

SCOPE OF APPLICATION All Project/Engineering		SHT/SHTS 1 / 146
Responsibility: Classic AUTOSAR Team 2	mobilgene Classic R44 Module User Manual	DOC. NO
Rte User Manual		

Document Change History			
Date (YYYY-MM-DD)	Ver.	Editor	Description
2024.05.15	1.8.0.0	DanhDC1 DongHyeon Lee SeungChul Kwon	<ul style="list-style-type: none"> <li>• Add validation rule SAFERTE_ERR_MC016, SAFERTE_ERR_0371, SAFERTE_ERR_GEN0003</li> <li>• Improve Warning/Error log – remove duplicates, readability</li> <li>• Update restriction of measurement support</li> <li>• Change the output Bswmd_Rte.arxml generated file</li> </ul>
2024.04.15	1.7.0.0	DongHyeon Lee KiYoung Yun SeungChul Kwon SeungJin Noh	<ul style="list-style-type: none"> <li>• Improve Copy IWrite/IRead function are not called in OIE direct function calls</li> <li>• Improve duplicate generation of NvBlock Runnable / NvDataUpdateFlag</li> <li>• Change prefix Rte_Gst_CSSignalProps_ComPartition</li> <li>• Improve Rte generator speed.</li> <li>• Improve compare logic between two operations when using Client-Server Interface mappings</li> <li>• Improve code mis alignment</li> <li>• Improve null-exception during generator operation when using PR-Port</li> <li>• Improve null-exception during generator operation when using R40 OS</li> <li>• Improve validation error message</li> <li>• Add Suspend/Resume Interrupts to part copy request mode into current mode in Rte_ModeMachine API</li> <li>• Remove Product Release Note from User Manual</li> <li>• Add configuration guide in User Manual</li> <li>• Add category of function user (user or BSW) to API description of User Manual</li> <li>• Remove deprecated parameter from PDF</li> <li>• Clarify features not covered in Rte</li> </ul>
2024.03.22	1.6.0.0_HF1	SeungChul Kwon	<ul style="list-style-type: none"> <li>• Improve RTE_E_SEG_FAULT conditional expression</li> </ul>
2023.12.15	1.6.0.0	KiYoung Yun SeungJin Noh SeungChul Kwon	<ul style="list-style-type: none"> <li>• Support Schedule Point</li> <li>• Support NonPreemptiveOsTaskSchedule – SenderReceiver Communication</li> <li>• Support for missing Bsw Include Path option for Rte in SCons.arxml in Rte.bat</li> <li>• Improve validation error message</li> </ul>
2023.11.30	1.5.1.0	SeungChul Kwon SeungJin Noh HunDong Lee	<ul style="list-style-type: none"> <li>• Support MULTI-SpinLock</li> <li>• Improve to use of SpinLock in Rte_IWrite function</li> <li>• Improve declaration method of Struct Array in Rte_Type.h</li> <li>• Improve to prevent Null Pointer Exception errors of the generator</li> <li>• Improve to be callable Rte_SetMirror before Rte_Start and Rte_GetMirror after Rte_Stop</li> </ul>

Edition Date: 2024-05-15	File Name RTE_UM.pdf	Creation SC Kwon	Check HM Kim	Approval JS Jang
Document Management System		2024/05/15	2024/05/15	2024/05/15

# Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
2 / 146

## Document Change History

Date (YYYY-MM-DD)	Ver.	Editor	Description
			<ul style="list-style-type: none"> <li>Delete Rte_GetMirror/Rte_SetMirror ReturnType RTE_E_UNINIT</li> <li>Fix bug where using NvM_ReadAll before Rte_Start with NvBlockSWC due to initialize ram buffer of NvBlockDescriptor</li> <li>Support setting options specified in Rte.bat in Scons</li> <li>Improve Rte Generator speed</li> </ul>
2023.07.28	1.5.0.0	KiYoung Yun	<ul style="list-style-type: none"> <li>Support Client Server InterECU Communication with LDCOM_TP</li> <li>Support Client Server Autonomous Reaction</li> <li>Improve performance of Rte_MemCpy and Rte_MemSet</li> <li>Change the order of declaration of data types in Rte_Type.h</li> <li>Change arguments of SomelpXf API in case of Autonomous Reaction</li> <li>Improve to prevent Null Pointer Exception errors of the generator</li> </ul>
2023.06.30	1.4.0.1	SeungJin Noh OhEun Kwon KiYoung Yun	<ul style="list-style-type: none"> <li>Add option to support ARM compiler</li> <li>Improve Compatibility rule of ClientServerOperation</li> <li>Improve Rte Generation time</li> <li>Fix Values of ModeDeclarationGroup Generation rule</li> <li>Change naming rule about &lt;re&gt; in implicit Rte API</li> <li>Support Vs/Vts for ApplicationValueSpec</li> <li>Improve naming runnable entity arguments</li> <li>Improve generation of VFB trace events</li> <li>Improve CS APIs of bsw instance with vendorId and vendorApiInfix</li> <li>Add property base type encoding to NONE for uint8_base and uint32_base in os_ioc.arxml</li> </ul>
2023.01.09	1.4.0.0	KiYoung Yun SeungMo Koo	<ul style="list-style-type: none"> <li>Add Post Build feature</li> <li>Improve unexpected operation of RteGen because of NPE</li> <li>Improve OutofMemory Error message</li> </ul>
2022.09.06	1.3.3.0	SeungMo Koo	<ul style="list-style-type: none"> <li>Improve to satisfy cyber security static analysis rule</li> <li>Improve to generate always Os_ioc.arxml</li> <li>Fixing Rte Schema version</li> </ul>
2022.06.30	1.3.2.0	SeungMo Koo	<ul style="list-style-type: none"> <li>Improve SAFERTE_ERR_TRG005 validation logic</li> <li>Improve unexpected error message of generator</li> <li>Improve to generate glucode for a runnable triggered by OIE</li> </ul>
2021-12-29	1.3.1.0	SeungMo Koo	<ul style="list-style-type: none"> <li>Change code to satisfy cyber security law of the UNECE</li> <li>Change Error Code SAFE_ERR_0000 to SAFE_ERR_GEN_0001</li> <li>Add Error Code SAFE_ERR_GEN_0002</li> <li>Fix Rte.bat file</li> </ul>
2021-10-19	1.3.0.0	SeungMo Koo	<ul style="list-style-type: none"> <li>Support different time value of the synchronous server call point</li> <li>Improve the Splitable logic of the generator</li> <li>The default generated codes for the Rte_Enter/Exit and the SchM_Enter/Exit are macro.</li> </ul>
2021-09-30	1.2.6.0	SeungMo Koo	<ul style="list-style-type: none"> <li>Fix Rte PDF Reference proxy url</li> <li>Fix SchM_Enter/Exit VendorPrefix Generation</li> </ul>
2021-08-31	1.2.5.0	KiYoung Yun	<ul style="list-style-type: none"> <li>Fix the location of initialization of SchM Mode Machine instance</li> <li>Fix NullPointerException when Alivetimeout is Null</li> </ul>

# Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
3 / 146

## Document Change History

Date (YYYY-MM-DD)	Ver.	Editor	Description
2021-08-09	1.2.4.0	SeungMo Koo	<ul style="list-style-type: none"> <li>Fix the Rte_IntClientCbk Generation Condition</li> </ul>
2021-07-30	1.2.3.0	SeungMo Koo KiYoung Yun	<ul style="list-style-type: none"> <li>Project root path has been removed in the generated from of the c source/haeder code</li> <li>Add the project_root execution options</li> <li>Remove the project root path in the Generated From field in the Generated Code</li> <li>Fix the generated location when satisfy SAFERTE_WARN_0036</li> <li>Fix the queue algorithm for client server communication</li> <li>Fix the improper mode disabling operation</li> </ul>
2021-06-22	1.2.2.0	SeungMo Koo	<ul style="list-style-type: none"> <li>Improve the generation time</li> <li>Add SAFERTE_ERR_0000</li> </ul>
2021-06-01	1.2.1.0	KiYoung Yun	<ul style="list-style-type: none"> <li>#18728 - print warning message when checkCSTimeout option is not configured</li> <li>#18718 - Change validation for checking positionintask regarding Mode Switch Event</li> </ul>
2021-05-06	1.2.0.1	SeungMo Koo	<ul style="list-style-type: none"> <li>Rte PDF 내 ArReleaseInformation 추가</li> </ul>
2021-04-22	1.2.0	KiYoung Yun	<ul style="list-style-type: none"> <li>#18308 - Support Cs Inter Partition Communication without Timeout Configuration</li> <li>#18567 - Ecud_Ecuc에 Partition 설정이 없는 경우 Rte_Memmap.h 파일 생성 수정 (skipOsTaskSection 옵션 추가)</li> </ul>
2021-04-05	1.1.2	SeungMo Koo	<ul style="list-style-type: none"> <li>#18193 - Rte_IllegalInvocationCheck_Irv Syntax 오류 수정</li> <li>#18424 - 서로 다른 Partition의 component 사이에 S-C Interface를 사용할 때 생성되는 Data Type</li> <li>#18342 - Rte UninitDet 가 SchM_Trigger, SchM_ActMainFunction에서 생성되지 않는 오류 수정</li> <li>#18268 - Rte Buffer 연산 괄호 생성 오류 수정(SomelpXf 사용 시 컴파일 오류 발생)</li> <li>#18449 - Rte_ERR_0366 Check Logic 개선</li> </ul>
2021-03-05	1.1.1	SeungMo Koo	<ul style="list-style-type: none"> <li>Improvement E2EXf / SomelpXf Xfrm Chain Buffer calculation</li> <li>Improvement Validation Rule</li> <li>SAFERTE_WARN_0368 --&gt; SAFERTE_ERR_0368</li> <li>SAFERTE_ERR_0155, SAFERTE_ERR_0156, SAFERTE_ERR_0157, SAFERTE_ERR_0164, SAFERTE_ERR_MSW014</li> <li>#27550, #27789, #27526, #27392, #27074, #26944, #26942, #26895, #27185, #27332, #27562, #27830, #27541, #26709</li> </ul>
2020-12-15	1.1.0	SeungMo Koo	<ul style="list-style-type: none"> <li>generateNoASIL 실행 옵션 추가</li> <li>EventFlagType 생성 시 VendorAPIInfix, VendorId 붙여서 생성하도록 변경</li> <li>#24638, #25010, #25811, #24166, #27091, #26944, #27074</li> </ul>
2020-12-04	1.0.0	SeungMo Koo	<ul style="list-style-type: none"> <li>Init Version</li> </ul>

## Table of Contents

<b>1. OVERVIEW .....</b>	<b>9</b>
<b>2. ABBREVIATIONS .....</b>	<b>9</b>
<b>3. INTRODUCTION .....</b>	<b>11</b>
<b>3.1 Control Flow for Rte .....</b>	<b>11</b>
<b>3.2 Scope .....</b>	<b>12</b>
<b>3.3 RTE Features .....</b>	<b>12</b>
3.3.1 Features covered in Rte include .....	13
3.3.1.1 Post Build Features implemented in Rte include .....	13
3.3.2 Features not covered in Rte include .....	15
3.3.2.1 Post Build Features not implemented in Rte include .....	16
<b>3.4 Additional/Unique/Custom features of Safe RTE .....</b>	<b>16</b>
3.4.1 Minimum AUTOSAR Schema Version Support .....	16
3.4.2 Null pointer argument detection for all RTE and SchM APIs .....	16
3.4.3 OS service failure detection and reporting .....	16
3.4.4 Pre-Initialization and Post-Deinitialization run-time checks for APIs and Tasks .....	16
3.4.5 Illegal invocation detection for RTE/SchM APIs .....	17
3.4.6 Invalid mode value detection in <Rte/SchM>_Switch API .....	17
3.4.7 DET reporting for incorrect sequence and NULL arguments from RTE LdCom Callbacks .....	17
3.4.8 DET Reporting for Transformer failure from Trigger APIs .....	17
3.4.9 RTE/SchM Initialization Policy .....	17
3.4.10 API selection for starting an OsScheduleTable .....	18
3.4.11 OsScheduleTable usage for periodic entities .....	18
3.4.12 WaitPoint handling and configuration .....	19
3.4.13 Restrictions for PortInterfaceMapping .....	20
3.4.14 Memory mapping of RTE lower layers in multi-partition RTE .....	20
3.4.15 Memory mapping of Lifecycle API's in multi-partition RTE .....	21
3.4.16 Cross-core communication and timeout .....	22
3.4.17 OsNeeds generation for multi-partition configuration .....	22
3.4.18 McSupportData generation in RTE BSWMD file .....	23
3.4.19 RTE buffer handling when Transformer is configured .....	24
3.4.20 Partition Restart .....	24
3.4.21 Postbuild generation phase not existing .....	25
3.4.22 VFB Tracing clients ordering .....	25
3.4.23 Schedule Point .....	25
<b>3.5 Pre-Conditions .....</b>	<b>26</b>
<b>4. CONFIGURATION GUIDE .....</b>	<b>26</b>
<b>4.1 RteGeneration .....</b>	<b>26</b>
<b>4.2 RteSwComponentType .....</b>	<b>28</b>

4.2.1 RteComponentTypeCalibration .....	28
<b>4.3 RteSwComponentInstance .....</b>	<b>28</b>
4.3.1 RteEventToTaskMapping.....	28
4.3.2 RteExclusiveAreaImplementation.....	30
4.3.3 RteExternalTriggerConfig .....	31
4.3.4 RteInternalTriggerConfig.....	31
4.3.5 RteNvRamAllocation.....	31
<b>4.4 RteBswModuleInstance.....</b>	<b>32</b>
4.4.1 RteBswEventToTaskMapping.....	32
4.4.2 RteBswExclusiveAreaImpl.....	34
4.4.3 RteBswRequiredClientServerConnection .....	34
4.4.4 RteBswRequiredSenderReceiverConnection .....	35
4.4.5 RteBswRequiredModeGroupConnection .....	35
4.4.6 RteBswRequiredTriggerConnection .....	36
4.4.7 RteBswExternalTriggerConfig .....	36
4.4.8 RteBswInternalTriggerConfig .....	37
<b>4.5 RteOsInteraction .....</b>	<b>37</b>
4.5.1 RteSyncPoint .....	37
4.5.2 RteUsedOsActivation.....	37
4.5.3 RteModeToScheduleTableMapping .....	38
4.5.3.1 RteModeSchtblMapBsw .....	38
4.5.3.2 RteModeSchtblMapSwc .....	39
<b>4.6 RteBswGeneral .....</b>	<b>39</b>
<b>4.7 CommonPublishedInformation .....</b>	<b>39</b>
<b>4.8 RteImplicitCommunication .....</b>	<b>39</b>
<b>4.9 RtePostBuildVariantConfiguration .....</b>	<b>40</b>
<b>4.10 RteInitializationBehavior .....</b>	<b>40</b>
<b>4.11 RteInitializationRunnableBatch.....</b>	<b>40</b>
<b>5. APPLICATION PROGRAMMING INTERFACE (API) .....</b>	<b>40</b>
<b>5.1 Type Definitions.....</b>	<b>40</b>
5.1.1 Predefined Error Codes .....	40
<b>5.2 Interaction with another module .....</b>	<b>43</b>
5.2.1 OS.....	43
5.2.2 COM.....	43
5.2.3 LdCom.....	43
5.2.4 DET .....	43
5.2.5 Transformer Modules .....	43
<b>5.3 Detailed Description of RTE.....</b>	<b>44</b>

5.3.1 SchM Lifecycle APIs .....	44
5.3.1.1 SchM_Init .....	44
5.3.1.2 SchM_DelInit .....	44
5.3.1.3 SchM_Start.....	44
5.3.1.4 SchM_StartTiming.....	45
5.3.1.5 SchM_GetVersionInfo .....	45
5.3.2 RTE Lifecycle APIs .....	45
5.3.2.1 Rte_Start .....	46
5.3.2.2 Rte_Stop .....	46
5.3.2.3 Rte_Init .....	46
5.3.2.4 Rte_StartTiming .....	47
5.3.2.5 Rte_PartitionTerminated_<PID> .....	47
5.3.2.6 Rte_PartitionRestarting_<PID> .....	47
5.3.2.7 Rte_RestartPartition_<PID> .....	48
5.3.3 Sender Receiver Communication APIs .....	48
5.3.3.1 Rte_DRead .....	48
5.3.3.2 Rte_Feedback .....	49
5.3.3.3 Rte_Invalidate .....	50
5.3.3.4 Rte_Write .....	51
5.3.3.5 Rte_IsUpdated .....	53
5.3.3.6 Rte_Read .....	53
5.3.3.7 Rte_Send .....	54
5.3.3.8 Rte_Receive .....	56
5.3.3.9 Rte_IRead .....	57
5.3.3.10 Rte_IWrite.....	58
5.3.3.11 Rte_IWriteRef .....	58
5.3.3.12 Rte_IInvalidate .....	58
5.3.3.13 Rte_IStatus .....	59
5.3.3.14 Rte_IFeedback.....	60
5.3.4 Client-Server Communication APIs .....	62
5.3.4.1 Rte_Call.....	62
5.3.4.2 Rte_Result .....	63
5.3.5 RTE Exclusive Area APIs.....	64
5.3.5.1 Rte_Enter .....	64
5.3.5.2 Rte_Exit.....	65
5.3.6 COM Callbacks .....	65
5.3.6.1 Rte_COMCbK_<sn> .....	65
5.3.6.2 Rte_COMCbK_<sg> .....	65
5.3.6.3 Rte_COMCbKTack_<sn> .....	66
5.3.6.4 Rte_COMCbKTack_<sg> .....	66
5.3.6.5 Rte_COMCbKTErr_<sn> .....	66
5.3.6.6 Rte_COMCbKTErr_<sg> .....	67
5.3.6.7 Rte_COMCbKInv_<sn> .....	67
5.3.6.8 Rte_COMCbKInv_<sg> .....	67
5.3.6.9 Rte_COMCbKRxTOut_<sn> .....	68
5.3.6.10 Rte_COMCbKRxTOut_<sg>.....	68
5.3.6.11 Rte_COMCbKTxTOut_<sn> .....	68
5.3.6.12 Rte_COMCbKTxTOut_<sg> .....	69
5.3.7 LdCOM Callbacks .....	69
5.3.7.1 Rte_LdComCbKRxIndication_<sn> .....	69
5.3.7.2 Rte_LdComCbKStartOfReception_<sn> .....	70
5.3.7.3 Rte_LdComCbKCopyRxData_<sn> .....	71
5.3.7.4 Rte_LdComCbKTpRxIndication_<sn> .....	71

