E.4.2.1 CallbackClearEventAllowed

Name	CallbackClearEventAllowed		
Comment	If configured, it gets the permission to clear a specific event from the SW-C. For each event, there can be one port of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	ClearEventAllowed		
Comment	–		
Variation	–		
Parameters	Allowed		
	Type	boolean	
	Direction	OUT	
	Comment	True - clearance of event is allowed False - clearance of event is not allowed	
	Variation	–	
Possible Errors	E_OK E_NOT_OK		

E.4.2.2 CallbackEventDataChanged

Name	CallbackEventDataChanged		
Comment	If configured it triggers SW-Cs on event related data changes. For each event, there can be one port of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	EventDataChanged		
Comment	–		
Variation	–		
Possible Errors	–		

E.4.2.3 CallbackEventUdsStatusChanged

Name	CallbackEventUdsStatusChanged		
Comment	If configured it triggers SW-Cs on event status byte changes. For each event, there can be several ports of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	CallbackEventUdsStatusChanged		
Comment	–		
Variation	–		
Parameters	EventStatusByteOld		
	Type	Dem_UdsStatusByteType	
	Direction	IN	
	Comment	–	
	Variation	–	
	EventStatusByteNew		
	Type	Dem_UdsStatusByteType	
	Direction	IN	
Comment	–		
Variation	–		
Possible Errors	–		

E.4.2.4 CallbackGetFaultDetectCounter

Name	CallbackGetFaultDetectCounter		
Comment	If configured it get the monitor-internal fault detection counter value of a specific event from the SW-C.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetFaultDetectionCounter		
Comment	–		
Variation	–		
Parameters	FaultDetectionCounter		
	Type	sint8	
	Direction	OUT	
	Comment	Value of FaultDetectionCounter	
	Variation	–	
Possible Errors	E_OK E_NOT_OK		

E.4.2.5 CallbackInitMonitorForEvent

Name	CallbackInitMonitorForEvent		
Comment	If configure it triggers an event-specific initialization of the monitor part of the SW-C). For each event, there can be one port of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	InitMonitorForEvent		
Comment	–		
Variation	–		
Parameters	InitMonitorReason		
	Type	Dem_InitMonitorReasonType	
	Direction	IN	
	Comment	–	
	Variation	–	
Possible Errors	E_OK E_NOT_OK		

E.4.2.6 CallbackMonitorStatusChange

Name	CallbackMonitorStatusChange		
Comment	If configured it triggers SW-Cs on monitor status changes. For each event, there can be several ports of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	MonitorStatusChanged		
Comment	–		
Variation	–		
Possible Errors	–		

E.4.2.7 DiagnosticInfo

Name	DiagnosticInfo		
Comment	Provides the capability to obtain the event information. One port of this interface type is provided per diagnostic event by the Dem Service Component. It has EventId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed





	10	DEM_E_NO_DTC_AVAILABLE	there is no DTC configured in the requested format
	14	DEM_E_NO_FDC_AVAILABLE	there is no fault detection counter available for the requested event
	21	DEM_BUFFER_TOO_SMALL	The provided buffer size is too small
	48	DEM_NO_SUCH_ELEMENT	The requested event data is not currently stored (but the request was valid) OR The requested record number is not supported by the event OR The requested DID is not supported by the freeze frame (GetEventFreezeFrameDataEx)

Operation	GetDTCOfEvent	
Comment	–	
Variation	–	
Parameters	DTCFormat	
	Type	Dem_DTCFormatType
	Direction	IN
	Comment	–
	Variation	–
	DTCOfEvent	
	Type	uint32
	Direction	OUT
Possible Errors	E_OK E_NOT_OK DEM_E_NO_DTC_AVAILABLE	

Operation	GetDebouncingOfEvent	
Comment	–	
Variation	((({ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass)} instanceof {ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceCounterBased)})) ({ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass)} instanceof {ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceTimeBase)}))	
Parameters	DebouncingState	
	Type	Dem_DebouncingStateType
	Direction	OUT
	Comment	Bit 0 Temporarily Defective (corresponds to 0 < FDC < 127) Bit 1 finally Defective (corresponds to FDC = 127) Bit 2 temporarily healed (corresponds to -128 < FDC < 0) Bit 3 Test complete (corresponds to FDC = -128 or FDC = 127) Bit 4 DTR Update (= Test complete && Debouncing complete && enable conditions / storage conditions fulfilled)
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetEventAvailable	
Comment	–	





Variation	-	
Parameters	AvailableStatus	
	Type	boolean*
	Direction	OUT
	Comment	TRUE if the event is available. FALSE if the event is not available.
	Variation	-
Possible Errors	E_OK E_NOT_OK	

Operation	GetEventExtendedDataRecordEx	
Comment	-	
Variation	-	
Parameters	RecordNumber	
	Type	uint8
	Direction	IN
	Comment	-
	Variation	-
	DestBuffer	
	Type	Dem_MaxDataValueType
	Direction	OUT
	Comment	-
	Variation	-
	Bufsize	
	Type	uint16
	Direction	INOUT
	Comment	-
	Variation	-
Possible Errors	E_OK E_NOT_OK DEM_BUFFER_TOO_SMALL DEM_NO_SUCH_ELEMENT	

Operation	GetEventFreezeFrameDataEx	
Comment	-	
Variation	-	
Parameters	RecordNumber	
	Type	uint8
	Direction	IN
	Comment	-
	Variation	-
	DataId	
	Type	uint16
	Direction	IN
	Comment	-
	Variation	-
	DestBuffer	
	Type	Dem_MaxDataValueType
	Direction	OUT