5.3.7.5 Rte_LdComCbkCopyTxData_<sn> .....	72
5.3.7.6 Rte_LdComCbkTpTxConfirmation_<sn> .....	73
5.3.7.7 Rte_LdComCbkTriggerTransmit_<sn> .....	73
5.3.7.8 Rte_LdComCbkTxConfirmation_<sn> .....	74
5.3.8 NVM Service Callbacks.....	74
5.3.8.1 Rte_SetMirror .....	74
5.3.8.2 Rte_GetMirror .....	75
5.3.8.3 Rte_NvMNotifyJobFinished .....	75
5.3.8.4 Rte_NvMNotifyInitBlock .....	76
5.3.9 SchM Exclusive Area APIs .....	76
5.3.9.1 SchM_Enter .....	76
5.3.9.2 SchM_Exit .....	77
5.3.10 Inter Runnable Variables APIs .....	77
5.3.10.1 Rte_IrvIRead .....	77
5.3.10.2 Rte_IrvIWrite .....	77
5.3.10.3 Rte_IrvIWriteRef .....	78
5.3.10.4 Rte_IrvRead .....	78
5.3.10.5 Rte_IrvWrite.....	79
5.3.11 RTE Mode Switch APIs .....	79
5.3.11.1 Rte_Switch.....	80
5.3.11.2 Rte_SwitchAck .....	80
5.3.11.3 Rte_Mode .....	81
5.3.11.4 Rte_Mode (Enhanced).....	81
5.3.12 SchM Mode Switch APIs .....	82
5.3.12.1 SchM_Switch .....	82
5.3.12.2 SchM_SwitchAck .....	83
5.3.12.3 SchM_Mode .....	83
5.3.12.4 SchM_Mode (Enhanced) .....	84
5.3.13 RTE Trigger APIs.....	84
5.3.13.1 Rte_Trigger .....	84
5.3.13.2 Rte_IrTrigger .....	85
5.3.14 SchM Trigger APIs .....	85
5.3.14.1 SchM_Trigger .....	85
5.3.14.2 SchM_ActMainFuntion.....	86
5.3.15 RTE Calibration APIs .....	87
5.3.15.1 Rte_Pim.....	87
5.3.15.2 Rte_CData .....	87
5.3.15.3 Rte_Prm .....	88
5.3.16 SchM Calibration APIs .....	88
5.3.16.1 SchM_Pim.....	88
5.3.16.2 SchM_CData .....	89
5.3.17 SchM Sender Receiver APIs .....	89
5.3.17.1 SchM_Send .....	89
5.3.17.2 SchM_Receive .....	90
5.3.18 SchM Client Server APIs .....	90
5.3.18.1 SchM_Call.....	91
5.3.18.2 SchM_Result .....	91

## 6. CODE GENERATOR.....92

### 6.1 Inputs ..... 92

<b>6.2 Output .....</b>	<b>93</b>
6.2.1 Inc folder .....	93
6.2.2 Src folder.....	95
6.2.3 Xml folder.....	95
<b>6.3 CLI Checks .....</b>	<b>96</b>
<b>6.4 CLI Usage .....</b>	<b>96</b>
<b>6.5 Set CLI and Java Heap Memory in SCons .....</b>	<b>100</b>
6.5.1 Fixed options .....	100
6.5.2 Modifiable options .....	100
6.5.2.1 Prefix .....	101
6.5.2.2 [HYP].....	101
6.5.2.3 [MOD].....	102
6.5.2.4 [GEN].....	102
6.5.2.5 [XMS] and [XMX].....	102
<b>7. DEBUGGING AND TROUBLESHOOTING .....</b>	<b>102</b>
<b>7.1 Code Generator Consistency Checks .....</b>	<b>103</b>
7.1.1 Common / Generic Error Messages .....	104
7.1.2 Sender Receiver Error Messages .....	122
7.1.3 Client Server Error Messages.....	123
7.1.4 Mode Switch Error Messages.....	128
7.1.5 Trigger Error Messages .....	133
7.1.6 Nv Data Error Messages.....	136
7.1.7 Calibration Error Messages .....	138
7.1.8 Multi Partition Error Messages .....	139
7.1.9 PortInterfaceMapping Error Messages.....	142
7.1.10 Warning Messages.....	143



## 1. Overview

본 문서는 AUTOSAR 표준 개발 방법론에 의해 설계된 Application 디자인 문서 (Software Component Description arxml) 기반의 RTE (Runtime Environment) Layer에 해당하는 코드 생성을 위한, 설정 및 각종 제반 사항에 대하여 설명한다.

## 2. Abbreviations

The abbreviations used in RTE user manual, and their meanings are given in table below:

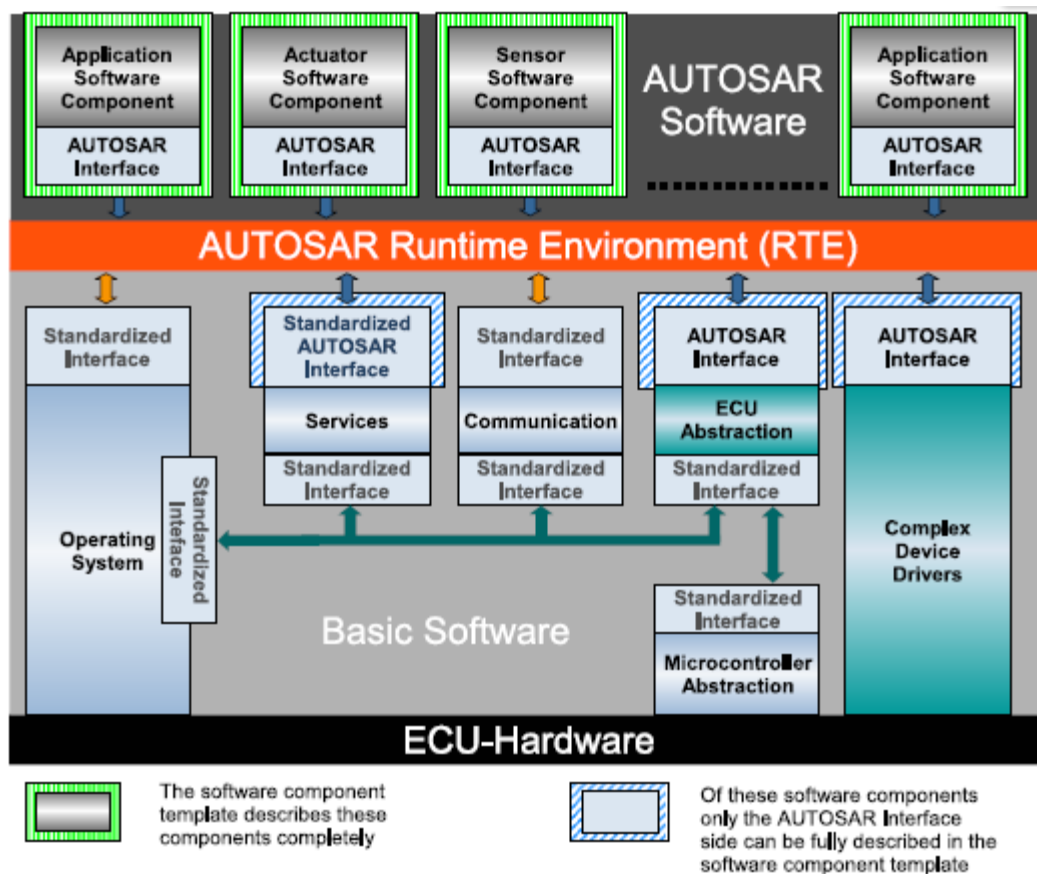
Abbreviation	Meaning
API	Application Programming Interface
ARTOP	AUTOSAR Tool Platform User Group
ARXML/arxml	AutosAR eXtensible Markup Language
ASIL	Automotive Safety Integrity Level
AUTOSAR	AUTomotive Open System ARchitecture
BSW	Basic Software
BSWMD	Basic Software Module Description
CLI	Command Line Interface
COM	AUTOSAR COM module
CPU	Central Processing Unit
CS	Client Server
DET	Default Error Tracer
ECU	Electronic Control Unit
EcuC	ECU Configuration
EA	Enterprise Architecture
I-PDU	Interaction Layer PDU. An I-PDU consists of data, length and I-PDU ID
IOC	Inter OS-Application Communicator
IDT	Implementation Data Type
JRE	Java Runtime Environment
NVM	Non-Volatile Memory

NVRAM	Non-Volatile RAM
N-SDU	Network Layer SDU
N-PDU	Network Layer PDU
OS	Operating System
PDF	Parameter Definition File
PDU	Protocol Data Unit
PDUR	PDU Router module
PDP	Parameter Data Prototype
PID	Prototype Identifier
PIM	Per Instance Memory
PRM	Parameter
POC	Points of organizational contact
RAM	Random Access Memory
RTE	Run Time Environment
SDU	Service Data Unit
SCHM	Schedule Manager
SR	Sender Receiver
SVN	Sub Version
SWC	Software Component
SWS	Software Specification
UM	User Manual
VFB	Virtual Function Bus

**Table 1: Abbreviations**

### 3. Introduction

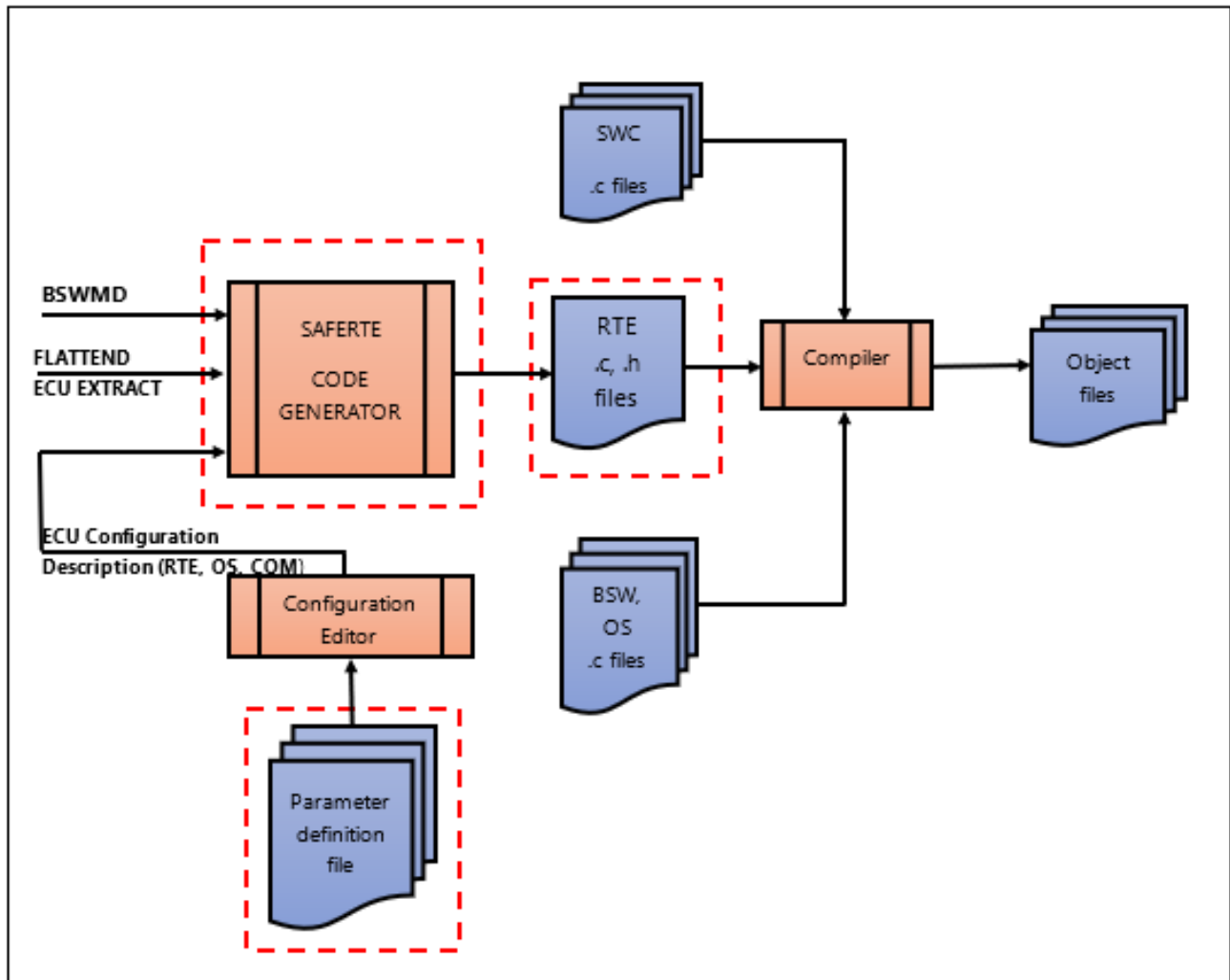
Rte ensures safe communication between software components and basic software and the scheduling of the software components and basic software modules. The Rte is a code generator tool that generates embedded RTE code. The generated code is compliant to ASIL-D standards. The standard AUTOSAR architecture given in below diagram is considered.



**Figure 1 – Rte in AUTOSAR Architecture**

#### 3.1 Control Flow for Rte

The Rte development will be based on AUTOSAR RTE requirement version 4.4 and the Assumed Software Safety Requirements. Rte code generator supports CLI Executable. Given below is the Control Flow diagram for Rte code generation.



**Figure 2: Rte Control Flow Diagram**

## 3.2 Scope

The Rte User Manual, is a guidance to user for usage of RTE code generator tool. It provides the necessary information regarding the inputs, outputs and how the code can be generated. It also describes how the generated embedded code can be utilized. It describes the dependencies with other modules. This document also covers safety features considered in the tool.

## 3.3 RTE Features

Rte development is divided into two main parts, the generation tool and the embedded code. The Design for generation tool and the embedded code have been done using EA. The code generator is developed on Java based XTEND code. Command Line plugin invokes RTE code generator. The plugin shall also refer various arxml files as input for the RTE generation. The code generator validates the input arxml files and only when

there are no validation errors, the embedded code is generated. Embedded code is in C (as per AUTOSAR specification).

### 3.3.1 Features covered in Rte include

1. RTE and SchM Life Cycle APIs
2. RTE Sender Receiver non-queued explicit communication
3. RTE Sender Receiver queued explicit communication
4. RTE Sender Receiver implicit communication
5. Data Transformation for Non-queued and queued RTE sender receiver communication
6. RTE and SchM Exclusive Area
7. Dynamic Data Type with category VSA\_LINEAR, VSA\_SQUARE, VSA\_RECTANGULAR
8. Coherency Groups for Intra ECU RTE Implicit Sender Receiver Communication
9. RTE and SchM Client Server Communication
10. ExecutableEntity Activation Reason
11. RTE Contract Phase, BSW Contract Phase and BSW Generation Phase
12. LdCom support for Inter ECU sender receiver, client server and trigger communication
13. RTE and SchM Mode Switch Communication
14. RTE and SchM External and Internal Trigger Communication
15. Inter Runnable Variables
16. Multi Partition Support for RTE and SchM
17. RTE and SchM Calibration Features
18. RTE NvBlockSwComponent and NvData Interface Support
19. RTE Hooks
20. RTE and SchM Per Instance Memory
21. SchM Sender Receiver Communication
22. Mode Switch Error Behavior
23. Generation of McSupportData information for Measurement & Calibration
24. PortInterfaceMapping (SR-SR, CS-CS, NV-SR, SR-NV, PRM-SR, PRM-PRM, NV-NV)
25. Synchronized Mode Switch
26. SwcModeManagerErrorEvent
27. SchedulePoint

#### 3.3.1.1 Post Build Features implemented in Rte include

1. Variant Handling at Assembly Connector

# Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
14 / 146

- A. RTE Intra ECU communication for Explicit NonQueued Sender Receiver APIs.
  - B. RTE Intra ECU communication for Explicit Queued Sender Receiver APIs.
  - C. RTE Intra ECU communication for Implicit Sender Receiver APIs
  - D. Client Server Synchronous Communication APIs
  - E. Client Server Asynchronous Communication APIs
  - F. Mode Switch communication APIs
2. Variant Handling at System Mapping / System Group
    - A. RTE Inter ECU communication for Explicit NonQueued Sender Receiver APIs.
    - B. RTE Inter ECU communication for Explicit Queued Sender Receiver APIs.
    - C. RTE Inter ECU communication for Implicit Sender Receiver APIs.
    - D. Client Server Synchronous Communication APIs
    - E. Client Server Asynchronous Communication APIs
3. Variant Handling at Isignal/IsignalGroup
    - A. RTE Inter ECU communication for Explicit NonQueued Sender Receiver APIs.
    - B. RTE Inter ECU communication for Explicit Queued Sender Receiver APIs.
    - C. RTE Inter ECU communication for Implicit Sender Receiver APIs.
    - D. Client Server Synchronous Communication APIs
    - E. Client Server Asynchronous Communication APIs
4. Variant Handling at Delegation Connector
    - A. RTE Inter ECU communication for Explicit NonQueued Sender Receiver APIs.
    - B. RTE Inter ECU communication for Explicit Queued Sender Receiver APIs.
    - C. RTE Inter ECU communication for Implicit Sender Receiver APIs.
    - D. Client Server Synchronous Communication APIs
    - E. Client Server Asynchronous Communication APIs
    - F. Trigger Communication
5. Variant Handling for Multi Partition communication
    - A. Explicit NonQueued Sender Receiver APIs.
    - B. Explicit Queued Sender Receiver APIs.
    - C. Implicit Sender Receiver APIs
    - D. Synchronous Client Server APIs
    - E. Asynchronous Client Server APIs
    - F. Trigger APIs
6. Processing of multiple input arxml files
  7. Multivariant support for RTE Post Build variants

## 3.3.2 Features not covered in Rte include

1. Handle Out of Range, Data Invalidation, Dynamic Data and WaitPoint for Inter-Intra Sender Receiver Communication
2. Incoherent implicit access and Inter ECU coherent implicit Sender Receiver communication
3. Minimum Start Interval for runnable entities and BSW schedulable entities
4. Optimization of generated RTE code for memory consumption or execution runtime based on RteOptimizationMode
5. FanIn and FanOut for LdCom with Sender Receiver Communication
6. FanIn and FanOut with Data Transformation with Sender Receiver Communication
7. External Replacement Policy for Handle Out of Range for inter-ECU Communication
8. Implementation Data Type of type Double
9. AliveTimeout for LdCom
10. Data Filter for category ONE EVERY\_N is not supported, and category NEVER is not handled correctly
11. Synchronized Trigger
12. Composite PortDefinedArgumentValues are not supported for Client Server communication
13. PortDefinedArgumentValue is not supported for N:1 Client Server Inter ECU communication
14. Signals with different Transformation Technology's mapped to the same dataElements are not supported for fan-out
15. MultipleInstantiation
16. Data Conversion is not supported for Variable Size Arrays and all DataPrototypes involved in intra-ECU sender-receiver communication
17. Generation using unflattened ECU extracts i.e. rootSwComposition shall contain all atomic SwComponentPrototypes and no further compositions shall be allowed
18. Dynamic Data Type with category VSA\_FULLY\_FLEXIBLE
19. RTE Compatibility mode
20. RTE Bypass support
21. Rapid Prototyping support
22. RTE\_E\_DET\_ILLEGAL\_SIGNAL\_ID, RTE\_E\_DET\_ILLEGAL\_VARIANT\_CRITERION\_VALUE DET errors are not supported
23. HandleOutOfRange as EXTERNAL REPLACEMENT is not supported for queued data prototype involved in sender receiver communication
24. RTE Vendor mode
25. Mode Switch Interface Mapping
26. Virtual Task Mapping
27. Synchronization Points for Mode Switch communication

- 28. Data Conversion for Inter ECU Sender Receiver Communication
- 29. Data Conversion with Transformer for Inter ECU Sender Receiver Communication
- 30. PRPortPrototype

### 3.3.2.1 Post Build Features not implemented in Rte include

- 1. Rte\_PBCon API
- 2. Pre-compile Variant Handling

Some features are implemented with restrictions/limitations or are implemented with a customization/deviation. They are elaborated in section 3.4 Additional/Unique/Custom features of Safe RTE.

## 3.4 Additional/Unique/Custom features of Safe RTE

### 3.4.1 Minimum AUTOSAR Schema Version Support

It is recommended to provide arxmls made with AUTOSAR schema version 4.3.1 or 4.4.0. But the SafeRTE code generator can still parse arxml files with schema minimum upto AUTOSAR version 4.0.3.

### 3.4.2 Null pointer argument detection for all RTE and SchM APIs

The input parameters passed to Safe RTE APIs are verified for NULL pointers. The RTE APIs return value as RTE\_E\_INVALID\_ARGS and SchM APIs return value as SCHM\_E\_INVALID\_ARGS in such cases. More details are specified in section 5.3 Detailed Description of RTE.