	Comment	–
	Variation	–
	BufSize	
	Type	uint16
	Direction	INOUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_BUFFER_TOO_SMALL DEM_NO_SUCH_ELEMENT	

Operation	GetEventUdsStatus	
Comment	Gets the current UDS status byte assigned to the DTC for the event	
Variation	–	
Parameters	UDSStatusByte	
	Type	Dem_UdsStatusByteType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetFaultDetectionCounter	
Comment	–	
Variation	–	
Parameters	FaultDetectionCounter	
	Type	sint8
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_E_NO_FDC_AVAILABLE	

Operation	GetMonitorStatus	
Comment	–	
Variation	–	
Parameters	MonitorStatus	
	Type	Dem_MonitorStatusType
	Direction	OUT
	Comment	Monitor status byte of the requested event. If the return value of the function call is E_NOT_OK, this parameter does not contain valid data.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

E.4.2.8 DiagnosticMonitor

Name	DiagnosticMonitor		
Comment	Provide the capability to modify the event information. One port of this interface type is provided per application-related diagnostic event by the Dem Service Component. It has Event Id as a port-defined argument.		
IsService	true		
Variation	(((Dem/DemGeneral/DemStorageCondition.DemStorageConditionReplacementEvent Ref))!={Dem/DemConfigSet/DemEventParameter}) && ((Dem/DemConfigSet/DemEventParameter/DemEventReportingType) == STANDARD_REPORTING))		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	ClearPrestoredFreezeFrame		
Comment	–		
Variation	{ecuc(Dem/DemGeneral/DemMaxNumberPrestoredFF)} > 0		
Possible Errors	E_OK E_NOT_OK		

Operation	PrestoreFreezeFrame		
Comment	–		
Variation	{ecuc(Dem/DemGeneral/DemMaxNumberPrestoredFF)} > 0		
Possible Errors	E_OK E_NOT_OK		

Operation	ResetEventDebounceStatus		
Comment	–		
Variation	((ecuc(Dem/DemGeneral/DemDebounceCounterBasedSupport)) == true) ((ecuc(Dem/DemGeneral/DemDebounceTimeBasedSupport)) == true)		
Parameters	DebounceResetStatus		
	Type	Dem_DebounceResetStatusType	
	Direction	IN	
	Comment	–	
	Variation	–	
Possible Errors	E_OK E_NOT_OK		

Operation	ResetEventStatus		
Comment	–		
Variation	–		
Possible Errors	E_OK E_NOT_OK		

Operation	SetEventDisabled		
Comment	–		
Variation	{ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT		
Possible Errors	E_OK E_NOT_OK		

Operation	SetEventStatus	
Comment	–	
Variation	–	
Parameters	EventStatus	
	Type	Dem_EventStatusType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

E.4.2.9 DiagnosticMonitor_MonitorData

Name	DiagnosticMonitor_MonitorData		
Comment	Provide means to report diagnostic monitor status with monitor data.		
IsService	true		
Variation	({Dem/DemConfigSet/DemEventParameter/DemEventReportingType} == STANDARD_REPORTING_WITH_MONITOR_DATA)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetEventStatusWithMonitorData	
Comment	–	
Variation	–	
Parameters	EventStatus	
	Type	Dem_EventStatusType
	Direction	IN
	Comment	–
	Variation	–
	monitorData0	
	Type	Dem_MonitorDataType
	Direction	IN
	Comment	–
	Variation	–
	monitorData1	
	Type	Dem_MonitorDataType
	Direction	IN
	Comment	–
Variation	–	
Possible Errors	E_OK E_NOT_OK	

E.4.2.10 EventStatus

Name	EventStatus		
Comment	Provides the capability modify the event status. One port of this interface type is provided per application-related diagnostic event by the Dem Service Component. It has EventId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetWIRStatus		
Comment	–		
Variation	–		
Parameters	WIRStatus		
	Type	boolean	
	Direction	IN	
	Comment	–	
	Variation	–	
Possible Errors	E_OK E_NOT_OK		

E.4.3 Ports

E.4.3.1 Event

Name	Event_{Name}		
Kind	ProvidedPort	Interface	DiagnosticMonitor
Description	–		
Port Defined Argument Value(s)	Type	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

E.4.3.2 DiagnosticMonitor_MonitorData

Name	DiagnosticMonitor_MonitorData_{Name}		
Kind	ProvidedPort	Interface	DiagnosticMonitor_MonitorData
Description	–		
Port Defined Argument Value(s)	Type	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

E.4.3.3 EventStatus

Name	EventStatus_{Name}		
Kind	ProvidedPort	Interface	EventStatus
Description	-		
Port Defined Argument Value(s)	Type	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

E.4.3.4 EventInfo

Name	EventInfo_{Name}		
Kind	ProvidedPort	Interface	DiagnosticInfo
Description	-		
Port Defined Argument Value(s)	Type	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

E.4.3.5 CBClrEvt

Name	CBClrEvt_{Name}		
Kind	RequiredPort	Interface	CallbackClearEventAllowed
Description	-		
Variation	(({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed)} != NULL) && ({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed/DemCallback ClearEventAllowedFnc)} == NULL)) Name = {ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackClearEvent Allowed.SHORT-NAME)}		

E.4.3.6 CBDataEvt

Name	CBDataEvt_{Name}		
Kind	RequiredPort	Interface	CallbackEventDataChanged
Description	-		
Variation	(({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackEventDataChanged)} != NULL) && ({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackEventDataChanged/DemCallback EventDataChangedFnc)} == NULL)) Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

E.4.3.7 CBFaultDetectCtr

Name	CBFaultDetectCtr_{Name}		
Kind	RequiredPort	Interface	CallbackGetFaultDetectCounter
Description	–		
Variation	({{ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceMonitorInternal/DemCallbackGetFDC/DemCallbackGetFDCFnc)} == NULL) Name = {ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceMonitorInternal/DemCallbackGetFDC.SHORT-NAME)}		

E.4.3.8 CBInitEvt

Name	CBInitEvt_{Name}		
Kind	RequiredPort	Interface	CallbackInitMonitorForEvent
Description	–		
Variation	({{ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackInitMForE)} != NULL} && ({{ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackInitMForE/DemCallbackInitMForEFnc)} == NULL)) Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

E.4.3.9 CBEventUdsStatusChanged

Name	CBEventUdsStatusChanged_{EventName}_{CallbackName}		
Kind	RequiredPort	Interface	CallbackEventUdsStatusChanged
Description	–		
Variation	EventName = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)} CallbackName = {ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackEventUdsStatusChanged.SHORT-NAME)}		

E.4.3.10 CBMonitorStatusChanged

Name	CBMonitorStatusChanged_{EventName}_{CallbackName}		
Kind	RequiredPort	Interface	CallbackMonitorStatusChange
Description	–		
Variation	({{ecuc(Dem/DemConfigSet/DemEventParameter.DemEventKind)} == DEM_EVENT_KIND_SWC) EventName = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)} CallbackName = {ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackMonitorStatusChanged.SHORT-NAME)}		

E.5 FiM

E.5.1 Implementation Data Types

E.5.1.1 FiM_FunctionIdType

Name	FiM_FunctionIdType		
Kind	Type		
Derived from	Basetype	Variation	
	uint16	platform depended	
	uint8	platform depended	
Range	0..255, 0..65535	–	Identifier of functionality Configurable, size depends on System complexity. Remark: Not all numbers are valid. The FiM data generation tool shall only assign valid values.
Description	Type for the FunctionID		
Variation	–		
Available via	Rte_FiM_Type.h		

E.5.2 Client-Server-Interfaces

E.5.2.1 FiM_FunctionInhibition

Name	FunctionInhibition		
Comment	The SW Components can use this service to query for the permission to execute a certain functionality represented by a FID.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetFunctionPermission		
Comment	Get the permission state of the respective FID.		
Variation	–		
Parameters	Permission		
	Type	boolean	
	Direction	OUT	
	Comment	The permission of the requested FID. TRUE: FID has permission to run FALSE: FID has no permission to run, i.e. shall not be executed	
Variation	–		
Possible Errors	E_OK E_NOT_OK		

E.5.2.2 FiM_ControlFunctionAvailable

Name	ControlFunctionAvailable		
Comment	SW Components can use this service to set the availability of a function.		
IsService	true		
Variation	({ecuc(FiM/FiMGeneral/FiMAvailabilitySupport)} == True)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetFunctionAvailable		
Comment	Sets the availability of a function.		
Variation	-		
Parameters	Availability		
	Type	boolean	
	Direction	IN	
	Comment	The permission of the requested FID: TRUE: Function is available. FALSE: Function is not available.	
Variation	-		
Possible Errors	E_OK E_NOT_OK		

E.5.3 Ports

E.5.3.1 FiM_FunctionInhibition

Name	Func_{Name}		
Kind	ProvidedPort	Interface	FunctionInhibition
Description	A client can query the FiM for execution permission for a specific function. The FIDs which represent the functions are not directly used by the client SW-C. Instead, the mechanism of "port-defined argument values" is used and every FID is mapped to a separate port that is responsible for the data exchange via RTE.		
Port Defined Argument Value(s)	Type	FiM_FunctionIdType	
	Value	{ecuc(FiM/FiMConfigSet/FiMFID/FiMFunctionId.value)}	
Variation	Name = {ecuc(FiM/FiMConfigSet/FiMFID.SHORT-NAME)}		

E.5.3.2 FiM_ControlFunctionAvailable

Name	Control_{Name}		
Kind	ProvidedPort	Interface	ControlFunctionAvailable
Description	A client can set the availability for a specific function.		
Port Defined Argument Value(s)	Type	FiM_FunctionIdType	
	Value	{ecuc(FiM/FiMConfigSet/FiMFID/FiMFunctionId.value)}	





Variation	(({ecuc(FiM/FiMGeneral/FiMAvailabilitySupport)} == True) Name = {ecuc(FiM/FiMConfigSet/FiMFID.SHORT-NAME)})
------------------	--

F Changes History

F.1 Changes in R20-11 compared to R19-11

Please note that the document was initially created in R20-11.