### 3.4.3 OS service failure detection and reporting

Return values of all OS services invoked in Safe RTE from APIs are checked and handled wherever possible otherwise DET error is reported if the configuration parameter RteOsErrorDetect is set to true. If the error can be reported to application, RTE APIs return value as RTE\_E\_OS\_ERROR and SchM APIs return value as SCHM\_E\_OS\_ERROR in such cases. More details are specified in section 5.3 Detailed Description of RTE.

### 3.4.4 Pre-Initialization and Post-Deinitialization run-time checks for APIs and Tasks

To protect calling/usage of RTE/SchM APIs before RTE/SchM is started and after RTE/SchM is stopped, return values RTE\_E\_UNINIT for RTE and SCHM\_E\_UNINIT for SchM are reported by APIs if called during the applicable scenarios. More details are specified in section 5.3 Detailed Description of RTE.

They shall also report to DET if RteDevErrorDetect is enabled.

Similarly, the task bodies generated by RTE shall also be protected against invocations before starting and after stopping of the RTE/SchM.



## 3.4.5 Illegal invocation detection for RTE/SchM APIs

If RteDevErrorDetect is enabled, most RTE APIs return RTE\_E\_ILLEGAL\_INVOCATION if the API is invoked from wrong context. For some cases when detection and reporting is not possible due to lack of inputs in the configuration model, a warning shall be thrown by the RTE Code generator to notify the user about it. Additionally, the SchM Sender Receiver APIs return SCHM\_E\_ILLEGAL\_INVOCATION if the API is invoked from wrong context.

More details are specified in section 7.

**CAUTION:** Illegal invocation detection is not implemented for APIs if the calling context (OsTask) of the API cannot be calculated. In such cases, SafeRTE throws a warning notifying the user about the same and the ECU integrator shall ensure correct API usage on run-time. For e.g. executable entities not scheduled by RTE/SchM but are using RTE/SchM APIs.

## 3.4.6 Invalid mode value detection in <Rte/SchM>\_Switch API

SchM\_Switch and Rte\_Switch API returns SCHM\_E\_INVALID and RTE\_E\_INVALID respectively if the input mode provided is invalid. More details are specified in section 5.3 Detailed Description of RTE.

## 3.4.7 DET reporting for incorrect sequence and NULL arguments from RTE LdCom Callbacks

Additional DET errors are reported by the RTE from the LdCom communication callbacks in case of the below scenarios when RteDevErrorDetect is enabled:

1. RTE\_E\_PARAM\_POINTER – in case of NULL arguments
2. RTE\_E\_INCORRECT\_SEQUENCE – in case of incorrect sequence of callbacks called e.g. TpCopyRxData called before TpStartOfReception.

## 3.4.8 DET Reporting for Transformer failure from Trigger APIs

DET Error RTE\_E\_XFRM\_FAILURE shall be reported from Trigger APIs in case of Inter ECU Trigger communication in case of Transformer failure when RteDevErrorDetect is enabled. This is provided as Rte\_Trigger API does not return a status value.

## 3.4.9 RTE/SchM Initialization Policy

SafeRTE code generator only considers RteInitializationStrategy of type “RTE\_INITIALIZATION\_STRATEGY\_AT\_DATA\_DECLARATION”. The RteInitializationBehavior configuration containers are not used by the SafeRTE code generator.

Additionally, in a multi-partition environment, the partition-specific data is initialized during the initialization process. During a partition restart, data needed by RTE is re-initialized by `Rte_RestartPartition_<PID>` API.

## 3.4.10 API selection for starting an OsScheduleTable

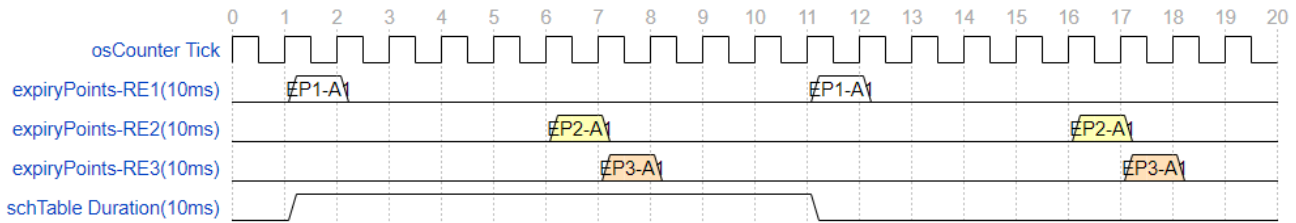
The AUTOSAR Operating System provides multiple APIs to start a schedule table mapped to RTEEvents and BswEvents which are `StartScheduleTableRel`, `StartScheduleTableAbs` and `StartScheduleTableSynchron`. It is decided by the SafeRTE code generator which API shall be used depending on the following rule:

```
IF ((OsScalabilityClass == SC2 || OsScalabilityClass == SC4) && ScheduleTable Synchronization is configured)
{
    IF (OsScheduleTableSyncStrategy == EXPLICIT)
    {
        StartScheduleTableSynchron shall be used.
    }
    ELSE
    {
        StartScheduleTableAbs shall be used.
    }
}
ELSE
{
    StartScheduleTableRel shall be used.
}
```

## 3.4.11 OsScheduleTable usage for periodic entities

As the upper multiplicity of `RteEventToTaskMapping.RteUsedOsSchTblExpiryPointRef` and `RteBswEventToTaskMapping.RteBswUsedOsSchTblExpiryPointRef` is 1, it is not possible to map multiple expiry points to a single `TimingEvent` or `BswTimingEvent`.

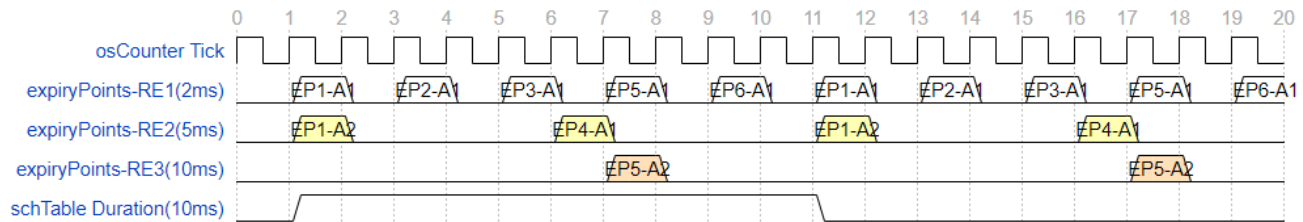
Hence with the existing metamodel, it is restricted to map multiple `TimingEvents` or `BswTimingEvents` with unequal period to the same `ScheduleTable`. This means for a `TimingEvent/BswTimingEvent` mapped to a `ScheduleTableExpiryPoint`, the period of the Event shall be equal to the product of the `OsScheduleTableDuration` and the tick duration of the counter driving the `ScheduleTable`.



**Figure 3: Supported configuration as per AUTOSAR RTE SWS**

Above constraint is enforced on the RTE ECU configuration even though the OS ScheduleTable configuration allows mapping the OsEvent-OsTask combination used for a TimingEvent/BswTimingEvent to multiple expiry points of the same scheduleTable.

Above restriction is relaxed by the SafeRTE where it allows grouping of TimingEvent(s)/BswTimingEvent(s) with unequal periods to the same schedule table. A sample scenario which may be configured using SafeRTE and OS is explained below.



**Figure 4: Supported configuration with SafeRTE**

SafeRTE expects any one of the expiry points used for the TimingEvent/BswTimingEvent to be referred from the RteEventToTaskMapping/RteBswEventToTaskMapping container created for the event.

**CAUTION:** With the above extension, SafeRTE allows generation of scenarios for which there can be potential run-time timing irregularities if user has not created sufficient expiry points. SafeRTE throws a warning for such scenarios and the ECU integrator shall ensure that the periodic scheduling of all executable entities involved in this scenario is correct on run-time.

## 3.4.12 WaitPoint handling and configuration

WaitPoints are implemented by SafeRTE with the following constraints so that deterministic behavior can be ensured on runtime.

1. It is recommended to configure not more than one WaitPoint inside a Runnable Entity.
2. The RunnableEntity owning the WaitPoint shall always be mapped to an Extended OsTask.
3. The RunnableEntity owning the WaitPoint shall always be scheduled using a TimingEvent.
4. The RunnableEntity owning the WaitPoint shall be mapped to only one TimingEvent.

5. The timeout parameter inside the WaitPoint configuration shall be mandatory and non-zero.
6. The timeout value configured for the WaitPoint shall be lesser than the period of the TimingEvent which schedules the runnableEntity owning the WaitPoint.

WaitPoints are implemented using additional Os objects referred through the below reference parameters configured inside the RteEventToTaskMapping created for the TimingEvent scheduling the runnableEntity owning the WaitPoint.

1. RteUsedWaitPointOsEventRef

- A. Reference to an OsEvent (additional to the one referred from RteUsedOsEventRef) which is used to enter the waiting state. This OsEvent shall be set by the RTE upon activation of the RTEEvent e.g.

DataSendCompletedEvent/DataReceivedEvent/AsynchronousServerCallReturnsEvent/ModeSwitchedAckEvent which acts as the trigger for the WaitPoint.

2. RteUsedWaitPointTOutOsAlarmRef

- A. Reference to an OsAlarm which is started from the blocking RTE API before it enters the WaitPoint. This OsAlarm is required for deadline monitoring the waitPoint condition i.e. to ensure that the waitPoint is exited after timeout occurs. The OsAlarm shall refer to an additional OsEvent (additional to the OsEvents referred from RteUsedOsEventRef and RteUsedWaitPointOsEventRef) in the role OsAlarmExpiryAction.OsAlarmSetEvent.OsAlarmSetEventRef which shall be set on the expiry of the OsAlarm.

The waitpoint shall be implemented using a WaitEvent call with mask created from the OsEvents:

RteUsedWaitPointOsEventRef and "RteUsedWaitPointTOutOsAlarmRef.

OsAlarmExpiryAction.OsAlarmSetEvent.OsAlarmSetEventRef".

A separate RteEventToTaskMapping shall not be needed for the RTEEvent which acts as the trigger to the WaitPoint.

### 3.4.13 Restrictions for PortInterfaceMapping

1. For VariableAndParameterInterfaceMapping, two DataPrototypes mapped together shall have same or compatible DataTypes. Compatibility of DataTypes shall be ensured by the user.
2. For ClientServerInterfaceMapping, two ClientServerOperations mapped together shall have compatible ArgumentDataPrototypes i.e. DataTypes of all Arguments on either side shall be same or compatible.
3. For ClientServerInterfaceMapping, two ClientServerOperations mapped together shall have identical ApplicationErrors as ClientServerApplicationErrorMapping is not supported.
4. ModeSwitchInterfaceMapping is not supported.
5. TriggerInterfaceMapping is not supported.

### 3.4.14 Memory mapping of RTE lower layers in multi-partition RTE

Com, LdCom and NvM are identified as lower layers of RTE which provide services to the RTE for various use-cases. Whenever SWC's are either involved in Inter ECU communication or have NvData needs in a multi-partition environment, it is required for the RTE to identify in which partitions and cores are these modules present. This information is required to generate interference-free RTE code.

The presence of Com, LdCom and NvM on a partition on each core shall be derived using creation of BswDistinguishedPartitions for the BswModuleDescription of the respective modules and mapping them to EcucPartitions.

The SafeRTE can generate code accordingly depending on the scenarios whether the SWC and the lower layer are on the same partition, different partitions, or different cores.

### 3.4.15 Memory mapping of Lifecycle API's in multi-partition RTE

In an ECU involving more than one partition, RTE shall be implemented in a way where the generated artefacts such as variables, functions etc. are intended for writing and execution from a specific partition only. In such cases, it is required for the code generator to identify partitions for all APIs to be generated so that partition boundaries are respected and freedom from interference is achieved.

For common lifecycle APIs, the RTE depends on the BswModuleDescriptions provided as the input to derive the caller partitions.

It is expected that on each core, there shall be exactly one EcucPartition that refers to a unique BswDistinguishedPartition created inside the BswModuleDescription of "EcuM" module. RTE shall generate Lifecycle APIs and supporting core-specific artefacts in a way that the initialization/starting for RTE/SchM shall be performed in the following way: -

```
switch(GetCoreID())
{
    case OS_CORE_ID_0:
    {
        Initialize caller partition-specific (mapped to EcuM) data;
        Switch to context of other partitions on Core0 using IOC to initialize partition-specific data;
        Start timing on Core0**;
```

}

```
    case OS_CORE_ID_1:
    {
        Initialize caller partition-specific (mapped to EcuM) data;
        Switch to context of other partitions on Core1 using IOC to initialize partition-specific data;
```

```
        Start timing on Core1**;  
    }  
}
```

### 3.4.16 Cross-core communication and timeout

The SafeRTE relies on IOC functionality of the AUTOSAR OS to implement cross-partition and cross-core context switches. RTE implements its own partition-specific functions which are used as `OslocReceiverPullCBs` and activated by IOC on the receiver partition.

For synchronous cross-core communication that will be handled by the SafeRTE, whenever an `locWrite` is called from the caller partition, it is expected that till the `locReceiverPullCB` is called on the other core, the caller partition shall not proceed with further processing.

For some synchronous RTE functions such as blocking RTE APIs specified by the RTE SWS, timeout functionality is already present as per AUTOSAR RTE SWS.

For the rest of the scenarios, the SafeRTE shall implement a hard-blocking (control held at one point without rescheduling) internal synchronization point in the API after the `locWrite` call to wait for the operation to be finished on the callee core.

To ensure that every such synchronization point is resolved if the operation requested on the other core fails for some reason, RTE shall automatically exit this point after a timeout occurs.

The timeout value is global for all such internal synchronization points and configured using the RTE configuration parameter *RteCrossCoreServiceTOut*.

**CAUTION:** Although the SafeRTE provides this functionality to ensure deterministic behaviour, it is recommended to configure the *RteCrossCoreServiceTOut* parameter to the lowest possible value for maximum performance. The parameter value shall always be in the multiples of the `OsCounterTick` used for the time counting. It is additionally recommended to minimize use of cross-core synchronous operations for implementation of faster parallel multi-core systems.

### 3.4.17 OsNeeds generation for multi-partition configuration

The SafeRTE code generator generates an additional `Os_loc.arxml` file to be used for OS generation under the following situations:

1. RTE uses IOC for cross-partition communication
  - A. RTE uses IOC for implementing protected shared buffers for Data and triggering writes activating `locReceiverPullCB` to switch to RTE belonging to different partitions. `Osloc` container configurations for such use cases are generated along with the required data types in the `OsNeeds.arxml`.

OR

2. RTE requires an OsSpinlock for its internal critical sections

- A. If RTE requires OsSpinlock based critical sections to protect its own internal buffers and data from concurrent read-write accesses from different cores, it will use one OsSpinlock for this and generate the OsSpinlock configuration as part of the OsNeeds.xml.

**CAUTION:** The OsSpinlock configuration generation is disabled if the command line option “genlocOnly” is provided. In such cases, the ECU Integrator shall manually configure an OsSpinlock with the shortName “RteInternalSpinlock” and then generate OS to ensure successful compilation and build.

Apart from the above items, the SafeRTE code generator does not add any objects to the existing Os configuration. The command line option “genlocOnly” shall decide the content of the OsNeeds.xml as per the below table: -

**Table 2: genlocOnly impact on OsNeeds.xml**

Content	Generated with genlocOnly	Generated w/o genlocOnly
Osloc containers	Yes	Yes
Existing Os configuration	No	Yes
Existing EcuC module configuration (for Partition information)	No	Yes
OsSpinlock: RteInternalSpinlock	No	Yes
ImplementationDataTypes referred from Osloc configuration	Yes	Yes

## 3.4.18 McSupportData generation in RTE BSWMD file

The RTE BasicSoftwareModuleDescription (BSWMD) file is generated along with the source and header files. It contains version information and McSupportData information which can be used as an interchange file for preparation of an A2L file for measurement and calibration purposes.

The generation of McSupportData information is dependent on the RTE ECU Configuration as well as the individual DataPrototype properties. FlatMap configuration is required and needs to be referenced from the rootSwComposition for generation of McSupportData information.

Apart from what is specified in RTE SWS, following restrictions apply to the McSupportData information

generated by the SafeRTE:

1. Measurement Support is not supported for ModeDeclarationGroupPrototypes for all the mode machines.
2. Measurement Support is not supported for all ArgumentDataPrototypes used in client server communication.
3. Measurement support shall not be provided for DataPrototypes of Non-Queued Explicit and involved in an Intra ECU N:1 sender receiver communication with senders and receivers in different partitions.
4. In Inter ECU Sender Receiver Transmission, RTE shall provide measurement support for applicable DataPrototypes each that is mapped to same or different ComSignals/LdComIPdu's/ComSignalGroups.
5. In Inter ECU Sender Receiver Reception, RTE shall not be able to provide measurement support for DataPrototypes which have to be de-serialized before providing to the receiving SWC from COM/LdCom.
6. In Inter ECU Sender Receiver Reception, RTE shall not be able to provide measurement support for DataPrototypes which are not read from the COM Callbacks for i.e. 1: 1 & 1: N with SWC(s) in the same partition as communication layer module.
7. Measurement support is not supported for queue of sender receiver communication.

The rationale behind above restrictions is the implementation design does not allocate clear allocation of memory for DataPrototype's in such scenarios.

### 3.4.19 RTE buffer handling when Transformer is configured

1. SafeRTE always passes the base address of the buffer to COMBased Transformer.
2. SafeRTE always leaves a free space in front of the buffer for SOMEIP Transformer to insert its header information.
3. The E2E Header Size Macro is always generated as 0 when ComBased and E2E Transformer chain is involved. An error is also reported by SafeRTE when BufferProperties.headerLength is not configured as 0.

### 3.4.20 Partition Restart

1. Partitions of the ECU can be terminated and restarted if a partition has capability of Restarting. The Restart Task of that Partition shall have the highest OsTaskPriority on the core to ensure the safe Restarting Operation.



### 3.4.21 Postbuild generation phase not existing

1. The RTE Postbuild data set generation is embedded in the Safe RTE Code Generation. Postbuild data sets are generated in RTE Code Generator based on input command line option (-postbuild), Postbuild data sets are not generated if this option is not present in command line arguments..

### 3.4.22 VFB Tracing clients ordering

In case of multiple clients for one trace event, the individual action or start trace functions are called in the following order

1. The trace functions with client prefix
2. The trace functions with client prefix in alphabetically ascending order of the shortName of the RteVfbTraceClientPrefix

and the individual stop or return trace functions are called in the following order

1. The trace functions with client prefix in alphabetically descending order of the shortName of the RteVfbTraceClientPrefix
2. The trace functions without client prefix

### 3.4.23 Schedule Point

Schedule Point introduce an explicit Schedule API calling after execution of the runnable within generated NonPreemptable Task. The NonPreemptable Task will be suspended if a higher priority task is ready and Schedule API is called in the NonPreemptable Task.

If RteOsSchedulePoint is not set or set to NONE, Schedule Point is omitted. If RteOsSchedulePoint is set to CONDITIONAL, Schedule API is called after execution of the runnable entity. If RteOsSchedulePoint is set to UNCONDITIONAL, Schedule API is inserted after runnable entity's position.

**CAUTION:** Due to generated code design, UNCONDITIONAL Schedule Point may operate in a different way in some RteEvents that operate as a queue.

1. Synchronized Mode Switch Event: Unconditional Schedule Point is executed if the event queue is not empty.
2. Operation Invoked Event: Unconditional Schedule Point is executed after all Operation Invoked Events inserted into the queue are performed.
3. Transformer Hard Error Event: Unconditional Schedule Point is executed after Transformer Hard Error Events inserted into the queue are performed.