F.1.1 Added Traceables in R20-11

[\[SWS_SwCluC_00001\]](#) [\[SWS_SwCluC_00002\]](#) [\[SWS_SwCluC_00003\]](#) [\[SWS_SwCluC_00004\]](#) [\[SWS_SwCluC_00005\]](#) [\[SWS_SwCluC_00006\]](#) [\[SWS_SwCluC_00007\]](#) [\[SWS_SwCluC_00008\]](#) [\[SWS_SwCluC_00009\]](#) [\[SWS_SwCluC_00010\]](#) [\[SWS_SwCluC_00011\]](#) [\[SWS_SwCluC_00012\]](#) [\[SWS_SwCluC_00013\]](#) [\[SWS_SwCluC_00014\]](#) [\[SWS_SwCluC_00015\]](#) [\[SWS_SwCluC_00016\]](#) [\[SWS_SwCluC_00017\]](#) [\[SWS_SwCluC_00018\]](#) [\[SWS_SwCluC_00019\]](#) [\[SWS_SwCluC_00020\]](#) [\[SWS_SwCluC_00021\]](#) [\[SWS_SwCluC_00022\]](#) [\[SWS_SwCluC_00023\]](#) [\[SWS_SwCluC_00024\]](#) [\[SWS_SwCluC_00025\]](#) [\[SWS_SwCluC_00026\]](#) [\[SWS_SwCluC_00027\]](#) [\[SWS_SwCluC_00028\]](#) [\[SWS_SwCluC_00029\]](#) [\[SWS_SwCluC_00030\]](#) [\[SWS_SwCluC_00031\]](#) [\[SWS_SwCluC_00032\]](#) [\[SWS_SwCluC_00033\]](#) [\[SWS_SwCluC_00034\]](#) [\[SWS_SwCluC_00035\]](#) [\[SWS_SwCluC_00036\]](#) [\[SWS_SwCluC_00037\]](#) [\[SWS_SwCluC_00038\]](#) [\[SWS_SwCluC_00039\]](#) [\[SWS_SwCluC_00040\]](#) [\[SWS_SwCluC_00041\]](#) [\[SWS_SwCluC_00042\]](#) [\[SWS_SwCluC_00043\]](#) [\[SWS_SwCluC_00044\]](#) [\[SWS_SwCluC_00045\]](#) [\[SWS_SwCluC_00046\]](#) [\[SWS_SwCluC_00047\]](#) [\[SWS_SwCluC_00048\]](#) [\[SWS_SwCluC_00049\]](#) [\[SWS_SwCluC_00050\]](#) [\[SWS_SwCluC_00051\]](#) [\[SWS_SwCluC_00052\]](#) [\[SWS_SwCluC_00053\]](#) [\[SWS_SwCluC_00054\]](#) [\[SWS_SwCluC_00055\]](#) [\[SWS_SwCluC_00056\]](#) [\[SWS_SwCluC_00057\]](#) [\[SWS_SwCluC_00058\]](#) [\[SWS_SwCluC_00059\]](#) [\[SWS_SwCluC_00060\]](#) [\[SWS_SwCluC_00061\]](#) [\[SWS_SwCluC_00062\]](#) [\[SWS_SwCluC_00063\]](#) [\[SWS_SwCluC_00064\]](#) [\[SWS_SwCluC_00065\]](#) [\[SWS_SwCluC_00066\]](#) [\[SWS_SwCluC_00067\]](#) [\[SWS_SwCluC_00068\]](#) [\[SWS_SwCluC_00069\]](#) [\[SWS_SwCluC_00070\]](#) [\[SWS_SwCluC_00071\]](#) [\[SWS_SwCluC_00072\]](#) [\[SWS_SwCluC_00075\]](#) [\[SWS_SwCluC_00076\]](#) [\[SWS_SwCluC_00077\]](#) [\[SWS_SwCluC_00079\]](#) [\[SWS_SwCluC_00080\]](#) [\[SWS_SwCluC_00081\]](#) [\[SWS_SwCluC_00082\]](#) [\[SWS_SwCluC_00083\]](#) [\[SWS_SwCluC_00084\]](#) [\[SWS_SwCluC_00085\]](#) [\[SWS_SwCluC_00086\]](#) [\[SWS_SwCluC_00088\]](#) [\[SWS_SwCluC_00089\]](#) [\[SWS_SwCluC_00999\]](#) [\[SWS_SwCluC_01000\]](#) [\[SWS_SwCluC_01002\]](#) [\[SWS_SwCluC_02000\]](#) [\[SWS_SwCluC_02001\]](#) [\[SWS_SwCluC_02002\]](#) [\[SWS_SwCluC_02003\]](#) [\[SWS_SwCluC_02004\]](#) [\[SWS_SwCluC_02005\]](#) [\[SWS_SwCluC_02006\]](#) [\[SWS_SwCluC_02007\]](#) [\[SWS_SwCluC_02008\]](#) [\[SWS_SwCluC_02101\]](#) [\[SWS_SwCluC_02102\]](#) [\[SWS_SwCluC_02103\]](#)

[SWS_SwCluC_02104] [SWS_SwCluC_02105] [SWS_SwCluC_02106] [SWS_SwCluC_02107] [SWS_SwCluC_02108] [SWS_SwCluC_02109] [SWS_SwCluC_02110] [SWS_SwCluC_02111] [SWS_SwCluC_02112] [SWS_SwCluC_02113] [SWS_SwCluC_02114] [SWS_SwCluC_02115] [SWS_SwCluC_02116] [SWS_SwCluC_02117] [SWS_SwCluC_02118] [SWS_SwCluC_02119] [SWS_SwCluC_02120] [SWS_SwCluC_02121] [SWS_SwCluC_02122] [SWS_SwCluC_02123] [SWS_SwCluC_02124] [SWS_SwCluC_02125] [SWS_SwCluC_02126] [SWS_SwCluC_02127] [SWS_SwCluC_02128] [SWS_SwCluC_02129] [SWS_SwCluC_02132] [SWS_SwCluC_02133] [SWS_SwCluC_02136] [SWS_SwCluC_02137] [SWS_SwCluC_02138] [SWS_SwCluC_02139] [SWS_SwCluC_02140] [SWS_SwCluC_02142] [SWS_SwCluC_02143] [SWS_SwCluC_02144] [SWS_SwCluC_02200] [SWS_SwCluC_02201] [SWS_SwCluC_02202] [SWS_SwCluC_02204] [SWS_SwCluC_02205] [SWS_SwCluC_02206] [SWS_SwCluC_02210] [SWS_SwCluC_02211] [SWS_SwCluC_02212] [SWS_SwCluC_02213] [SWS_SwCluC_02214] [SWS_SwCluC_02215] [SWS_SwCluC_02216] [SWS_SwCluC_02217] [SWS_SwCluC_02218] [SWS_SwCluC_02219] [SWS_SwCluC_02220] [SWS_SwCluC_02221] [SWS_SwCluC_02222] [SWS_SwCluC_02223] [SWS_SwCluC_02224] [SWS_SwCluC_02225] [SWS_SwCluC_02226] [SWS_SwCluC_02227] [SWS_SwCluC_02229] [SWS_SwCluC_02230] [SWS_SwCluC_02250] [SWS_SwCluC_02251] [SWS_SwCluC_02252] [SWS_SwCluC_02253] [SWS_SwCluC_02254] [SWS_SwCluC_02255] [SWS_SwCluC_02256] [SWS_SwCluC_02257] [SWS_SwCluC_02258] [SWS_SwCluC_02259] [SWS_SwCluC_02260] [SWS_SwCluC_02261] [SWS_SwCluC_02262] [SWS_SwCluC_02263] [SWS_SwCluC_02264] [SWS_SwCluC_02265] [SWS_SwCluC_02266] [SWS_SwCluC_03000] [SWS_SwCluC_03001] [SWS_SwCluC_03002] [SWS_SwCluC_03003] [SWS_SwCluC_03004] [SWS_SwCluC_03005] [SWS_SwCluC_03006] [SWS_SwCluC_03007] [SWS_SwCluC_03008] [SWS_SwCluC_03009] [SWS_SwCluC_03010] [SWS_SwCluC_03011] [SWS_SwCluC_03012] [SWS_SwCluC_03013] [SWS_SwCluC_03015] [SWS_SwCluC_03016] [SWS_SwCluC_03017] [SWS_SwCluC_03018] [SWS_SwCluC_03019] [SWS_SwCluC_03022] [SWS_SwCluC_03023] [SWS_SwCluC_03024] [SWS_SwCluC_03025] [SWS_SwCluC_03026] [SWS_SwCluC_03027] [SWS_SwCluC_03028] [SWS_SwCluC_03029] [SWS_SwCluC_03030] [SWS_SwCluC_03031] [SWS_SwCluC_03033] [SWS_SwCluC_03034] [SWS_SwCluC_03035] [SWS_SwCluC_03036] [SWS_SwCluC_03037] [SWS_SwCluC_03038] [SWS_SwCluC_03039] [SWS_SwCluC_03040] [SWS_SwCluC_03041] [SWS_SwCluC_03042] [SWS_SwCluC_03043] [SWS_SwCluC_03044] [SWS_SwCluC_03045] [SWS_SwCluC_03046] [SWS_SwCluC_03047] [SWS_SwCluC_03048] [SWS_SwCluC_03049] [SWS_SwCluC_03050] [SWS_SwCluC_03051] [SWS_SwCluC_03052] [SWS_SwCluC_03055] [SWS_SwCluC_03056] [SWS_SwCluC_03057] [SWS_SwCluC_03058] [SWS_SwCluC_03060] [SWS_SwCluC_03061] [SWS_SwCluC_03062] [SWS_SwCluC_03063] [SWS_SwCluC_03064] [SWS_SwCluC_03065] [SWS_SwCluC_03068] [SWS_SwCluC_03071] [SWS_SwCluC_03072] [SWS_SwCluC_03073] [SWS_SwCluC_03074] [SWS_SwCluC_03075] [SWS_SwCluC_03076] [SWS_SwCluC_03077] [SWS_SwCluC_03101] [SWS_SwCluC_03102] [SWS_SwCluC_03103] [SWS_SwCluC_03106]