Therefore, when specifying UNCONDITIONAL Schedule Point for Synchronized Mode Switch Event, Operation Invoked Event, or Transformer Hard Error Event, you should check the generated code to check whether the

Unconditional Schedule Point is performed as intended.

## 3.5 Pre-Conditions

This section provides information list of Pre-Conditions for Rte code generator. The following are the minimum hardware and software requirements for execution of Rte generator tool:

1. Intel(R) Core (TM)i3 Processor – 1.7GHz
2. 4GB RAM
3. 32-bit OS or 64-bit OS
4. Minimum Hard Disk Space 1GB
5. Operating system: Windows 8 or Windows 10
6. Sun Java Run Time Environment (JRE) Version 1.8

The below given pre-conditions also need to be taken care:

1. ECU extract provided to Rte should always be in flattened. There should be only one top level composition present in the ECU Extract.
2. Schema validated input arxml files need to be provided to Rte code generator tool.
3. Valid input path and output path to be provided when using command line based Rte code generator tool
4. Mandatory parameters of dependent Bsw Modules should be provided to RTE. Dependent BSW module includes Os, Com, LdCom, DET, NvM and Transformer Modules (if configured in RTE).
5. Input arxml file should have the configuration parameters like variation points, postbuild criterion values, postbuild criterion value sets, and predefined variants in case of applying -postbuild option.

## 4. Configuration Guide

Without further comments, configuration parameters covered in this chapter are based on AUTOSAR specs.

### 4.1 RteGeneration

This container holds the parameters for the configuration of the RTE Generation.

Parameter Name	Value	Category
RteCalibrationSupport <sup>1)</sup>	NONE / DOUBLE_POINTERED / INITIALIZED_RAM / SINGLE_POINTERED	C
RteDevErrorDetect <sup>2)</sup>	true / false	C
RteDevErrorDetectUnInit <sup>3)</sup>	true / false	C
RteVfbTraceClientPrefix <sup>4)</sup>	<String>	C
RteVfbTraceEnabled <sup>5)</sup>	true / false	C
RteVfbTraceFunction <sup>6)</sup>	<Function Name>	C

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
27 / 146

Parameter Name	Value	Category
RteGenerationMode	COMPATIBILITY_MODE	N
RteMeasurementSupport <sup>7)</sup>	true / false	C
RteOptimizationMode	RUNTIME	N
RteValueRangeCheckEnabled <sup>8)</sup>	true / false	C
RteCodeVendorId	76	F
RteToolChainSignificantCharacters	<Integer>	N
RteBypassSupport	NONE	N
RteInExclusiveAreaCheckEnabled <sup>9)</sup>	true / false	C
RteRxAPIComDecoupling <sup>10)</sup> vendor-specific)	true / false	C
RteOsErrorDetect <sup>11)</sup> vector-specific)	true / false	C
RteCrossCoreServiceTOut <sup>12)</sup> vendor-specific)	<Float>	C

- 1) **RteCalibrationSupport**  
Set Calibration Method.
- 2) **RteDevErrorDetect**  
Turns on or off DET functionality for Rte module.
- 3) **RteDevErrorDetectUnInit**  
Turns on or off checking Init of Rte when DET functionality is active.
- 4) **RteVfbTraceClientPrefix**  
Defines an additional prefix for all VFB trace functions to be generated.
- 5) **RteVfbTraceEnabled**  
Turns on or off Vfb Trace functionality for Rte module.
- 6) **RteVfbTraceFunction**  
Function name of Vfb Trace.
- 7) **RteMeasurementSupport**  
Turns on or off support for measurement for generated RTE code. This option shall influence complete RTE code at once.
- 8) **RteValueRangeCheckEnabled**  
Turns on or off RangeCheck(HandleOutOfRange) functionality for Rte module.
- 9) **RteInExclusiveAreaCheckEnabled**  
Turns on or off the check for RTE\_E\_IN\_EXCLUSIVE\_AREA (for blocking APIs).
- 10) **RteRxAPIComDecoupling**  
If set to true, Com APIs' shall be called from call backs. If not configured or set to false, Com APIs' shall be called from Rte API.
- 11) **RteOsErrorDetect**  
If set to true, additional codes are generated to check and detect behavior and error of OS.
- 12) **RteCrossCoreServiceTOut**  
Sets timeout value of Cross-core services in seconds.

## 4.2 RteSwComponentType

Configure SwComponentType for Rte code generation.

Parameter Name	Value	Category
RteComponentTypeRef <sup>1)</sup>	<ref: SwComponentType>	C
RteImplementationRef <sup>2)</sup>	<ref: SwImplementation>	C
RteBypassSupportEnabled	true / false	N

- 1) RteComponentTypeRef  
Selects SwComponentType to be configured.
- 2) RteImplementationRef  
The Implementation which shall be assigned to the SwComponentType

### 4.2.1 RteComponentTypeCalibration

Configures Calibration functionality.

Parameter Name	Value	Category
RteCalibrationSupportEnabled <sup>1)</sup>	true / false	C
RteCalibrationSwAddrMethodRef	<ref: SwAddrMethod>	N

- 1) RteCalibrationSupportEnabled  
Set to true when Calibration functionality is being used (related Software Component is ParamterComponentType or Shared/PerInstance Parameter is set).

## 4.3 RteSwComponentInstance

This container should be added for each SW-C Prototype for code generation.

Parameter Name	Value	Category
RteSoftwareComponentInstanceRef <sup>1)</sup>	<ref: SwComponentPrototype>	C

- 1) RteSoftwareComponentInstanceRef  
Selects SW-C Prototype for code generation.

### 4.3.1 RteEventToTaskMapping

This container should be added for each RTEEvent which is mapped to OsTask and runs Runnable in SW-C.

Parameter Name	Value	Category
RteEventRef <sup>1)</sup>	<ref: RTEEvent>	C

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
29 / 146

Parameter Name	Value	Category
RteMappedToTaskRef <sup>2)</sup>	<ref: OsTask>	C
RtePositionInTask <sup>3)</sup>	<integer>	C
RteUsedOsAlarmRef <sup>4)</sup>	<ref: OsAlarm>	C
RteUsedOsEventRef <sup>5)</sup>	<ref: OsEvent>	C
RteVirtuallyMappedToTaskRef	<ref: OsTask>	N
RteActivationOffset <sup>6)</sup>	<Float>	C
RteImmediateRestart	false	N
RteOsSchedulePoint <sup>7)</sup>	NONE / CONDITIONAL / UNCONDITIONAL	C
RteUsedOsSchTblExpiryPointRef <sup>8)</sup>	<ref: OsScheduleTableExpiryPoint>	C
RteEventPredecessorSyncPointRef	<ref: RteSyncPoint>	N
RteEventSuccessorSyncPointRef	<ref: RteSyncPoint>	N
RteUsedInitFnc <sup>9)</sup>	<ref: RteInitializationRunnableBatch>	C
RteUsedWaitPointTOutOsAlarmRef <sup>10)</sup> vendor-specific)	<ref: OsAlarm>	C
RteUsedWaitPointOsEventRef <sup>11)</sup> vendor-specific)	<ref: OsEvent>	C

- 1) **RteEventRef**  
Selects RTEEvent to be configured.
- 2) **RteMappedToTaskRef**  
Select OsTask with Runnable that RTEEvent runs.
- 3) **RtePositionInTask**  
Sets position of related Runnable inside OsTask when one OsTask contains more than one Runnables.  
※ Position In Task shall not be duplicate among same OsTask.
- 4) **RteUsedOsAlarmRef**  
Selects OsAlarm which should activate OsTask selected on RteMappedToTaskRef, if RTEEvent is Timing Event. (Therefore OsAlarm should select ActivateTask as OsAlarmAction and the referenced OsTask should match and it should be identical to the one selected on RteMappedToTaskRef.)  
※ Unapplicable when not Timing Event
- 5) **RteUsedOsEventRef**  
If an OsEvent is used to activate the OsTask this RteEvent is mapped to it shall be referenced here.
- 6) **RteActivationOffset**  
Activation offset in seconds.
- 7) **RteOsSchedulePoint**  
If a schedule point is set here, it will be introduced 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. The absence of this parameter is interpreted as "NONE". 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.
- 8) **RteUsedOsSchTblExpiryPointRef**  
If an OsScheduleTableExpiryPoint is used to activate the OsTask this RteEvent is mapped to it shall be

referenced here.

## 9) RteUsedInitFnc

The RunnableEntity is executed during initialization in the context of the Rte\_Init\_<InitContainer> function.

## 10) RteUsedWaitPointTOutOsAlarmRef

Refer to section 4.4.12 WaitPoint handling and configuration.

## 11) RteUsedWaitPointOsEventRef

Refer to section 4.4.12 WaitPoint handling and configuration.

### 4.3.2 RteExclusiveAreaImplementation

This container should be added for each ExclusiveArea when ExclusiveArea is used inside Runnable of SW-C.

Parameter Name	Value	Category
RteExclusiveAreaRef <sup>1)</sup>	<ref: ExclusiveArea>	C
RteExclusiveAreaImplMechanism <sup>2)</sup>	NONE / ALL_INTERRUPT_BLOCKING / OS_INTERRUPT_BLOCKING / OS_RESOURCE	C
RteExclusiveAreaImplKind <sup>3)</sup> vendor-specific)	RTE_MACRO / RTE_FUNCTION	C
RteExclusiveAreaOsResourceRef <sup>4)</sup>	<ref: OsResource>	C

## 1) RteExclusiveAreaRef

Select ExclusiveArea which is configured by Software Component.

## 2) RteExclusiveAreaImplMechanism

Sets mechanism of referenced ExclusiveArea.

- A. ALL\_INTERRUPT\_BLOCKING: This value requests enabling and disabling of all Interrupts and is based on the Interrupt blocking strategy.
- B. OS\_INTERRUPT\_BLOCKING: This value requests enabling and disabling of Os Interrupts and is based on the Interrupt blocking strategy.
- C. OS\_RESOURCE: This value requests to apply the Usage of OS resources mechanism.
- D. NONE: RTE generator shall not apply any mechanisms for data consistency. Data consistency will be ensured by methods outside of RTE implementation control.

※ Refer to OS UM to find details about SuspendAllInterrupts, ResumeAllInterrupts, SuspendOsInterrupts, ResumeOsInterrupts and ReleaseResource API

## 3) RteExclusiveAreaImplKind

Sets how ExclusiveArea should be implemented in generated code.

## 4) RteExclusiveAreaOsResourceRef

Select OsResource to be used in ExclusiveArea when RteExclusiveAreaImplMechanism is set to OS\_RESOURCE.

## 4.3.3 RteExternalTriggerConfig

This container should be added to related RteSwComponentIntance if External Trigger Communication between SW-Cs is implemented as queued.

Parameter Name	Value	Category
RteSwcTriggerSourceRef <sup>1)</sup>	<ref: Trigger>	C
RteTriggerSourceQueueLength <sup>2)</sup>	<Integer>	C

- 1) RteSwcTriggerSourceRef  
Select TriggerInterface which is implemented as queue.
- 2) RteTriggerSourceQueueLength  
Length of queue of TriggerInterface referenced by RteSwcTriggerSourceRef.

## 4.3.4 RteInternalTriggerConfig

This container should be added to RteSwComponentIntance related to SW-C with TriggerInstance configured, when Internal Trigger Communication is used between Runnables inside SW-C, and is implemented as queued.

Parameter Name	Value	Category
RteSwcTriggerSourceRef <sup>1)</sup>	<ref: InternalTriggeringPoint>	C
RteTriggerSourceQueueLength <sup>2)</sup>	<Integer>	C

- 1) RteSwcTriggerSourceRef  
Select Trigger Instance which is implemented as queued.
- 2) RteTriggerSourceQueueLength  
Length of queue of Trigger Instance referenced at RteSwcTriggerSourceRef.

## 4.3.5 RteNvRamAllocation

Parameter Name	Value	Category
RteNvmRamBlockLocationSymbol <sup>1)</sup>	<Linker Symbol>	C
RteNvmRomBlockLocationSymbol <sup>2)</sup>	<Linker Symbol>	C
RteSwNvRamMappingRef <sup>3)</sup>	<ref: SwcServiceDependency>	C
RteSwNvBlockDescriptorRef <sup>4)</sup>	<ref: NvBlockDescriptor>	C
RteNvmBlockRef <sup>5)</sup>	<ref: EcucLinkerSymbolDef>	C

- 1) RteNvmRamBlockLocationSymbol  
This is the name of the linker object name where the NVRam Block will be mirrored by the Nvm. This symbol will be resolved into the parameter "NvmRamBlockDataAddress" from the "NvmBlockDescriptor".
- 2) RteNvmRomBlockLocationSymbol

This is the name of the linker object name where the NVRom Block will be accessed by the Nvm. This symbol will be resolved into the parameter "NvmRomBlockDataAddress" from the "NvmBlockDescriptor".

### 3) RteSwNvRamMappingRef

Reference to the SwServiceDependency which is used to specify the NvBlockNeeds.

### 4) RteSwNvBlockDescriptorRef

Reference to the NvBlockDescriptor in case the RTE needs to call the NvM directly (e.g. for the supportDirtyFlag feature, storeCyclic feature, server invocation for NV data management or mode switch based invocation NvM services).

### 5) RteNvmBlockRef

Reference to the used NvM block for storage of the NVRAMMapping information.

## 4.4 RteBswModuleInstance

This container should be added for each BSW modules.

- ※ This contains configuration of behavior of AUTOSAR BSW modules.
- ※ Since this information is fixed when HAE delivers platform including SWP SRS and required information of each BSW modules, it is not possible to be altered when designing and configuring Application.
- ※ Except under discussion with HAE when developing CCD Layer manually

Parameter Name	Value	Category
RteBswImplementationRef <sup>1)</sup>	<ref: BswImplementation>	C
RteBswModuleConfigurationRef <sup>2)</sup>	<ref: EcucModuleConfigurationValues>	C

### 1) RteBswImplementationRef

Set reference to BswImplementation.

### 2) RteBswModuleConfigurationRef

Set reference to BswModuleConfiguration.

### 4.4.1 RteBswEventToTaskMapping

This container maps BswEvent to OsTask in order for BswMd to run Entity. This container should be added for each BSWEvent.

Parameter Name	Value	Category
RteBswEventRef <sup>1)</sup>	<ref: BswEvent>	C
RteBswMappedToTaskRef <sup>2)</sup>	<ref: OsTask>	C
RteBswPositionInTask <sup>3)</sup>	<Integer>	C
RteBswUsedOsAlarmRef <sup>4)</sup>	<ref: OsAlarm>	C
RteBswUsedOsEventRef <sup>5)</sup>	<ref: OsEvent>	C



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
33 / 146

Parameter Name	Value	Category
RteBswActivationOffset <sup>6)</sup>	<Float>	C
RteBswImmediateRestart	false	N
RteOsSchedulePoint <sup>7)</sup>	NONE / CONDITIONAL / UNCONDITIONAL	C
RteBswUsedOsSchTblExpiryPointRef <sup>8)</sup>	<ref: OsScheduleTableExpiryPoint>	C
RteBswServerQueueLength <sup>9)</sup>	<Integer>	C
RteBswEventPredecessorSyncPointRef	<ref: RteSyncPoint>	N
RteBswEventSuccessorSyncPointRef	<ref: RteSyncPoint>	N
RteBswUsedWaitPointTOutOsAlarmRef <sup>10)</sup> vendor-specific)	<ref: OsAlarm>	C
RteBswUsedWaitPointOsEventRef <sup>11)</sup> vendor-specific)	<ref: OsEvent>	C

- 1) **RteBswEventRef**  
Sets BswEvent on which the Task should be mapped.
- 2) **RteBswMappedToTaskRef**  
Sets Task which includes Entity which the BswEvent calls.
- 3) **RteBswPositionInTask**  
Sets position of related Entity which the BswEvent calls inside the mapped (value starts from 0).  
※ Position In Task shall not be duplicate among same Task.
- 4) **RteBswUsedOsAlarmRef**  
If the BswEvent is Timing Event, sets the Alarm which activates the Task.  
※ Unapplicable when not Timing Event
- 5) **RteBswUsedOsEventRef**  
Sets the Event which activates the Task if the Task mapped to RteEvent is Extended Task.
- 6) **RteBswActivationOffset**  
Activation offset in seconds.
- 7) **RteOsSchedulePoint**  
RteOsSchedulePoint referenced here will be introduced 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. The absence of this parameter is interpreted as "NONE". 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.
- 8) **RteBswUsedOsSchTblExpiryPointRef**  
If an OsScheduleTableExpiryPoint is used to activate the OsTask this BswEvent is mapped to it shall be referenced here.
- 9) **RteBswServerQueueLength**  
Specifies the length of the queue for the server call serialization.
- 10) **RteBswUsedWaitPointTOutOsAlarmRef**  
Refer to section 4.4.12 WaitPoint handling and configuration.

## 11) RteBswUsedWaitPointOsEventRef

Refer to section 4.4.12 WaitPoint handling and configuration.

### 4.4.2 RteBswExclusiveAreaImpl

This container should be added for each ExclusiveArea when Entity code of BSW is implemented.

Parameter Name	Value	Category
RteBswExclusiveAreaRef <sup>1)</sup>	<ref: ExclusiveArea>	C
RteExclusiveAreaImplMechanism <sup>2)</sup>	NONE / ALL_INTERRUPT_BLOCKING / OS_INTERRUPT_BLOCKING / OS_RESOURCE / OS_SPINLOCK	C
RteBswExclusiveAreaOsResourceRef <sup>3)</sup>	<ref: OsResource>	C
RteBswExclusiveAreaOsSpinlockRef <sup>4)</sup>	<ref: OsSpinlock>	C
RteBswExclusiveAreaImplKind <sup>5)</sup> vendor-specific	RTE_MACRO / RTE_FUNCTION	C

#### 1) RteBswExclusiveAreaRef

Sets reference to ExclusiveArea which is set by BSW Module Description.

#### 2) RteExclusiveAreaImplMechanism

Sets mechanism of related ExclusiveArea.

- A. ALL\_INTERRUPT\_BLOCKING: This value requests enabling and disabling of all Interrupts and is based on the Interrupt blocking strategy.
- B. OS\_INTERRUPT\_BLOCKING: This value requests enabling and disabling of Os Interrupts and is based on the Interrupt blocking strategy.
- C. OS\_RESOURCE: This value requests to apply the Usage of OS resources mechanism.
- D. OS\_SPINLOCK: This value is used to co-ordinate concurrent access by TASKs/ISR2s on different cores to a shared resource.
- E. NONE: RTE generator shall not apply any mechanisms for data consistency. Data consistency will be ensured by methods outside of RTE implementation control.

※ Refer to OS UM to find details about SuspendAllInterrupts, ResumeAllInterrupts, SuspendOsInterrupts, ResumeOsInterrupts and ReleaseResource API

#### 3) RteBswExclusiveAreaOsResourceRef

Select OsResource to be used in ExclusiveArea when RteExclusiveAreaImplMechanism is set to OS\_RESOURCE.

#### 4) RteBswExclusiveAreaOsSpinlockRef

Optional reference to an OsSpinlock in case RteExclusiveAreaImplMechanism is configured to OS\_SPINLOCK for this ExclusiveArea.

#### 5) RteBswExclusiveAreaImplKind

Sets how ExclusiveArea should be implemented in generated code.