[SWS_SwCluC_03107] [SWS_SwCluC_03108] [SWS_SwCluC_03109] [SWS_SwCluC_03110] [SWS_SwCluC_03111] [SWS_SwCluC_03112] [SWS_SwCluC_03113] [SWS_SwCluC_03120] [SWS_SwCluC_03121] [SWS_SwCluC_03122] [SWS_SwCluC_03123] [SWS_SwCluC_03124] [SWS_SwCluC_03126] [SWS_SwCluC_03130] [SWS_SwCluC_03131] [SWS_SwCluC_03132] [SWS_SwCluC_03133] [SWS_SwCluC_03134] [SWS_SwCluC_03135] [SWS_SwCluC_03136] [SWS_SwCluC_03137] [SWS_SwCluC_03138] [SWS_SwCluC_03139] [SWS_SwCluC_03140] [SWS_SwCluC_03142] [SWS_SwCluC_03143] [SWS_SwCluC_03144] [SWS_SwCluC_03145] [SWS_SwCluC_03146] [SWS_SwCluC_03147] [SWS_SwCluC_03150] [SWS_SwCluC_03151] [SWS_SwCluC_03152] [SWS_SwCluC_03153] [SWS_SwCluC_03154] [SWS_SwCluC_03155] [SWS_SwCluC_03156] [SWS_SwCluC_03157] [SWS_SwCluC_03158] [SWS_SwCluC_03159] [SWS_SwCluC_03160] [SWS_SwCluC_03161] [SWS_SwCluC_03162] [SWS_SwCluC_03163] [SWS_SwCluC_03164] [SWS_SwCluC_03165] [SWS_SwCluC_03166] [SWS_SwCluC_03167] [SWS_SwCluC_03168] [SWS_SwCluC_03169] [SWS_SwCluC_03170] [SWS_SwCluC_03171] [SWS_SwCluC_03173] [SWS_SwCluC_03174] [SWS_SwCluC_03175] [SWS_SwCluC_03176] [SWS_SwCluC_03177] [SWS_SwCluC_10000] [SWS_SwCluC_10001] [SWS_SwCluC_10002] [SWS_SwCluC_10003] [SWS_SwCluC_10004] [SWS_SwCluC_10005] [SWS_SwCluC_10006] [SWS_SwCluC_10007] [SWS_SwCluC_10008] [SWS_SwCluC_10009] [SWS_SwCluC_10010] [SWS_SwCluC_10011] [SWS_SwCluC_10020] [SWS_SwCluC_10021] [SWS_SwCluC_10022] [SWS_SwCluC_10023] [SWS_SwCluC_10032] [SWS_SwCluC_10033] [SWS_SwCluC_10034] [SWS_SwCluC_11000] [SWS_SwCluC_11001] [SWS_SwCluC_12000] [SWS_SwCluC_12100] [SWS_SwCluC_90000] [SWS_SwCluC_90001] [SWS_SwCluC_90002] [SWS_SwCluC_90003] [SWS_SwCluC_90004] [SWS_SwCluC_90005] [SWS_SwCluC_90006] [SWS_SwCluC_90007] [SWS_SwCluC_90008] [SWS_SwCluC_90009] [SWS_SwCluC_90010] [SWS_SwCluC_CONSTR_00073] [SWS_SwCluC_CONSTR_00074] [SWS_SwCluC_CONSTR_00078] [SWS_SwCluC_CONSTR_00087] [SWS_SwCluC_CONSTR_02130] [SWS_SwCluC_CONSTR_02131] [SWS_SwCluC_CONSTR_02134] [SWS_SwCluC_CONSTR_02135] [SWS_SwCluC_CONSTR_02141] [SWS_SwCluC_CONSTR_02203] [SWS_SwCluC_CONSTR_02228] [SWS_SwCluC_CONSTR_02229] [SWS_SwCluC_CONSTR_02230] [SWS_SwCluC_CONSTR_02231] [SWS_SwCluC_CONSTR_02232] [SWS_SwCluC_CONSTR_02233] [SWS_SwCluC_CONSTR_02234] [SWS_SwCluC_CONSTR_02235] [SWS_SwCluC_CONSTR_02236] [SWS_SwCluC_CONSTR_02237] [SWS_SwCluC_CONSTR_03020] [SWS_SwCluC_CONSTR_03021] [SWS_SwCluC_CONSTR_03032] [SWS_SwCluC_CONSTR_03053] [SWS_SwCluC_CONSTR_03054] [SWS_SwCluC_CONSTR_03059] [SWS_SwCluC_CONSTR_03063] [SWS_SwCluC_CONSTR_03066] [SWS_SwCluC_CONSTR_03067] [SWS_SwCluC_CONSTR_03069]

F.1.2 Changed Traceables in R20-11

none

F.1.3 Deleted Traceables in R20-11

none

F.2 Changes in R21-11 compared to R20-11

F.2.1 Added Traceables in R21-11

[SWS_SwCluC_00090] [SWS_SwCluC_02270] [SWS_SwCluC_02271] [SWS_SwCluC_02272] [SWS_SwCluC_02273] [SWS_SwCluC_02274] [SWS_SwCluC_02275] [SWS_SwCluC_02276] [SWS_SwCluC_02277] [SWS_SwCluC_02300] [SWS_SwCluC_02301] [SWS_SwCluC_02302] [SWS_SwCluC_02303] [SWS_SwCluC_02304] [SWS_SwCluC_02305] [SWS_SwCluC_02306] [SWS_SwCluC_02307] [SWS_SwCluC_02308] [SWS_SwCluC_02309] [SWS_SwCluC_02310] [SWS_SwCluC_02311] [SWS_SwCluC_02312] [SWS_SwCluC_02313] [SWS_SwCluC_02314] [SWS_SwCluC_02315] [SWS_SwCluC_02316] [SWS_SwCluC_02317] [SWS_SwCluC_02318] [SWS_SwCluC_02319] [SWS_SwCluC_02320] [SWS_SwCluC_02321] [SWS_SwCluC_02322] [SWS_SwCluC_02323] [SWS_SwCluC_02324] [SWS_SwCluC_02325] [SWS_SwCluC_02330] [SWS_SwCluC_02331] [SWS_SwCluC_02332] [SWS_SwCluC_02333] [SWS_SwCluC_02334] [SWS_SwCluC_02335] [SWS_SwCluC_02336] [SWS_SwCluC_02337] [SWS_SwCluC_02338] [SWS_SwCluC_02339] [SWS_SwCluC_02341] [SWS_SwCluC_02342] [SWS_SwCluC_02350] [SWS_SwCluC_02351] [SWS_SwCluC_02352] [SWS_SwCluC_02353] [SWS_SwCluC_02354] [SWS_SwCluC_02355] [SWS_SwCluC_02356] [SWS_SwCluC_02357] [SWS_SwCluC_02358] [SWS_SwCluC_02359] [SWS_SwCluC_02360] [SWS_SwCluC_02361] [SWS_SwCluC_02362] [SWS_SwCluC_02363] [SWS_SwCluC_02364] [SWS_SwCluC_02365] [SWS_SwCluC_02366] [SWS_SwCluC_02367] [SWS_SwCluC_02368] [SWS_SwCluC_02369] [SWS_SwCluC_02370] [SWS_SwCluC_02371] [SWS_SwCluC_02372] [SWS_SwCluC_02373] [SWS_SwCluC_02374] [SWS_SwCluC_02375] [SWS_SwCluC_02376] [SWS_SwCluC_02380] [SWS_SwCluC_02381] [SWS_SwCluC_02382] [SWS_SwCluC_02383] [SWS_SwCluC_02384] [SWS_SwCluC_02385] [SWS_SwCluC_02386] [SWS_SwCluC_02387] [SWS_SwCluC_02388] [SWS_SwCluC_02390] [SWS_SwCluC_02391] [SWS_SwCluC_02392] [SWS_SwCluC_02393] [SWS_SwCluC_02394] [SWS_SwCluC_02395] [SWS_SwCluC_02396] [SWS_SwCluC_02397] [SWS_SwCluC_02400] [SWS_SwCluC_02401] [SWS_SwCluC_02402] [SWS_SwCluC_02403] [SWS_SwCluC_02404] [SWS_SwCluC_02405] [SWS_SwCluC_02406] [SWS_SwCluC_02407] [SWS_SwCluC_02408] [SWS_SwCluC_02409] [SWS_SwCluC_02410] [SWS_SwCluC_02411] [SWS_SwCluC_02412] [SWS_SwCluC_02413] [SWS_SwCluC_02414] [SWS_SwCluC_02415] [SWS_SwCluC_02416] [SWS_SwCluC_02417] [SWS_SwCluC_02418] [SWS_SwCluC_02419] [SWS_SwCluC_02420] [SWS_SwCluC_02421] [SWS_SwCluC_02422] [SWS_SwCluC_02423] [SWS_SwCluC_02424] [SWS_SwCluC_02425] [SWS_SwCluC_02426] [SWS_SwCluC_02500] [SWS_SwCluC_02501]