### 4.4.3 RteBswRequiredClientServerConnection

Parameter Name	Value	Category
RteBswProvidedClientServerEntryModInstRef <sup>1)</sup>	<ref: RteBswModuleInstance>	C
RteBswProvidedClientServerEntryRef <sup>2)</sup>	<ref: BswModuleClientServerEntry>	C
RteBswRequiredClientServerEntryRef <sup>3)</sup>	<ref: BswModuleClientServerEntry>	C

- 1) RteBswProvidedClientServerEntryModInstRef  
Reference to the RteBswModuleInstance configuration container which identifies the instance of the BSW Module. Used with the RteBswProvidedClientServerEntryRef to unambiguously identify the BswModuleClientServerEntry instance.
- 2) RteBswProvidedClientServerEntryRef  
Reference the providedClientServerEntry for this connection.
- 3) RteBswRequiredClientServerEntryRef  
Reference the requiredClientServerEntry for this connection.

## 4.4.4 RteBswRequiredSenderReceiverConnection

Parameter Name	Value	Category
RteBswProvidedDataModInstRef <sup>1)</sup>	<ref: RteBswModuleInstance>	C
RteBswProvidedVariableDataPrototypeRef <sup>2)</sup>	<ref: BswModuleClientServerEntry>	C
RteBswRequiredVariableDataPrototypeRef <sup>3)</sup>	<ref: BswModuleClientServerEntry>	C

- 1) RteBswProvidedDataModInstRef  
Reference to the RteBswModuleInstance configuration container which identifies the instance of the BSW Module. Used with the RteBswProvidedVariableDataPrototypeRef to unambiguously identify the VariableDataPrototype instance.
- 2) RteBswProvidedVariableDataPrototypeRef  
Reference the providedData for this connection.
- 3) RteBswRequiredVariableDataPrototypeRef  
Reference the requiredData for this connection.

## 4.4.5 RteBswRequiredModeGroupConnection

This container should be added when Mode Switch communication on BSW Module Description level is used.

Parameter Name	Value	Category
RteBswProvidedModeGroupRef <sup>1)</sup>	<ref: RteBswModuleInstance>	C
RteBswRequiredModeGroupRef <sup>2)</sup>	<ref: ModeDeclarationGroupPrototype>	C
RteBswProvidedModeGrpModInstRef <sup>3)</sup>	<ref: ModeDeclarationGroupPrototype>	C

Parameter Name	Value	Category
RteModeDeclarationMappingSetRef	<ref: ModeDeclarationMappingSet>	N

- 1) **RteBswProvidedModeGroupRef**  
Sets providedModeGroupPrototype which is connected to BswModuleEntity with role of manager in Mode Switch communication between BSW.  
※ Role of PPort in Mode Switch communication between Application SWC
- 2) **RteBswRequiredModeGroupRef**  
Sets requiredModeGroupPrototype which is connected to BswModuleEntity with role of user in Mode Switch communication between BSW.  
※ Role of RPort in Mode Switch communication between Application SWC
- 3) **RteBswProvidedModeGrpModInstRef**  
Sets ModeGroupPrototype instance which is connected by above RteBswProvidedModeGroupRef and RteBswRequiredModeGroupRef.

#### 4.4.6 RteBswRequiredTriggerConnection

This container should be added to RteBswComponentIntance related to Release Trigger BSW Module Description when Trigger Communication on BSW Module Description level is used.

Parameter Name	Value	Category
RteBswReleasedTriggerRef <sup>1)</sup>	<ref: Trigger>	C
RteBswRequiredTriggerRef <sup>2)</sup>	<ref: Trigger>	C
RteBswReleasedTriggerModInstRef <sup>3)</sup>	<ref: RteBswModuleInstance>	C

- 1) **RteBswReleasedTriggerRef**  
Set to Released Trigger Instance.
- 2) **RteBswRequiredTriggerRef**  
Set to Required Trigger Instance.
- 3) **RteBswReleasedTriggerModInstRef**  
Set to RteBswComponentIntance referenced by BSW Module Description with Released Trigger.

#### 4.4.7 RteBswExternalTriggerConfig

This container should be added to RteBswComponentIntance related to Release Trigger BSW Module Description when queued External Trigger Communication is used between BSW Module Description.

Parameter Name	Value	Category
RteBswTriggerSourceRef <sup>1)</sup>	<ref: Trigger>	C
RteBswTriggerSourceQueueLength <sup>2)</sup>	<Integer>	C

- 1) RteBswTriggerSourceRef  
Sets Release Trigger Instance which is queued.
- 2) RteBswTriggerSourceQueueLength  
Sets length of queue of Release Trigger Instance.

#### 4.4.8 RteBswInternalTriggerConfig

This container should be added to RteBswComponentInstance related to BSW Module Description with trigger set when Internal Trigger Communication is used between Entities inside BSW Module Description.

Parameter Name	Value	Category
RteBswTriggerSourceRef <sup>1)</sup>	<ref: BswInternalTriggeringPoint>	C
RteBswTriggerSourceQueueLength <sup>2)</sup>	<Integer>	C

- 1) RteBswTriggerSourceRef  
Sets Release Trigger Instance which is queued.
- 2) RteBswTriggerSourceQueueLength  
Sets length of queue of Release Trigger Instance.

### 4.5 RteOsInteraction

Interaction of the Rte with the Os.

#### 4.5.1 RteSyncPoint

The RteSyncPoint is necessary to provide an cross core synchronization in case of RteEvents triggered by the same event source but mapped to tasks belonging to different partitions on different cores.

The synchronization point must be reached by all referencing RteEvents before the execution in all related tasks is continued.

In case of Rte(Bsw)EventSuccessorSyncPointRef the ExecutableEntity activated by the mapped event is executed before the synchronization point is entered.

In case of Rte(Bsw)EventPredecessorSyncPointRef the ExecutableEntity activated by the mapped event is executed after the synchronization point is passed.

This container has no parameters or included containers.

#### 4.5.2 RteUsedOsActivation

Sets offset to activate OsTask mapped to Timing Event (in section 5.3.1 or 5.4.1) in specific time period. This

container should be added for each OsTask which needs offset to be configured.

Parameter Name	Value	Category
RteActivationType <sup>1)</sup> vendor-specific)	ABSOLUTE / RELATIVE / SYNCHRON	C
RteActivationOsAlarmRef <sup>2)</sup>	<ref: OsAlarm>	C
RteActivationOsTaskRef <sup>3)</sup>	<ref: OsTask>	C
RteExpectedActivationOffset <sup>4)</sup>	<Float>	C
RteExpectedTickDuration <sup>5)</sup>	<Float>	C
RteActivationOsSchTblRef <sup>6)</sup>	<ref: OsScheduleTable>	C

- 1) **RteActivationType**  
Enumeration Parameter which gives an option to RTE generator to generate specific ScheduleTable or Alarm function calls.
- 2) **RteActivationOsAlarmRef**  
Sets OsAlarm which is mapped to Timing Event.
- 3) **RteActivationOsTaskRef**  
Sets OsTask which is mapped to Timing Event.
- 4) **RteExpectedActivationOffset**  
Sets offset value in seconds.
- 5) **RteExpectedTickDuration**  
Activation offset in seconds. Important: This is a requirement from the Rte towards the Os/Mcu setup. The Rte Generator shall assume this activation offset to be fulfilled.
- 6) **RteActivationOsSchTblRef**  
Reference to an OsScheduleTable.

## 4.5.3 RteModeToScheduleTableMapping

Parameter Name	Value	Category
RteModeSchtblMapModeDeclarationRef <sup>1)</sup>	<ref: ModeDeclaration>	C
RteModeScheduleTableRef <sup>2)</sup>	<ref: OsScheduleTable>	C

- 1) **RteModeSchtblMapModeDeclarationRef**  
Reference to the ModeDeclarations.
- 2) **RteModeScheduleTableRef**  
Reference to the OsScheduleTable which shall be active in the specified RteModeSchtblMapModeDeclarationRefs.

### 4.5.3.1 RteModeSchtblMapBsw

Parameter Name	Value	Category
RteModeSchtblMapBswProvidedModeGroupRef <sup>1)</sup>	<ref: ModeDeclarationGroupPrototype>	C
RteModeSchtblMapBswInstanceRef <sup>2)</sup>	<ref: RteBswModuleInstance>	C

- 1) RteModeSchtblMapBswProvidedModeGroupRef  
Reference to an instance of a ModeDeclarationGroupPrototype of a Bsw-Module.
- 2) RteModeSchtblMapBswInstanceRef  
Reference to an instance specification of a Bsw-Module.

## 4.5.3.2 RteModeSchtblMapSwc

Parameter Name	Value	Category
RteModeSchtblMapSwcPortRef <sup>1)</sup>	<ref: AbstractProvidedPortPrototype>	C
RteModeSchtblMapSwcInstanceRef <sup>2)</sup>	<ref: RteSwComponentInstance>	C

- 1) RteModeSchtblMapSwcPortRef  
Reference to the PPortPrototype of a SwComponentPrototype.
- 2) RteModeSchtblMapSwcInstanceRef  
Reference to an instance specification of a SwComponentPrototype.

## 4.6 RteBswGeneral

Parameter Name	Value	Category
RteSchMVersionInfoApi <sup>1)</sup>	true / false	C

- 1) RteSchMVersionInfoApi  
Enables the generation of the SchM\_GetVersionInfo() API.

## 4.7 CommonPublishedInformation

Parameter Name	Value	Category
ArReleaseVersion	4.4.0	F
SwVersion	1.8.0	F
VendorApiInfix	<String>	N

## 4.8 RteImplicitCommunication

Parameter Name	Value	Category
RteCoherentAccess	true	F
RteImmediateBufferUpdate	<Boolean>	N
RteVariableReadAccessRef <sup>1)</sup>	<ref: VariableAccess>	C
RteVariableWriteAccessRef <sup>2)</sup>	<ref: VariableAccess>	C

Parameter Name	Value	Category
RteSoftwareComponentInstanceRef <sup>3)</sup>	<ref: SwComponentPrototype>	C

- 1) RteVariableReadAccessRef  
Reference to the VariableAccess in the dataReadAccess role.
- 2) RteVariableWriteAccessRef  
Reference to the VariableAccess in the dataWriteAccess role.
- 3) RteSoftwareComponentInstanceRef  
Reference to a SwComponentPrototype. This denotes the instances of the VariableAccess belonging to the RteImplicitCommunication.

## 4.9 RtePostBuildVariantConfiguration

Parameter Name	Value	Category
RtePostBuildUsedPredefinedVariant <sup>1)</sup>	<ref: PredefinedVariant>	C

- 1) RtePostBuildUsedPredefinedVariant  
Reference to the PredefinedVariant element which defines the values for PostBuildVariationCriterion elements. The shortName of the referenced PredefinedVariant defines the name of the RtePostBuildVariant.

## 4.10 RteInitializationBehavior

Parameter Name	Value	Category
RteSectionInitializationPolicy	<Enumeration>	N
RteInitializationStrategy	RTE_INITIALIZATION_STRATEGY_AT_DATA_DECLARATION	F

## 4.11 RteInitializationRunnableBatch

This container corresponds to an Rte\_Init\_<shortName of this container> function invoking the mapped RunnableEntities. This container has no parameters or included containers.

# 5. Application Programming Interface (API)

This section describes the error codes returned by the RTE APIs, SCHM APIs and Callbacks.

## 5.1 Type Definitions

### 5.1.1 Predefined Error Codes

This section describes the error codes returned by the RTE APIs, SCHM APIs and Callbacks.



**Table 3: RTE Error Codes**

Error Code	Value	Description
RTE_E_OK	0	No error occurred.
RTE_E_INVALID	1	Invalidation in Sender Receiver Communication or Invalid mode is passed as an input to Rte_Switch API in Mode Switch.
RTE_E_LOST_DATA	64	Overlaid error, indicates that some incoming data has been lost due to an overflow.
RTE_E_MAX_AGE_EXCEEDED	64	Overlaid error, indicates that the available data has exceeded the aliveTimeout limit.
RTE_E_COM_STOPPED	128	Com Service not available or an IPDU group was disabled.
RTE_E_TIMEOUT	129	Expiry of a local timeout for a blocking call API or non-blocking Rte_SwitchAck and Rte_Result APIs for Mode Switch and Client Server respectively or Client Server APIs is called before RTE initialization or RTE finalization or cross core timeout occurs during multipartition scenario.
RTE_E_LIMIT	130	An internal RTE limit exceeded.
RTE_E_NO_DATA	131	No data returned.
RTE_E_TRANSMIT_ACK	132	Transmission acknowledgement.
RTE_E_NEVER_RECEIVED	133	No data received since system start or partition restart.
RTE_E_UNCONNECTED	134	Unconnected port.
RTE_E_OUT_OF_RANGE	137	Received data is out of range.
RTE_E_HARD_TRANSFORMER_ERROR	138	Hard Transformer error.
RTE_E_SOFT_TRANSFORMER_ERROR	140	Soft Transformer error.
RTE_E_COM_BUSY	141	Transmission/Reception currently ongoing, hence another transmission/reception could not be performed by Com Module.
RTE_E_OS_ERROR	200	Error occurred in OS service API.
RTE_E_INVALID_ARGS	201	The input argument is Null pointer.
RTE_E_UNINIT	204	API called before RTE initialization or after RTE finalization.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
42 / 146

Error Code	Value	Description
RTE_E_ILLEGAL_INVOCATION	203	API is called from Different Runnable or Task context.
RTE_E_IN_EXCLUSIVE_AREA	135	Blocking API is called from within an Exclusive Area.
RTE_E_SEG_FAULT	136	The error can be returned by an RTE API when it refers a memory location that is not accessible from the caller's partition.
SCHM_E_OK	0	No error occurred
SCHM_E_TIMEOUT	129	The configured timeout exceeds before the intended result was ready or any mode users partition is stopped or restarting or has been restarted while the mode switch was requested. Incoming client server requests before the basic software scheduler is initialized completely or after it is stopped.
SCHM_E_LIMIT	130	An internal Basic Software Scheduler limit has been exceeded. Request could not be handled. OUT buffers are not modified.
SCHM_E_NO_DATA	131	An explicit read API call returned no data. The BswModuleEntry's result is not available, but no other error occurred within the API call or the BswModuleEntry was not called using SchM_Call. The buffers for the IN/OUT and OUT parameters shall not be modified.
SCHM_E_TRANSMIT_ACK	132	Transmission acknowledgement received.
SCHM_E_UNINIT	204	API called before SchM initialization or after SchM finalization.
SCHM_E_INVALID	1	Invalid mode is passed as an input to SchM_Switch API.
SCHM_E_OS_ERROR	200	Error occurred in OS service API.
SCHM_E_UNCONNECTED	134	Unconnected Provided ModeGroups or requiredData.
SCHM_E_LOST_DATA	64	Overlaid error, indicates that some incoming data has been lost due to an overflow.
SCHM_E_INVALID_ARGS	201	The input argument is Null pointer.
SCHM_E_ILLEGAL_INVOCATION	203	API is called from the wrong context.

## 5.2 Interaction with another module

This section details the interaction and dependency of RTE module on other Bsw modules.

### 5.2.1 OS

RTE shall use OS services like ActivateTask, SetEvent, SetRelAlarm, StartScheduleTableRel, StartScheduleTableSynchron, StartScheduleTableAbs for scheduling and triggering runnables. RTE uses Os APIs GetElapsedvalue, GetCounterValue (Os hardware counter) for time out in case of mode switch and asynchronous communication APIs. RTE uses IOC feature to transfer data across partition. RTE also uses IOC callback to invoke function calls across partition.

### 5.2.2 COM

Com module provides interfaces to RTE to send, receive and invalidate signals/signal groups. Com will call configurable callbacks to provide notifications.

### 5.2.3 LdCom

LdCom module provides interfaces to RTE to send data and uses the callbacks to provide notifications.

### 5.2.4 DET

RTE module invokes DET interfaces to report errors. Along with all the DET errors already mentioned in the RTE SWS, the following DET errors are also reported to DET

1. RTE\_E\_OS\_ERROR: Error occurred when an OS service was invoked.
2. RTE\_E\_INCORRECT\_SEQUENCE: Sequence of calling LdCom callback API is incorrect.
3. RTE\_E\_PARAM\_POINTER: Null pointer check in LdCom callback.
4. RTE\_E\_XFRM\_FAILURE: Transformation check in Trigger Interface.
5. SCHM\_E\_OS\_ERROR: Error occurred when an OS service was invoked.
6. RTE\_E\_DET\_ILLEGAL\_INVOCATION: RTE API is called by a Runnable which should not call that RTE API.
7. RTE\_E\_DET\_WAIT\_IN\_EXCLUSIVE\_AREA: Application has called Rte\_Enter API and subsequently asks the RTE to enter a wait state.
8. RTE\_E\_DET\_SEG\_FAULT: RTE API call contains a direct or indirect reference to memory that is not accessible from the caller's partition.
9. RTE\_E\_DET\_ILLEGAL\_NESTED\_EXCLUSIVE\_AREA: When exclusive areas do not exit in the reverse order they were entered.

### 5.2.5 Transformer Modules

RTE shall invoke transformer and inverse transformer API to perform data transformation and retransformation. SomelPXf, COMBasedXf and E2EXf modules are supported.

## 5.3 Detailed Description of RTE

This section provides a detailed description of the RTE APIs.

### 5.3.1 SchM Lifecycle APIs

#### 5.3.1.1 SchM\_Init

SchM_Init			
Syntax	void SchM_Init (const SchM_ConfigType* ConfigPtr)		
ServiceID	0x00		
Parameters (IN/INOUT/OUT)	ConfigPtr	IN	Pointer to configuration set in Variant Post-Build.
Return Values	-		-
Description	SchM_Init is intended to allocate and initialize system resources used by the Basic Software Scheduler part of the RTE for the core on which it is called. SchM_Init API is always generated by Rte Code Generator. This function is used by BSW.		

#### 5.3.1.2 SchM\_DeInit

SchM_DeInit			
Syntax	void SchM_DeInit (void)		
ServiceID	0x01		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	SchM Deinit is used to finalize Basic Software Scheduler part of the RTE of the core on which it is called. This service releases all system resources allocated by the Basic Software Scheduler part on that core. SchM_DeInit API is always generated by Rte Code Generator. This function is used by BSW.		

#### 5.3.1.3 SchM\_Start

**SchM\_Start**

Syntax	void SchM_Start (void)		
ServiceID	0x70		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	SchM_Start initializes the Basic Software Scheduler. The API is called before BswM_Init() in the integrated environment. Periodical events for BSW/SWCs. SchM_Start API is always generated by Rte Code Generator. This function is used by BSW.		

**5.3.1.4 SchM\_StartTiming****SchM\_StartTiming**

Syntax	void SchM_StartTiming (void)		
ServiceID	0x76		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	Start periodical events for BSW/SWCs. SchM_StartTiming API is always generated by Rte Code Generator. This function is used by BSW.		

SchM\_Init, SchM\_Start, SchM\_StartTiming should be called only once by EcuM in the integrated environment.

**5.3.1.5 SchM\_GetVersionInfo****SchM\_GetVersionInfo**

Syntax	void SchM_GetVersionInfo (Std_VersionInfoType* versioninfo)		
ServiceID	0x02		
Parameters (IN/INOUT/OUT)	versioninfo	OUT	Pointer to the memory location holding the version information of the module
Return Values	-		-
Description	Returns the version information of the Basic Software Scheduler. This function is used by BSW.		

**5.3.2 RTE Lifecycle APIs**

### 5.3.2.1 Rte\_Start

Rte_Start			
Syntax	Std_ReturnType Rte_Start (void)		
ServiceID	0x70		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	RTE_E_OK		No error occurred
	RTE_E_LIMIT		An internal limit has been exceeded. The allocation of a required resource has failed.
Description	Rte_Start is intended to allocate and initialize system resources and communication resources used by the RTE. Rte_Start API is always generated by Rte Code Generator. This function is used by BSW.		

### 5.3.2.2 Rte\_Stop

Rte_Stop			
Syntax	Std_ReturnType Rte_Stop (void)		
ServiceID	0x71		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	RTE_E_OK		No error occurred
	RTE_E_LIMIT		A resource could not be released.
Description	Rte_Stop is used to finalize the RTE on the core it is called. This service releases all system and communication resources allocated by the RTE on that core. Rte_Stop API is always generated by Rte Code Generator. This function is used by BSW.		

### 5.3.2.3 Rte\_Init

Rte_Init	
Syntax	void Rte_Init_<InitContainer> (void)
ServiceID	0x75

**Rte\_Init**

Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	The API Rte_Init schedules RunnableEntitys for initialization purpose which are mapped to the related RteInitializationRunnableBatch container. This function is used by BSW.		

**5.3.2.4 Rte\_StartTiming****Rte\_StartTiming**

Syntax	void Rte_StartTiming (void)		
ServiceID	0x76		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	Rte_StartTiming API is intended to release the activation of RunnableEntitys triggered by TimingEvents and BackgroundEvents after the last call of an Rte_Init function. This function is used by BSW.		

**5.3.2.5 Rte\_PartitionTerminated\_<PID>****Rte\_PartitionTerminated\_<PID>**

Syntax	void Rte_PartitionTerminated_<PID> (void)		
ServiceID	0x72		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	Rte_PartitionTerminated_<PID> API indicates to the RTE that a partition is going to be terminated and communication with the partition shall be ignored. The API is invoked from Os ProtectionHook function. This function is used by BSW.		

**5.3.2.6 Rte\_PartitionRestarting\_<PID>**

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
48 / 146

### Rte\_PartitionRestarting\_<PID>

Syntax	void Rte_PartitionRestarting_<PID>(void)		
ServiceID	0x73		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	Rte_PartitionRestarting_<PID> API indicates to the RTE that a partition is going to be restarted and communication with the partition shall be ignored. The partition may be restarted later in the ECU lifecycle. The API is invoked from Os ProtectionHook function. This function is used by BSW.		

### 5.3.2.7 Rte\_RestartPartition\_<PID>

#### Rte\_RestartPartition\_<PID>

Syntax	Std_ReturnType Rte_RestartPartition_<PID>(void)		
ServiceID	0x74		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	RTE_E_OK		No error occurred
	RTE_E_LIMIT		An internal limit has been exceeded. The allocation of a required resource has failed.
	RTE_E_UNINIT		API is invoked before Rte_Start or after Rte_Stop.
Description	Rte_RestartPartition_<PID> API initializes the RTE resource that are allocated to partition specified by <PID>. The API is invoked from Restart Task. This function is used by BSW.		

### 5.3.3 Sender Receiver Communication APIs

This section describes detailed description of the APIs that are used for sender receiver communication.

#### 5.3.3.1 Rte\_DRead



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
49 / 146

### Rte\_DRead

Syntax	<return> Rte_DRead_<p>_<o> ([OUT Rte_TransformerError transformerError])		
ServiceID	0x1A		
Parameters (IN/INOUT/OUT)	transformerError	OUT	optional parameter that contains transformer error which occurred during execution of transformer chain
Return Values	<return>		The return type of Rte_DRead is dependent on the ImplementationDataType of the data element. The return value provides access to the data value of the data element.
Description	Performs an explicit non-queued read on a sender-receiver communication data element. The Rte_DRead API is used for explicit read by value. This function is used by user.		

### 5.3.3.2 Rte\_Feedback

#### Rte\_Feedback

Syntax	Std_ReturnType Rte_Feedback_<p>_<o> (void)		
ServiceID	0x17		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	RTE_E_TRANSMIT_ACK		Data Element was transmitted successfully.
	RTE_E_UNCONNECTED		Provided Port is unconnected.
	RTE_E_UNINIT		DET error, hence data element is not passed (RTE is not initialized).
	RTE_E_ILLEGAL_INVOCATION		API is called from different Runnable or Task context than expected.
	RTE_E_OS_ERROR		Error occurred in the OS service invoked from within the Rte_Feedback API.
	RTE_E_HARD_TRANSFORMER_ERROR		Hard error occurred in transformer or in one of the transformers in a

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
50 / 146

### Rte\_Feedback

		transformer chain.
	RTE_E_SOFT_TRANSFORMER_ERROR	Soft error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_COM_STOPPED	The last transmission was rejected (when the Rte_Send or Rte_Write API was called), with an RTE_E_COM_STOPPED return code or an error notification from COM was received.
	RTE_E_TIMEOUT	A timeout notification was received from COM.
	RTE_E_IN_EXCLUSIVE_AREA	Blocking Rte_Feedback API is invoked from inside an Exclusive Area.
	RTE_E_NO_DATA	No acknowledgments or error notifications were received from COM when the API was called (non-blocking call) or when the waitpoint timeout expired or COM returns COM_BUSY.
Description	Provide acknowledgement notification or error notification for an explicit sender receiver communication. This function is used by user.	

### 5.3.3.3 Rte\_Invalidate

#### Rte\_Invalidate

Syntax	Std_ReturnType Rte_Invalidate_<p>_<o> ([OUT Rte_TransformerError transformerError])		
ServiceID	0x16		
Parameters (IN/INOUT/OUT)	transformerError	OUT	optional parameter that contains transformer error that occurred during execution of transformer chain
Return Values	RTE_E_OK		No error or if the provided port is unconnected.
	RTE_E_INVALID_ARGS		The argument <transformerError> is a

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
51 / 146

### Rte\_Invalidate

		NULL pointer.
	RTE_E_UNINIT	API invoked before calling Rte_Start or after calling Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION	API is called from different Runnable or Task context than expected.
	RTE_E_OS_ERROR	Error occurred in the OS service invoked from within the Rte_Invalidate API.
	RTE_E_HARD_TRANSFORMER_ERROR	Hard error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_SOFT_TRANSFORMER_ERROR	Soft error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_COM_STOPPED	Communication service is currently not available (inter ECU)
	RTE_E_COM_BUSY	The transmission is rejected due to a currently ongoing transmission (inter ECU)
	RTE_E_TIMEOUT	Cross Core timeout occurs, during multipartition scenario
Description	Invalidate a data element for an explicit non-queued sender-receiver transmission. This function is used by user.	

### 5.3.3.4 Rte\_Write

#### Rte\_Write

Syntax	Std_ReturnType Rte_Write_<p>_<o> (IN <data>, [OUT Rte_TransformerError transformerError])		
ServiceID	0x14		
Parameters (IN/INOUT/OUT)	<data>	IN	Data element to write. <data> is passed by value or reference according to the

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
52 / 146

### Rte\_Write

			ImplementationDataType.
	transformerError	OUT	Optional parameter that contains transformer error that occurred during execution of transformer chain.
Return Values	RTE_E_OK		Data passed to communication service successfully or there are no DataMappings or if the provided port is unconnected or if receiver partition is terminated and RteOsErrorDetect parameter is set to false
	RTE_E_INVALID_ARGS		The argument <data> or <transformerError> is a NULL pointer.
	RTE_E_UNINIT		API invoked before calling Rte_Start or after calling Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION		API is called from different Runnable or Task context than expected.
	RTE_E_OS_ERROR		Error occurred in the OS service invoked from within the Rte_Write API or if receiver partition is terminated and RteOsErrorDetect parameter is set to true.
	RTE_E_HARD_TRANSFORMER_ERROR		Hard error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_SOFT_TRANSFORMER_ERROR		Soft error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_COM_STOPPED		Communication service is currently not available (inter ECU)
	RTE_E_COM_BUSY		The transmission is rejected due to a currently ongoing transmission (inter ECU)

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
53 / 146

### Rte\_Write

	RTE_E_TIMEOUT	Cross Core timeout occurs, during multipartition scenario
	RTE_E_SEG_FAULT	A segmentation violation is detected in the handed over parameters to the RTE API
Description	Initiates an explicit non-queued sender-receiver transmission of data elements. This function is used by user.	

### 5.3.3.5 Rte\_IsUpdated

#### Rte\_IsUpdated

Syntax	<return> Rte_IsUpdated_<p>_<o> (void)		
ServiceID	0x30		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	TRUE		Data element updated since last read.
	FALSE		Data element not updated since read.
Description	Provide access to the update flag for an explicit non-queued receiver. This function is used by user.		

### 5.3.3.6 Rte\_Read

#### Rte\_Read

Syntax	Std_ReturnType Rte_Read_<p>_<o> (OUT <data>, [OUT Rte_TransformerError transformerError])		
ServiceID	0x19		
Parameters (IN/INOUT/OUT)	<data>	OUT	Data element to pass back the received data. <data> is passed by value or reference according to the ImplementationDataType.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
54 / 146

### Rte\_Read

	transformerError	OUT	Optional parameter that contains transformer error which occurred during execution of transformer chain.
Return Values	RTE_E_OK		Data read successfully.
	RTE_E_INVALID_ARGS		The argument <data> or <transformerError> is a NULL pointer.
	RTE_E_UNINIT		API invoked before calling Rte_Start or after calling Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION		API is called from different Runnable or Task context than expected.
	RTE_E_UNCONNECTED		Required Port is unconnected.
	RTE_E_HARD_TRANSFORMER_ERROR		Hard error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_SOFT_TRANSFORMER_ERROR		Soft error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_COM_STOPPED		Communication service is currently not available (inter ECU)
	RTE_E_COM_BUSY		The read request is rejected due to a currently ongoing reception (inter ECU)
	RTE_E_NEVER_RECEIVED		No data received since Rte initialization.
	RTE_E_INVALID		Received data is invalid.
	RTE_E_OUT_OF_RANGE		Received data is out of range.
	RTE_E_MAX_AGE_EXCEEDED		Received data is outdated.
Description	Performs an explicit non-queued read on a sender-receiver communication for a data element. This function is used by user.		

### 5.3.3.7 Rte\_Send

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
55 / 146

### Rte\_Send

Syntax	Std_ReturnType Rte_Send_<p>_<o> (IN <data>, [OUT Rte_TransformerError transformerError])		
ServiceID	0x13		
Parameters (IN/INOUT/OUT)	<data>	IN	Data element to write. <data> is passed by value or reference according to the ImplementationDataType.
	transformerError	OUT	Optional parameter that contains transformer error that occurred during execution of transformer chain.
Return Values	RTE_E_OK		Data passed to communication service successfully or there are no Data Mappings or if the provided port is unconnected.
	RTE_E_INVALID_ARGS		The argument <data> or <transformerError> is a NULL pointer.
	RTE_E_UNINIT		API invoked before calling Rte_Start or after calling Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION		API is called from different Runnable or Task context than expected.
	RTE_E_OS_ERROR		Error occurred in the OS service invoked from within the Rte_Send API.
	RTE_E_HARD_TRANSFORMER_ERROR		Hard error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_SOFT_TRANSFORMER_ERROR		Soft error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_COM_STOPPED		Communication service is currently not available (inter ECU).
	RTE_E_COM_BUSY		The transmission is rejected due to a currently ongoing transmission (inter

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
56 / 146

### Rte\_Send

		ECU).
	RTE_E_LIMIT	<data> is discarded as receiver's queue is full (intra communication).
	RTE_E_SEG_FAULT	A segmentation violation is detected in the handed over parameters to the RTE API
	RTE_E_TIMEOUT	Cross Core timeout occurs, during multipartition scenario
Description	Initiates an explicit queued sender-receiver transmission of data elements. This function is used by user.	

### 5.3.3.8 Rte\_Receive

#### Rte\_Receive

Syntax	Std_ReturnType Rte_Receive_<p>_<o> (OUT <data>, [OUT Rte_TransformerError transformerError])		
ServiceID	0x1B		
Parameters (IN/INOUT/OUT)	<data>	OUT	Data element to pass back the received data. <data> is passed by reference.
	transformerError	OUT	Optional parameter that contains transformer error which occurred during execution of transformer chain.
Return Values	RTE_E_OK		Data read successfully.
	RTE_E_INVALID_ARGS		The argument <data> or <transformerError> is a NULL pointer.
	RTE_E_UNINIT		API invoked before calling Rte_Start or after calling Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION		API is called from different Runnable or Task context than expected.
	RTE_E_UNCONNECTED		Required Port is unconnected.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
57 / 146

### Rte\_Receive

	RTE_E_HARD_TRANSFORMER_ERROR	Hard error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_SOFT_TRANSFORMER_ERROR	Soft error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_COM_BUSY	The read request is rejected due to a currently ongoing reception (inter ECU LdCom)
	RTE_E_NO_DATA	No data was received since Rte initialization (Non-Blocking only).
	RTE_E_TIMEOUT	Data was not received inside the timeout value (Blocking only).
	RTE_E_OS_ERROR	Error occurred in the OS service invoked in the Rte_Receive API. (Blocking only)
	RTE_E_IN_EXCLUSIVE_AREA	Blocking Rte_Receive API is invoked from inside an Exclusive Area.
	RTE_E_LOST_DATA	Received data is lost due to receive buffer overflow or due to an error of the underlying communication layers.
Description	Performs an explicit queued read on a sender-receiver communication data element. This function is used by user.	

### 5.3.3.9 Rte\_IRead

#### Rte\_IRead

Syntax	<return> Rte_IRead_<re>_<p>_<o> ()		
ServiceID	0x21		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return>		Data element to be read. The type of return depends on ImplementationDataType.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
58 / 146

### Rte\_IRead

Description	Performs an implicit read on a sender-receiver communication data element in the runnable entity. This function is used by user.
-------------	--

### 5.3.3.10 Rte\_IWrite

#### Rte\_IWrite

Syntax	void Rte_IWrite_<re>_<p>_<o> (IN <data>)		
ServiceID	0x22		
Parameters (IN/INOUT/OUT)	<data>	IN	Data Element to be written.
Return Values	-		-
Description	Performs an implicit write on a sender-receiver communication data element in a runnable entity. This function is used by user.		

### 5.3.3.11 Rte\_IWriteRef

#### Rte\_IWriteRef

Syntax	<return reference> Rte_IWriteRef_<re>_<p>_<o> ()		
ServiceID	0x23		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return reference>		returns a reference to the corresponding data element
Description	Returns the reference data element in a runnable entity to a sender-receiver communication. This function is used by user.		

### 5.3.3.12 Rte\_IInvalidate

#### Rte\_IInvalidate

Syntax	<void> Rte_IInvalidate_<re>_<p>_<o> ()		
ServiceID	0x24		
Parameters (IN/INOUT/OUT)	-	-	-

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
59 / 146

### Rte\_Invalidate

Return Values	-	-
Description	Invalidate a data element for an implicit sender-receiver transmission in a runnable entity. This function is used by user.	

### 5.3.3.13 Rte\_IStatus

#### Rte\_IStatus

Syntax	Std_ReturnType Rte_IStatus_<re>_<p>_<o> ([OUT Rte_TransformerError transformerError])		
ServiceID	0x25		
Parameters (IN/INOUT/OUT)	transformerError	OUT	Optional parameter that contains transformer error which occurred during execution of transformer chain.
Return Values	RTE_E_OK		No error
	RTE_E_INVALID_ARGS		The argument <transformerError> is a NULL pointer.
	RTE_E_UNINIT		API invoked before calling Rte_Start or after calling Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION		API is called from different Runnable or Task context than expected.
	RTE_E_UNCONNECTED		Required Port is unconnected.
	RTE_E_HARD_TRANSFORMER_ERROR		Hard error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_SOFT_TRANSFORMER_ERROR		Soft error occurred in transformer or in one of the transformers in a transformer chain.
	RTE_E_COM_STOPPED		<ul style="list-style-type: none"> <li>– in case of COM the corresponding service returns COM_SERVICE_NOT_AVAILABLE</li> <li>– in case of LdCom the corresponding LdCom_Transmit returns E_NOT_OK</li> <li>– in case of LdCom, the data is Invalid</li> </ul>

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
60 / 146

### Rte\_IStatus

	RTE_E_COM_BUSY	The read request is rejected due to a currently ongoing reception (inter ECU)
	RTE_E_NEVER_RECEIVED	No data received since Rte initialization.
	RTE_E_INVALID	Data element is invalid.
	RTE_E_OUT_OF_RANGE	Data element is out of range.
	RTE_E_MAX_AGE_EXCEEDED	Data element is outdated.
Description	Provide the error status of a data element. This function is used by user.	

### 5.3.3.14 Rte\_IFeedback

#### Rte\_IFeedback

Syntax	Std_ReturnType Rte_IFeedback_<re>_<p>_<o> (void)		
ServiceID	0x2F		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	RTE_E_TRANSMIT_ACK	Data Element was transmitted successfully.	
	RTE_E_UNCONNECTED	Provided Port is unconnected.	
	RTE_E_UNINIT	DET error, hence data element is not passed (RTE is not initialized).	
	RTE_E_ILLEGAL_INVOCATION	API is called from different Runnable or Task context than expected.	
	RTE_E_HARD_TRANSFORMER_ERROR	Hard error occurred in transformer or in one of the transformers in a transformer chain.	
	RTE_E_SOFT_TRANSFORMER_ERROR	Soft error occurred in transformer or in one of the transformers in a transformer chain.	
	RTE_E_COM_STOPPED	The last transmission was rejected (when Rte_IWrite API was called), with an RTE_E_COM_STOPPED return code	

## Rte User Manual

Document number  
(DOC NO)SHT/SHTS  
61 / 146

## Rte\_IFeedback

		or an error notification from COM was received.
	RTE_E_TIMEOUT	A timeout notification was received from COM.
	RTE_E_NO_DATA	No acknowledgments or error notifications were received from COM when the API was called.
Description	Provide access to acknowledgement notifications for implicit sender receiver communication and to pass error notification to sender. This function is used by user.	

### 5.3.4 Client-Server Communication APIs

The section provides detailed description of the APIs that are used for client-server communication.

#### 5.3.4.1 Rte\_Call

Rte_Call			
Syntax	Std_ReturnType Rte_Call_<p>_<o> ([IN IN/OUT OUT] <data_1>...<data_n>, [OUT Rte_TransformerError transformerError])		
ServiceID	0x1C		
Parameters (IN/INOUT/OUT)	<data_1>...<data_n>	[IN IN/OUT OUT]	Argument data of the server operation.
	transformerError	OUT	optional parameter that contains transformer error that occurred during execution of transformer.
Return Values	RTE_E_OK		API call completed successfully.
	RTE_E_INVALID_ARGS		The argument <data> or <transformerError> is a NULL pointer.
	RTE_E_UNCONNECTED		Client port is not connected.
	RTE_E_TIMEOUT		No reply was received within timeout period from server (Sync only). API is invoked before Rte_Start or after Rte_Stop. Cross Core timeout occurs, during multipartition scenario
	RTE_E_ILLEGAL_INVOCATION		API is called from different Runnable or Task context than expected.
	RTE_E_OS_ERROR		Error occurred in the OS service invoked from within the Rte_Call API.
	RTE_E_IN_EXCLUSIVE_AREA		Rte_Call (Sync Inter ECU) with timeout invoked from within an exclusive area
	RTE_E_HARD_TRANSFORMER_ERROR		Hard error occurred in transformer.
	RTE_E_SOFT_TRANSFORMER_ERROR		Soft error occurred in transformer.
	RTE_E_COM_STOPPED		in case of COM the corresponding service returns COM_SERVICE_NOT_AVAILABLE

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
63 / 146

### Rte\_Call

		in case of LdCom the corresponding LdCom_Transmit returns E_NOT_OK
	RTE_E_COM_BUSY	The transmission is rejected due to a currently ongoing transmission (inter ECU)
	RTE_E_LIMIT	Client has multiple outstanding asynchronous client-server invocations of the same operation in the same port
	Application Errors	Application specific errors are returned when no other Error exist (Sync only).
	RTE_E_SEG_FAULT	A segmentation violation is detected in the handed over parameters to the RTE API
Description	Initiate a synchronous or asynchronous client-server communication. This function is used by user.	