[SWS_SwCluC_02502] [SWS_SwCluC_02503] [SWS_SwCluC_02504] [SWS_SwCluC_02505] [SWS_SwCluC_02506] [SWS_SwCluC_02507] [SWS_SwCluC_02508] [SWS_SwCluC_02509] [SWS_SwCluC_02510] [SWS_SwCluC_02511] [SWS_SwCluC_02512] [SWS_SwCluC_02513] [SWS_SwCluC_02514] [SWS_SwCluC_02515] [SWS_SwCluC_02516] [SWS_SwCluC_02517] [SWS_SwCluC_02518] [SWS_SwCluC_02519] [SWS_SwCluC_02520] [SWS_SwCluC_02521] [SWS_SwCluC_02522] [SWS_SwCluC_02523] [SWS_SwCluC_02524] [SWS_SwCluC_02525] [SWS_SwCluC_02526] [SWS_SwCluC_02527] [SWS_SwCluC_02528] [SWS_SwCluC_02529] [SWS_SwCluC_02530] [SWS_SwCluC_02531] [SWS_SwCluC_02532] [SWS_SwCluC_02533] [SWS_SwCluC_02534] [SWS_SwCluC_02535] [SWS_SwCluC_02536] [SWS_SwCluC_02537] [SWS_SwCluC_02538] [SWS_SwCluC_02539] [SWS_SwCluC_02540] [SWS_SwCluC_02541] [SWS_SwCluC_02542] [SWS_SwCluC_02543] [SWS_SwCluC_02544] [SWS_SwCluC_02545] [SWS_SwCluC_02548] [SWS_SwCluC_02549] [SWS_SwCluC_02550] [SWS_SwCluC_02551] [SWS_SwCluC_02552] [SWS_SwCluC_02600] [SWS_SwCluC_02601] [SWS_SwCluC_02602] [SWS_SwCluC_02603] [SWS_SwCluC_02604] [SWS_SwCluC_02605] [SWS_SwCluC_02606] [SWS_SwCluC_02607] [SWS_SwCluC_02608] [SWS_SwCluC_02609] [SWS_SwCluC_02610] [SWS_SwCluC_02611] [SWS_SwCluC_02612] [SWS_SwCluC_02613] [SWS_SwCluC_02614] [SWS_SwCluC_02615] [SWS_SwCluC_02616] [SWS_SwCluC_02617] [SWS_SwCluC_02618] [SWS_SwCluC_02619] [SWS_SwCluC_02620] [SWS_SwCluC_02621] [SWS_SwCluC_02622] [SWS_SwCluC_02623] [SWS_SwCluC_02624] [SWS_SwCluC_02625] [SWS_SwCluC_02626] [SWS_SwCluC_02627] [SWS_SwCluC_02628] [SWS_SwCluC_02629] [SWS_SwCluC_02630] [SWS_SwCluC_02631] [SWS_SwCluC_02632] [SWS_SwCluC_02633] [SWS_SwCluC_02634] [SWS_SwCluC_02636] [SWS_SwCluC_02637] [SWS_SwCluC_02638] [SWS_SwCluC_02639] [SWS_SwCluC_02640] [SWS_SwCluC_02642] [SWS_SwCluC_02643] [SWS_SwCluC_02644] [SWS_SwCluC_02645] [SWS_SwCluC_02646] [SWS_SwCluC_02647] [SWS_SwCluC_02648] [SWS_SwCluC_02649] [SWS_SwCluC_02650] [SWS_SwCluC_02651] [SWS_SwCluC_02652] [SWS_SwCluC_02661] [SWS_SwCluC_02662] [SWS_SwCluC_02663] [SWS_SwCluC_02700] [SWS_SwCluC_02701] [SWS_SwCluC_02702] [SWS_SwCluC_02703] [SWS_SwCluC_02704] [SWS_SwCluC_02705] [SWS_SwCluC_02706] [SWS_SwCluC_02707] [SWS_SwCluC_02708] [SWS_SwCluC_02709] [SWS_SwCluC_02710] [SWS_SwCluC_02711] [SWS_SwCluC_02712] [SWS_SwCluC_02713] [SWS_SwCluC_02714] [SWS_SwCluC_02715] [SWS_SwCluC_02716] [SWS_SwCluC_02717] [SWS_SwCluC_02718] [SWS_SwCluC_02719] [SWS_SwCluC_02723] [SWS_SwCluC_02724] [SWS_SwCluC_02725] [SWS_SwCluC_02726] [SWS_SwCluC_02727] [SWS_SwCluC_02728] [SWS_SwCluC_02729] [SWS_SwCluC_02730] [SWS_SwCluC_02731] [SWS_SwCluC_02732] [SWS_SwCluC_02733] [SWS_SwCluC_02734] [SWS_SwCluC_02735] [SWS_SwCluC_02736] [SWS_SwCluC_02737] [SWS_SwCluC_02738] [SWS_SwCluC_02739] [SWS_SwCluC_02740] [SWS_SwCluC_02741] [SWS_SwCluC_02742] [SWS_SwCluC_02743] [SWS_SwCluC_02744] [SWS_SwCluC_02745] [SWS_SwCluC_02746] [SWS_SwCluC_02747] [SWS_SwCluC_02748] [SWS_SwCluC_02749]