### 5.3.4.2 Rte\_Result

#### Rte\_Result

Syntax	Std_ReturnType Rte_Result_<p>_<o> ([IN/OUT OUT <param 1>]... [IN/OUT OUT <paramN>], [OUT Rte_TransformerError transformerError])		
ServiceID	0x1D		
Parameters (IN/INOUT/OUT)	<param1>...<paramN>	[IN/OUT OUT]	Result data of the operation.
	transformerError	OUT	Optional parameter that contains transformer error that occurred during execution of transformer.
Return Values	RTE_E_OK	API call completed successfully.	
	RTE_E_INVALID_ARGS	The argument <param> or <transformerError> is a NULL pointer.	
	RTE_E_UNCONNECTED	Client port is not connected	
	RTE_E_TIMEOUT	No reply was received from server within timeout period from server.	

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
64 / 146

### Rte\_Result

		API is invoked before Rte_Start or after Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION	API is called from different Runnable or Task context than expected.
	RTE_E_OS_ERROR	Error occurred in the OS service invoked from within the Rte_Result API (Blocking).
	RTE_E_IN_EXCLUSIVE_AREA	Blocking Rte_Result is invoked from within an exclusive area.
	Application Errors	Application specific errors are returned when no other Error exist.
	RTE_E_NO_DATA	The server's result is not available or previous call of Rte_Call returned RTE_E_HARD_TRANSFORMER_ERROR, RTE_E_SEG_FAULT
	RTE_E_SEG_FAULT	A segmentation violation is detected in the handed over parameters to the RTE API
Description	Get the result of an asynchronous client-server call. This function is used by user.	

**Note:** RTE\_E\_HARD\_TRANSFORMER\_ERROR, RTE\_E\_SOFT\_TRANSFORMER\_ERROR, RTE\_E\_COM\_STOPPED, RTE\_E\_COM\_BUSY: Based on design decisions SAFE\_RTE\_DD\_0168 and SAFE\_RTE\_DD\_0172, above errors are not supported in case of Inter ECU Client Server Communication.

### 5.3.5 RTE Exclusive Area APIs

#### 5.3.5.1 Rte\_Enter

### Rte\_Enter

Syntax	void Rte_Enter_[<re_>] <name> ()		
ServiceID	0x2A		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	Enter an exclusive area. This function is used by user.		



### 5.3.5.2 Rte\_Exit

Rte_Exit			
Syntax	void Rte_Exit_[<re_>] <name> ()		
ServiceID	0x2B		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	Leave an exclusive area. This function is used by user.		

### 5.3.6 COM Callbacks

This section describes detailed description of the callbacks that are used for communication with AUTOSAR COM.

#### 5.3.6.1 Rte\_COMCbK\_<sn>

Rte_COMCbK			
Syntax	void Rte_COMCbK_<sn> (void)		
ServiceID	0x9F		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a signal for a primitive data is ready for reception. This function is used by BSW.		

#### 5.3.6.2 Rte\_COMCbK\_<sg>

Rte_COMCbK			
Syntax	void Rte_COMCbK_<sg> (void)		
ServiceID	0x95		
Parameters (IN/INOUT/OUT)	-	-	-

## Rte\_COMCbK

Return Values	-	-
Description	This callback indicates that a signal group for a composite data is ready for reception. This function is used by BSW.	

### 5.3.6.3 Rte\_COMCbKTAck\_<sn>

## Rte\_COMCbKTAck

Syntax	void Rte_COMCbKTAck_<sn> (void)		
ServiceID	0x90		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a signal for a primitive data is transmitted by COM to the PDU-Router. This function is used by BSW.		

### 5.3.6.4 Rte\_COMCbKTAck\_<sg>

## Rte\_COMCbKTAck

Syntax	void Rte_COMCbKTAck_<sg> (void)		
ServiceID	0x96		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a signal group for a composite data is transmitted by COM to the PDU-Router. This function is used by BSW.		

### 5.3.6.5 Rte\_COMCbKTErr\_<sn >

## Rte\_COMCbKTErr

Syntax	void Rte_COMCbKTErr_<sn> (void)		
ServiceID	0x91		
Parameters (IN/INOUT/OUT)	-	-	-

## Rte\_COMCbkTErr

Return Values	-	-
Description	This callback indicates that a signal for a primitive data is not transmitted due to error that occurred when data was handed from COM to the PDU-Router. This function is used by BSW.	

### 5.3.6.6 Rte\_COMCbkTErr\_<sg>

## Rte\_COMCbkTErr

Syntax	void Rte_COMCbkTErr_<sg> (void)		
ServiceID	0x97		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a signal group for a composite data is not transmitted due to error that occurred when data was handed from COM to the PDU-Router.  This function is used by BSW.		

### 5.3.6.7 Rte\_COMCbkJnv\_<sn>

## Rte\_COMCbkJnv

Syntax	void Rte_COMCbkInv_<sn> (void)		
ServiceID	0x92		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a signal for a primitive data is received as invalid. This function is used by BSW.		

### 5.3.6.8 Rte\_COMCbkJnv\_<sg>

## Rte\_COMCbkJnv

Syntax	void Rte_COMCbkJnv_<sg> (void)		
--------	--------------------------------	--	--

## Rte\_COMCbklInv

ServiceID	0x98		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a signal group for a composite data is received as invalid. This function is used by BSW.		

### 5.3.6.9 Rte\_COMCbRxtOut\_<sn>

## Rte\_COMCbRxtOut

Syntax	void Rte_COMCbRxtOut_<sn> (void)		
ServiceID	0x93		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a no primitive data is received when the aliveTimeOut expires for a signal. This informs that the data is outdated. This function is used by BSW.		

### 5.3.6.10 Rte\_COMCbRxtOut\_<sg>

## Rte\_COMCbRxtOut

Syntax	void Rte_COMCbRxtOut_<sg> (void)		
ServiceID	0x99		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a no composite data is received when the aliveTimeOut expires for a signal group. This informs that the data is outdated. This function is used by BSW.		

### 5.3.6.11 Rte\_COMCbTxTOut\_<sn >

## Rte\_COMCbKTxTOut

Syntax	void Rte_COMCbKTxTOut_<sn> (void)		
ServiceID	0x94		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a signal for a primitive data was not transmitted due to timeout of TransmissionAcknowledgement. This function is used by BSW.		

### 5.3.6.12 Rte\_COMCbKTxTOut\_<sg>

## Rte\_COMCbKTxTOut

Syntax	void Rte_COMCbKTxTOut_<sg> (void)		
ServiceID	0x9A		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	This callback indicates that a signal group for a composite data was not transmitted due to timeout of TransmissionAcknowledgement. This function is used by BSW.		

## 5.3.7 LdCOM Callbacks

This section describes detailed description of the callbacks that are used for communication with AUTOSAR LdCOM.

### 5.3.7.1 Rte\_LdComCbKRxIndication\_<sn>

## Rte\_LdComCbKRxIndication

Syntax	void Rte_LdComCbKRxIndication_<sn> (const PduInfoType* PduInfoPTR)		
ServiceID	0xA0		
Parameters (IN/INOUT/OUT)	PduInfoPtr	IN	Contains the length (SduLength) of the received PDU, a pointer to a buffer

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
70 / 146

### Rte\_LdComCbKRxIndication

			(SduDataPtr) containing the PDU, and the MetaData related to this PDU
Return Values	-	-	-
Description	Initiates received PDU from a lower layer communication interface module. This function is used by BSW.		

### 5.3.7.2 Rte\_LdComCbKStartOfReception\_<sn>

#### Rte\_LdComCbKStartOfReception

Syntax	BufReq_ReturnType Rte_LdComCbKStartOfReception_<sn> (const PduInfoType* info, PduLengthType TpSduLength, PduLengthType* bufferSizePTR)		
ServiceID	0xA1		
Parameters (IN/INOUT/OUT)	info	IN	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
	TpSduLength	IN	Total length of the N-SDU to be received
	bufferSizePtr	OUT	Available receive buffer in the receiving module. This parameter will be used to compute the Block Size in the transport protocol module
Return Values	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.
	BUFREQ_E_NOT_OK		Connection has been rejected; reception is aborted. bufferSizePtr remains unchanged.

## Rte\_LdComCbKStartOfReception

	BUFREQ_E_OVFL	No buffer of the required length can be provided; reception is aborted. bufferSizePtr remains unchanged.
Description	This callback initiates the start of receiving an N-SDU. The N-SDU might be fragmented into multiple N-PDUs or might consist of a single N-PDU. This function is used by BSW.	

### 5.3.7.3 Rte\_LdComCbKCopyRxData\_<sn>

## Rte\_LdComCbKCopyRxData

Syntax	BufReq_ReturnType Rte_LdComCbKCopyRxData_<sn> (const PduInfoType* info, PduLengthType* bufferSizePTR)		
ServiceID	0xA2		
Parameters (IN/INOUT/OUT)	info	IN	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). 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
	bufferSizePtr	OUT	Available receive buffer after data has been copied
Return Values	BUFREQ_OK		Data copied successfully.
	BUFREQ_E_NOT_OK		Data was not copied because an error occurred.
Description	This callback initiates the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this callback provides the next part of the I-PDU data. The size of the remaining data is written to the position indicated by bufferSizePtr. This function is used by BSW.		

### 5.3.7.4 Rte\_LdComCbKTpRxIndication\_<sn>

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
72 / 146

### Rte\_LdComCbKTPRxIndication

Syntax	void Rte_LdComCbKTPRxIndication_<sn> (Std_ReturnType resuLT)		
ServiceID	0xA3		
Parameters (IN/INOUT/OUT)	result	IN	Result of the reception
Return Values	-		-
Description	This callback indicates whether the reception is successful or not. This function is used by BSW.		

### 5.3.7.5 Rte\_LdComCbKCopyTxData\_<sn>

#### Rte\_LdComCbKCopyTxData

Syntax	BufReq_ReturnType Rte_LdComCbKCopyTxData_<sn> (const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPTR)		
ServiceID	0xA4		
Parameters (IN/INOUT/OUT)	info	IN	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). 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
	retry	IN	Will not be handled by LdCom or RTE it only is used to check.
	availableDataPtr	OUT	Indicates the remaining number of bytes that are available in the upper layer module's Txbuffer.
Return Values	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. No data has been copied.
	BUFREQ_E_NOT_OK		Data has not been copied. Request



## Rte\_LdComCbkJCopyTxData

failed.

### Description

This callback transmits the 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. This function is used by BSW.

## 5.3.7.6 Rte\_LdComCbkJTpTxConfirmation\_<sn>

### Rte\_LdComCbkJTpTxConfirmation

#### Syntax

void Rte\_LdComCbkJTpTxConfirmation\_<sn> (Std\_ReturnType resuLT)

#### ServiceID

0xA5

#### Parameters (IN/INOUT/OUT)

result

IN

informs if the transmission was successful(E\_OK) or not successful(E\_NOT\_OK)

#### Return Values

-

-

#### Description

This callback indicates that a Signal has been transmitted via the TP-API on its network. This function is used by BSW.

## 5.3.7.7 Rte\_LdComCbkJTriggerTransmit\_<sn>

### Rte\_LdComCbkJTriggerTransmit

#### Syntax

Std\_ReturnType Rte\_LdComCbkJTriggerTransmit\_<sn> (PduInfoType\* PduInfoPTR)

#### ServiceID

0xA6

#### Parameters (IN/INOUT/OUT)

PduInfoPtr

INOUT

Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLength. On return, the service will indicate the length of the copied SDU data in SduLength

## Rte\_LdComCbKTriggerTransmit

Return Values	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	This callback checks 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. This function is used by BSW.	

### 5.3.7.8 Rte\_LdComCbKTxConfirmation\_<sn>

## Rte\_LdComCbKTxConfirmation

Syntax	void Rte_LdComCbKTxConfirmation_<sn> (Std_ReturnType resuLT)		
ServiceID	0xA5		
Parameters (IN/INOUT/OUT)	result	IN	informs if the transmission was successful(E_OK) or not successful(E_NOT_OK)
Return Values	-	-	-
Description	This callback indicates the transmission of a PDU, or the failure to transmit a PDU. This function is used by BSW.		

## 5.3.8 NVM Service Callbacks

This section describes detailed description of the callbacks that are used for communication with AUTOSAR NvM.

### 5.3.8.1 Rte\_SetMirror

## Rte\_SetMirror\_<b>\_<d>

Syntax	Std_ReturnType Rte_SetMirror_<b>_<d>(const void* NVMBuffer)
ServiceID	0x9B

## Rte\_SetMirror\_<b>\_<d>

Parameters (IN/INOUT/OUT)	NVMBuffer	IN	Source buffer pointer
Return Values	E_OK		The copy is successful.
	SCHM_E_UNINIT		API called before SchM initialization or after SchM finalization.
	E_NOT_OK		The copy could not be performed.
Description	The Rte_SetMirror API copies the values of the VariableDataPrototypes contained in a NvBlockDescriptor from an NVM internal buffer to their locations in the RTE. This function is used by BSW.		

### 5.3.8.2 Rte\_GetMirror

## Rte\_GetMirror\_<b>\_<d>

Syntax	Std_ReturnType Rte_GetMirror_<b>_<d>(void* NVMBuffer)		
ServiceID	0x9C		
Parameters (IN/INOUT/OUT)	NVMBuffer	IN	Destination buffer pointer
Return Values	E_OK		The copy is successful.
	SCHM_E_UNINIT		API called before SchM initialization or after SchM finalization.
	E_NOT_OK		The copy could not be performed.
Description	The Rte_GetMirror API copies the values of the VariableDataPrototypes contained in a NvBlockDescriptor to a specified NVM internal buffer. This function is used by BSW.		

### 5.3.8.3 Rte\_NvMNotifyJobFinished

## Rte\_NvMNotifyJobFinished\_<b>\_<d>

Syntax	Std_ReturnType Rte_NvMNotifyJobFinished_<b>_<d> (uint8 ServiceId, NvM_RequestResultType JobResult)		
ServiceID	0x9D		
Parameters (IN/INOUT/OUT)	ServiceId	IN	Unique Service ID of NVRAM manager service.

**Rte\_NvMNotifyJobFinished\_<b>\_<d>**

	JobResult	IN	Covers the job result of the processed NvM job.
Return Values	E_OK		The Rte_NvMNotifyJobFinished API shall return E_OK.
	RTE_E_UNINIT		The callback is invoked before Rte_Start or after Rte_Stop.
Description	The Rte_NvMNotifyJobFinished receives the notification from the NvM when a job is finished and forward it to the SW-C. This function is used by BSW.		

**5.3.8.4 Rte\_NvMNotifyInitBlock****Rte\_NvMNotifyInitBlock\_<b>\_<d>**

Syntax	Std_ReturnType Rte_NvMNotifyInitBlock_<b>_<d>(void)		
ServiceID	0x9E		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	E_OK		The Rte_NvMNotifyInitBlock API shall return E_OK.
	RTE_E_UNINIT		The callback is invoked before Rte_Start or after Rte_Stop.
Description	The Rte_NvMNotifyInitBlock API receives the notification from the NvM when initialization of the mirror is requested. This function is used by BSW.		

**5.3.9 SchM Exclusive Area APIs****5.3.9.1 SchM\_Enter****SchM\_Enter**

Syntax	void SchM_Enter_<bsnp>[_<vi>_<ai>] _<me>_<name> ()		
ServiceID	0x03		
Parameters (IN/INOUT/OUT)	-	-	-

## SchM\_Enter

Return Values	-	-
Description	SchM_Enter function enters an exclusive area of a Basic Software Module. This function is used by user.	

### 5.3.9.2 SchM\_Exit

## SchM\_Exit

Syntax	void SchM_Exit_<bsnp>[_<vi>_<ai>] _[<me>_] <name> ()		
ServiceID	0x04		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	SchM_Exit function leaves an exclusive area of a Basic Software Module. This function is used by user.		

## 5.3.10 Inter Runnable Variables APIs

This section describes detailed description of the APIs that are used for Inter Runnable Variables.

### 5.3.10.1 Rte\_IrvIRead

## Rte\_IrvIRead

Syntax	void Rte _IrvIRead_<re>_<o> (void)		
ServiceID	0x26		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	Provides read access to the InterRunnableVariables with implicit behavior of an AUTOSAR SW-C. This function is used by user.		

### 5.3.10.2 Rte\_IrvIWrite

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
78 / 146

### Rte\_IrvWrite

Syntax	void Rte_IrvWrite_<re>_<o> (IN <data>)		
ServiceID	0x27		
Parameters (IN/INOUT/OUT)	<data>	IN	Placeholder for the data the InterRunnableVariable shall be set to.
Return Values	-		-
Description	Provides write access to the InterRunnableVariables with implicit behavior of an AUTOSAR SW-C. This function is used by user.		

### 5.3.10.3 Rte\_IrvWriteRef

#### Rte\_IrvWriteRef

Syntax	<return reference> Rte_IrvWriteRef_<re>_<o> (void)		
ServiceID	0x31		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return reference>		The return reference type of Rte_IrvWriteRef is dependent on the ImplementationDataType of the VariableDataPrototype and is a pointer to the location where the value can be accessed.
Description	Provide a reference to the VariableDataPrototype defined with the implicit InterRunnableVariable role referenced by a VariableAccess in the written Local Variable role. This function is used by user.		

### 5.3.10.4 Rte\_IrvRead

Primitive type signature:

#### Rte\_IrvRead

Syntax	<return> Rte_IrvRead_<re>_<o> (void)		
ServiceID	0x28		

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
79 / 146

### Rte\_IrvRead

Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return>		The return value is the requested data value.
Description	Provides read access to the InterRunnableVariables with explicit behavior of an AUTOSAR SW-C. This function is used by user.		

Complex type signature:

### Rte\_IrvRead

Syntax	void Rte_IrvRead_<re>_<o> (OUT <data>)		
ServiceID	0x28		
Parameters (IN/INOUT/OUT)	<data>	OUT	The OUT parameter <data> is typed as reference (pointer) to the type of the InterRunnableVariable.
Return Values	-		-
Description	Provides read access to the InterRunnableVariables with explicit behavior of an AUTOSAR SW-C. This function is used by user.		

### 5.3.10.5 Rte\_IrvWrite

### Rte\_IrvWrite

Syntax	void Rte_IrvWrite_<re>_<o> (IN <data>)		
ServiceID	0x29		
Parameters (IN/INOUT/OUT)	<data>	IN	Placeholder for the data the InterRunnableVariable shall be set to.
Return Values	-		-
Description	Provides write access to the InterRunnableVariables with explicit behavior of an AUTOSAR SW-C. This function is used by user.		

### 5.3.11 RTE Mode Switch APIs

This section describes detailed description of the APIs that are used for Mode Switch by SWC.

## 5.3.11.1 Rte\_Switch

Rte_Switch			
Syntax	Std_ReturnType Rte_Switch_<p>_<o> (IN <mode>)		
ServiceID	0x15		
Parameters (IN/INOUT/OUT)	<mode>	IN	<mode >represents the next mode. <mode> is passed by value according to the ImplementationDataType on which the ModeDeclarationGroup is mapped.
Return Values	RTE_E_OK		Data passed to service successfully.
	RTE_E_LIMIT		A mode switch has been discarded by the receiver due to a full queue (Only synchronous mode switch).
	RTE_E_OS_ERROR		Error occurred in the OS service invoked from within the Rte_Switch API.
	RTE_E_INVALID		Invalid mode is provided as an input which is not present in ModeDeclarationGroup.
	RTE_E_UNINIT		API invoked before calling Rte_Start or after calling Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION		API is called from different Runnable or Task context than expected.
Description	Initiate a mode switch. The Rte_Switch API call is used for explicit sending of a mode switch notification. This function is used by user.		