[SWS_SwCluC_02750] [SWS_SwCluC_02751] [SWS_SwCluC_03080] [SWS_SwCluC_03081] [SWS_SwCluC_03082] [SWS_SwCluC_03200] [SWS_SwCluC_03201] [SWS_SwCluC_03202] [SWS_SwCluC_03203] [SWS_SwCluC_03204] [SWS_SwCluC_03205] [SWS_SwCluC_03206] [SWS_SwCluC_03207] [SWS_SwCluC_03208] [SWS_SwCluC_03209] [SWS_SwCluC_03210] [SWS_SwCluC_03211] [SWS_SwCluC_03212] [SWS_SwCluC_03213] [SWS_SwCluC_03214] [SWS_SwCluC_03215] [SWS_SwCluC_03216] [SWS_SwCluC_03217] [SWS_SwCluC_03218] [SWS_SwCluC_12200] [SWS_SwCluC_12300] [SWS_SwCluC_12400] [SWS_SwCluC_13000] [SWS_SwCluC_13001] [SWS_SwCluC_13002] [SWS_SwCluC_13003] [SWS_SwCluC_13004] [SWS_SwCluC_13005] [SWS_SwCluC_13006] [SWS_SwCluC_13007] [SWS_SwCluC_13008] [SWS_SwCluC_13009] [SWS_SwCluC_13010] [SWS_SwCluC_13011] [SWS_SwCluC_13012] [SWS_SwCluC_13013] [SWS_SwCluC_13014] [SWS_SwCluC_13015] [SWS_SwCluC_90011] [SWS_SwCluC_90012] [SWS_SwCluC_90013] [SWS_SwCluC_90014] [SWS_SwCluC_90015] [SWS_SwCluC_90016] [SWS_SwCluC_90017] [SWS_SwCluC_90018] [SWS_SwCluC_90019] [SWS_SwCluC_90020] [SWS_SwCluC_90021] [SWS_SwCluC_90022] [SWS_SwCluC_CONSTR_00091] [SWS_SwCluC_CONSTR_00092] [SWS_SwCluC_CONSTR_00093] [SWS_SwCluC_CONSTR_00094] [SWS_SwCluC_CONSTR_02427] [SWS_SwCluC_CONSTR_02428] [SWS_SwCluC_CONSTR_02429] [SWS_SwCluC_CONSTR_02546] [SWS_SwCluC_CONSTR_02547] [SWS_SwCluC_CONSTR_02553] [SWS_SwCluC_CONSTR_02554] [SWS_SwCluC_CONSTR_02555] [SWS_SwCluC_CONSTR_02556] [SWS_SwCluC_CONSTR_02557] [SWS_SwCluC_CONSTR_02558] [SWS_SwCluC_CONSTR_02559] [SWS_SwCluC_CONSTR_02560] [SWS_SwCluC_CONSTR_02561] [SWS_SwCluC_CONSTR_02562] [SWS_SwCluC_CONSTR_02653] [SWS_SwCluC_CONSTR_02654] [SWS_SwCluC_CONSTR_02655] [SWS_SwCluC_CONSTR_02656] [SWS_SwCluC_CONSTR_02657] [SWS_SwCluC_CONSTR_02658] [SWS_SwCluC_CONSTR_02659] [SWS_SwCluC_CONSTR_02660] [SWS_SwCluC_CONSTR_02664] [SWS_SwCluC_CONSTR_02720] [SWS_SwCluC_CONSTR_02721] [SWS_SwCluC_CONSTR_02752] [SWS_SwCluC_CONSTR_03210] [SWS_SwCluC_CONSTR_03211]

F.2.2 Changed Traceables in R21-11

[SWS_SwCluC_00005] [SWS_SwCluC_00018] [SWS_SwCluC_00019] [SWS_SwCluC_00031] [SWS_SwCluC_00040] [SWS_SwCluC_00086] [SWS_SwCluC_00999] [SWS_SwCluC_02259] [SWS_SwCluC_02262] [SWS_SwCluC_02263] [SWS_SwCluC_02264] [SWS_SwCluC_03035] [SWS_SwCluC_10011]

F.2.3 Deleted Traceables in R21-11

[SWS_SwCluC_00038] [SWS_SwCluC_00039] [SWS_SwCluC_00071] [SWS_SwCluC_00084] [SWS_SwCluC_02257] [SWS_SwCluC_02258] [SWS_SwCluC_02260] [SWS_SwCluC_02261] [SWS_SwCluC_CONSTR_00074]