## 5.3.11.2 Rte\_SwitchAck

Rte_SwitchAck			
Syntax	Std_ReturnType Rte_SwitchAck_<p>_<o> (void)		
ServiceID	0x18		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	RTE_E_TRANSMIT_ACK		The mode switch has been completed.
	RTE_E_TIMEOUT		- The configured timeout exceeds before the mode transition was



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
81 / 146

### Rte\_SwitchAck

		completed. - Any mode users partition is stopped or restarting or has been restarted while the mode switch was requested.
	RTE_E_OS_ERROR	Error occurred in the OS service invoked from within the Blocking Rte_SwitchAck API.
	RTE_E_NO_DATA	The mode switch is still in progress. Only for Non-Blocking Rte_SwitchAck API.
	RTE_E_UNINIT	API invoked before calling Rte_Start or after calling Rte_Stop.
	RTE_E_ILLEGAL_INVOCATION	API is called from different Runnable or Task context than expected.
	RTE_E_IN_EXCLUSIVE_AREA	Blocking Rte_SwitchAck API is invoked from inside an Exclusive Area.
	RTE_E_UNCONNECTED	Mode Provider port is not connected.
Description	Provide access to mode switch completed acknowledgements to mode managers. This function is used by user.	

### 5.3.11.3 Rte\_Mode

#### Rte\_Mode

Syntax	<return> Rte_Mode_<p>_<o> (void)		
ServiceID	0x2C		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return>		Returns the current mode.
Description	Provides the currently active mode of a mode switch port. This function is used by user.		

### 5.3.11.4 Rte\_Mode (Enhanced)

## Rte\_Mode

Syntax	<return> Rte_Mode_<p>_<o> (OUT <previousmode>, OUT <nextmode>)		
ServiceID	0x2C		
Parameters (IN/INOUT/OUT)	previousmode	OUT	If mode machine is in transition this OUT parameter provides previous mode.
	nextmode	OUT	If mode machine is in transition this OUT parameter provides next mode.
Return Values	<return>		Returns the current mode.
Description	Provides the currently active mode of a mode switch port. If the mode machine instance is in transition additionally the values of the previous and the next mode are provided. This function is used by user.		

## 5.3.12 SchM Mode Switch APIs

This section describes detail of the APIs that are used for Mode Switch by Bsw Modules.

### 5.3.12.1 SchM\_Switch

#### SchM\_Switch

Syntax	Std_ReturnType SchM_Switch_<bsnp> (IN <mode>)		
ServiceID	0x06		
Parameters (IN/INOUT/OUT)	<mode>	IN	<mode> represents the next mode. <mode> is passed by value according to the ImplementationDataType on which the ModeDeclarationGroup is mapped.
Return Values	SCHM_E_OK		Data passed to service is successful or Provided Mode Group is unconnected.
	SCHM_E_LIMIT		A mode switch has been discarded by the receiver due to a full queue (Only synchronous mode switch).
	SCHM_E_OS_ERROR		Error occurred in the OS service invoked from within the SchM_Switch API.
	SCHM_E_INVALID		Invalid mode is provided as an input which is not present in ModeDeclarationGroup.

## SchM\_Switch

	SCHM_E_UNINIT	API invoked before calling SchM_Init or after calling SchM_DeInit.
Description	This API Initiates a mode switch. The SchM_Switch API call used for sending of a mode switch notification by a Basic Software Module. This function is used by user.	

### 5.3.12.2 SchM\_SwitchAck

## SchM\_SwitchAck

Syntax	Std_ReturnType SchM_SwitchAck _<bsnp>_<name>(void)		
ServiceID	0x08		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	SCHM_E_TRANSMIT_ACK		The mode switch has been completed.
	SCHM_E_TIMEOUT		The configured timeout exceeds before the mode transition was completed.
	SCHM_E_NO_DATA		The mode switch is still in progress.  Only for Non-Blocking SchM_SwitchAck API.
	SCHM_E_UNCONNECTED		Provided Mode Group is unconnected.
	SCHM_E_UNINIT		API invoked before calling SchM_Init or after calling SchM_DeInit.
Description	This API Provides access to acknowledgment notifications for mode communication.  This function is used by user.		

### 5.3.12.3 SchM\_Mode

## SchM\_Mode

Syntax	<return> SchM_Mode_<bsnp>_<name> (void)		
ServiceID	0x07		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return>		Returns the current mode or in case of

## SchM\_Mode

		unconnected Mode Group returns the Initial Mode.
Description	This API Provides the currently active mode of a ModeDeclarationGroupPrototype. This function is used by user.	

### 5.3.12.4 SchM\_Mode (Enhanced)

## SchM\_Mode

Syntax	<return> SchM_Mode_<bsnp>_<name> (OUT <previousmode>, OUT <nextmode>)		
ServiceID	0x07		
Parameters (IN/INOUT/OUT)	previousmode	OUT	If mode machine is in transition this OUT parameter provides previous mode.
	nextmode	OUT	If mode machine is in transition this OUT parameter provides next mode.
Return Values	<return>		Returns the current mode or in case of unconnected Mode Group returns the Initial Mode.
Description	This API provides the currently active mode of a ModeDeclarationGroupPrototype. If the mode machine instance is in transition additionally the values of the previous and the next mode are provided. This function is used by user.		

### 5.3.13 RTE Trigger APIs

This section describes detailed description of the APIs that are used for Triggering by SWC.

#### 5.3.13.1 Rte\_Trigger

## Rte\_Trigger

Syntax	void Rte_Trigger_<p>_<o> ([OUT Rte_TransformerError transformerError])		
ServiceID	0x2D		
Parameters (IN/INOUT/OUT)	transformerError	OUT	Optional parameter that contains transformer error that occurred during execution of transformer chain.
Return Values	-		-

**Rte\_Trigger**

Description	The API raises an external trigger of a trigger port. This function is used by user.
-------------	--

**Rte\_Trigger (Queuing support)**

Syntax	Std_ReturnType Rte_Trigger_<p>_<o> (void)		
ServiceID	0x2D		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	RTE_E_OK		Trigger request queued successfully.
	RTE_E_LIMIT		Trigger request not queued as buffer is full.
	RTE_E_OS_ERROR		Error occurred in the OS service invoked from within the Rte_Trigger API.
	RTE_E_UNINIT		API invoked before calling Rte_Start or after calling Rte_Stop.
Description	The API raises an external trigger of a trigger port. This function is used by user.		

**5.3.13.2 Rte\_IrTrigger****Rte\_IrTrigger**

Syntax	void Rte_Trigger_<p>_<o>(void)		
ServiceID	0x2E		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	The API raises an internal trigger to activate Runnable entities of the same software component. This function is used by user.		

**5.3.14 SchM Trigger APIs**

This section describes detailed description of the APIs that are used for Triggering by Basic Software Modules.

**5.3.14.1 SchM\_Trigger**

## SchM\_Trigger

Syntax	void SchM_Trigger_<bsnp>_<name> (void)		
ServiceID	0x09		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	The API raises an external trigger of a trigger port. This function is used by user.		

## SchM\_Trigger (Queuing Support)

Syntax	Std_ReturnType SchM_Trigger_<bsnp>_<name> (void)		
ServiceID	0x09		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	SCHM_E_OK		Trigger request queued successfully.
	SCHM_E_LIMIT		Trigger request not queued as buffer is full.
	SCHM_E_OS_ERROR		Error occurred in the OS service invoked from within the SchM_Trigger API.
	SCHM_E_UNINIT		API invoked before calling SchM_Init or after calling SchM_DeInit.
Description	The API raises an external trigger of a trigger port. This function is used by user.		

### 5.3.14.2 SchM\_ActMainFuntion

## SchM\_ActMainFunction

Syntax	void SchM_ActMainFunction_<bsnp>_<name> (void)		
ServiceID	0x05		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	-		-
Description	The API triggers the activation of BswSchedulableEntity associated with activationPoint of same BSW module. This function is used by user.		

## SchM\_ActMainFunction (Queuing Support)

Syntax	Std_ReturnType SchM_ActMainFunction_<bsnp>_<name> (void)		
--------	--	--	--

## SchM\_ActMainFunction (Queuing Support)

ServiceID	0x05		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	SCHM_E_OK		Trigger request queued successfully.
	SCHM_E_LIMIT		Trigger request not queued as buffer is full.
	SCHM_E_OS_ERROR		Error occurred in the OS service invoked from within the SchM_Trigger API.
	SCHM_E_UNINIT		API invoked before calling SchM_Init or after calling SchM_DeInit.
Description	The API triggers the activation of BswSchedulableEntity associated with activationPoint of same BSW module. This function is used by user.		

## 5.3.15 RTE Calibration APIs

This section provides description of the APIs that are used for Calibration by SWC.

### 5.3.15.1 Rte\_Pim

Rte_Pim			
Syntax	<type>/<return reference>Rte_Pim_<name> ()		
ServiceID	0x1E		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<type>/<return reference>		A typed reference to the per-instance memory
Description	The API provides access to the defined per instance memory of a Software Component. This function is used by user.		

### 5.3.15.2 Rte\_CData

Rte_CData	
Syntax	<return> Rte_CData_<name> ()

**Rte\_CData**

ServiceID	0X1F		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return>		Provide access to the data value of the shared Parameter.
Description	The API provides access to the defined calibration parameter within a software-component. This function is used by user.		

**5.3.15.3 Rte\_Prm****Rte\_Prm**

Syntax	<return> Rte_Prm_<p>_<o> ()		
ServiceID	0x20		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return>		Primitive data <return> is a value and composite <return> is reference.
Description	The API provides access to the defined parameter within a ParameterSwComponentType. This function is used by user.		

**5.3.16 SchM Calibration APIs**

This section provides description of the APIs that are used for Calibration by Basic Software Module.

**5.3.16.1 SchM\_Pim****SchM\_Pim**

Syntax	<return> SchM_Pim_<bsnp>_<name> (void)		
ServiceID	-		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return>		A typed reference to the arTypedPerInstanceMemory
Description	The API provides access to the defined per instance memory of a Basic Software Module. This function is used by user.		



## 5.3.16.2 SchM\_CData

SchM_CData			
Syntax	<return> SchM_CData_<bsnp>_<name> (void)		
ServiceID	-		
Parameters (IN/INOUT/OUT)	-	-	-
Return Values	<return>		provide access to the data value of ParameterDataPrototype in the role perInstanceParameter.
Description	API provides access to the defined calibration parameter of a Basic Software Module defined internally. This function is used by user.		

## 5.3.17 SchM Sender Receiver APIs

This section describes detailed description of the APIs that are used for SchM Sender-Receiver communication.

### 5.3.17.1 SchM\_Send

SchM_Send			
Syntax	Std_ReturnType SchM_Send_<bsnp>[_<vi>_<ai>]_<name>(IN <data>)		
ServiceID	0x0A		
Parameters (IN/INOUT/OUT)	<data>	IN	Data element to write. <data> is passed by value or reference according to the ImplementationDataType.
Return Values	SCHM_E_OK		Data passed to communication the service successfully or the provided Data is Unconnected.
	SCHM_E_INVALID_ARGS		The argument <data> is a NULL pointer.
	SCHM_E_UNINIT		API invoked before calling SchM_Init or after calling SchM_Deinit.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
90 / 146

### SchM\_Send

	SCHM_E_ILLEGAL_INVOCATION	API is called from the wrong context.
	SCHM_E_OS_ERROR	Error occurred in the OS service invoked from within the SchM_Send API.
	SCHM_E_LIMIT	<data> is discarded as receiver's queue is full (intra communication).
Description	Initiates an explicit queued SchM Sender-Receiver transmission of data elements. This function is used by user.	

### 5.3.17.2 SchM\_Receive

### SchM\_Receive

Syntax	Std_ReturnType SchM_Receive_<bsnp>[_<vi>_<ai>]_<name>(OUT <data>)		
ServiceID	0x0B		
Parameters (IN/INOUT/OUT)	<data>	OUT	Data element to pass back the received data. <data> is passed by reference.
Return Values	SCHM_E_OK		Data read successfully.
	SCHM_E_INVALID_ARGS		The argument <data> is a NULL pointer.
	SCM_E_UNINIT		API invoked before calling SchM_Init or after calling SchM_Deinit.
	SCHM_E_ILLEGAL_INVOCATION		API is called from the wrong context.
	SCHM_E_UNCONNECTED		The Required Data is unconnected.
	SCHM_E_NO_DATA		No events (means queued data) were received and no other error occurred while the read was attempted.
	SCHM_E_LOST_DATA		Received data is lost due to receive buffer overflow or due to an error of the underlying communication layers.
Description	Performs an explicit queued read on a SchM Sender-Receiver communication data element. This function is used by user.		

### 5.3.18 SchM Client Server APIs

The section provides detailed description of the APIs that are used for SchM Client-Server communication.

## 5.3.18.1 SchM\_Call

SchM_Call			
Syntax	Std_ReturnType SchM_Call_<bsnp>[_<vi>_<ai>]_<name> ([IN IN/OUT OUT] <data_1>...<data_n>, [OUT <typeOfReturnValue> returnValue])		
ServiceID	0x0C		
Parameters (IN/INOUT/OUT)	<data_1>...<data_n>	[IN IN/OUT OUT]	Argument data of the server operation.
	returnValue	OUT	The OUT parameter returnValue shall only exist if the returnType of BswModuleEntry is not void and the SchM_Call is synchronous and datatype of the OUT parameter returnValue shall be equal to returnType of the called BswModuleEntry.
Return Values	SCHM_E_OK		API call completed successfully.
	SCHM_E_INVALID_ARGS		The argument <data> is a NULL pointer.
	SCHM_E_UNCONNECTED		BswModuleClientServerEntry is unconnected.
	SCHM_E_TIMEOUT		Incoming client server requests before the basic software scheduler is initialized completely or after it is stopped.
	SCHM_E_OS_ERROR		Error occurred in the OS service invoked from within the SchM_Call API.
	SCHM_E_LIMIT		Multiple outstanding asynchronous calls of the same BswModuleEntry. The invocation shall be discarded, the buffers of the return parameters shall not be modified.
Description	Invokes a Client-Server operation between BSW modules, possibly crossing partition boundaries. This function is used by user.		

## 5.3.18.2 SchM\_Result

SchM_Result			
Syntax	Std_ReturnType SchM_Result_<bsnp>[_<vi>_<ai>]_<name> ([IN/OUT OUT <param 1>]... [IN/OUT OUT <paramN>], [OUT <typeOfReturnValue> returnValue])		
ServiceID	0x0D		
Parameters (IN/INOUT/OUT)	<param1>...<paramN>	[IN/OUT OUT]	Result data of the operation.
	returnValue	OUT	The OUT parameter returnValue shall exist if the returnType of BswModuleEntry is different from void. The datatype of the OUT parameter returnValue shall be equal to returnType of the called BswModuleEntry.
Return Values	SCHM_E_OK		API call completed successfully.
	SCHM_E_INVALID_ARGS		The argument <param> is a NULL pointer.
	SCHM_E_UNCONNECTED		BswModuleClientServerEntry is unconnected.
	SCHM_E_TIMEOUT		Incoming client server requests before the basic software scheduler is initialized completely or after it is stopped.
	SCHM_E_NO_DATA		The BswModuleEntry's result is not available, but no other error occurred within the API call or the BswModuleEntry was not called using SchM_Call. The buffers for the IN/OUT and OUT parameters shall not be modified.
Description	Get the result of an asynchronous call of a BswModuleEntry. This function is used by user.		

## 6. Code Generator

This section describes the code generator usage and explains the inputs and outputs.

### 6.1 Inputs

Following Input files are required by Rte Code generator to execute.

- BSW ECU Description File (arxml): This file includes configuration of all Bsw modules. RTE mandatorily

needs ECU configuration for Os, Com, LdCom, Xfrm and NvM.

- ECU Extract File (arxml): This file includes SWC configuration and Communication.
- BSW Module Description File (arxml): This file includes BSW MainFunction timings and exclusive area information needed for SchM generation.
- Os Parameter Definition File (arxml): This file is only needed in case of Multi Partition Scenario.

## 6.2 Output

This section provides information on Output files generated by Rte Code generator.

### 6.2.1 Inc folder

This folder contains all the generated header (.h) files.

File name	Purpose
Rte.h	The RTE header file will define the fixed elements of the RTE that do not need to be generated or configured for each ECU.
Rte_Main.h	The RTE generator will provide the declarations for RTE and SchM Lifecycle APIs.
Rte_<name>.h <name> is the AUTOSAR software component type name.	An application header file will define the RTE API and any associated data structures that are required by the SW-C to use the RTE implementation.
Rte_Type.h	The RTE Types Header File includes the RTE specific type declarations derived from the ImplementationDataTypes created from the definitions of AUTOSAR metamodel classes within the RTE generator's input.
Rte_DataHandleType.h	The RTE Data Handle Types Header File contains the Data Handle type declarations necessary for the component data structures
Rte_<name>_Type.h. <name> is the AUTOSAR software component type name.	The Application Types Header File provides a component local name space for enumeration literals and range values.
Rte_Hook.h	The VFB Tracing Header File defines the configured VFB Trace events
Rte_Cfg.h	The RTE Configuration Header File contains user definitions that affect the behavior of the generated RTE.

# Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
94 / 146

File name	Purpose
SchM_<bsnp>[_<vi>_<ai>]Type.h  Where here <bsnp> is the BSW Scheduler Name. <vi> is the vendorId of the BSW module <ai> is the vendorApilInfix of the BSW module.  The sub part in squared brackets [<vi>_<ai>] is omitted if no vendorApilInfix is the defined for the Basic Software Module.	The Module Interlink Types Header defines specific types related to this basic software module derived either from the input configuration or from the RTE / Basic Software Scheduler implementation
SchM_<bsnp>[_<vi>_<ai>].h  Where here <bsnp> is the BSW Scheduler Name. <vi> is the vendorId of the BSW module <ai> is the vendorApilInfix of the BSW module.  The sub part in squared brackets [<vi>_<ai>] is omitted if no vendorApilInfix is defined for the Basic Software Module.	The Module Interlink Header defines the Basic Software Scheduler API and any associated data structures that are required by the Basic Software Scheduler implementation.
Rte_Internal.h	The Rte Internal defines the macros that is generated by Rte.
Rte_MemMap.h	The Rte MemMap files provides information on memory mapping section used in Header and Source files.
Rte_User.h	The Rte User file contains the declaration of MemCpy. It also contains CheckActiveRunnableAndTask, UpdateActiveTask and ClearActiveTask declaration only when DET is ON or Exclusive Area is configured.
Rte_Static.h	The static file contains information on OS macros and other RTE macros.
Rte_Cbk.h	This file contains the external declaration of Com and LdCom callbacks
Rte_PBCfg.h	This file provides the declarations of PB data structures.

**Table 4: Rte Header Files**

## 6.2.2 Src folder

This folder contains all the generated source (.c and .txt) files.

File name	Purpose
Rte.c	This file will define all the RTE APIs definition and the global variable declaration.
Rte_Info.txt	This file contains following information: <ul style="list-style-type: none"> <li>1. List of Hash Code.</li> <li>2. List of Code MemClass for SWCs.</li> <li>3. List of Linker Memory Sections expected by RTE.</li> </ul>
Rte_User.c	This file provides the definition of MemCpy. It also provides the definition of UpdateActiveTask, ClearActiveTask and CheckActiveRunnableAndTask only when DET is ON or Exclusive Area is configured.
Rte_Partition_<partition>.c	This file will define all the partition specific RTE APIs definition and the global variable declaration. This file will be generated during Multi Partition scenario.
Rte_PBCfg.c	This file provides the definitions of PB data structures.

**Table 5: Rte Source Files**

## 6.2.3 Xml folder

This folder contains all the arxml files generated by RTE Code Generator.

File name	Purpose
Bswmd_Rte.arxml	This file provides version information of the RTE Module.
Os_loc.arxml	This file contains the copy of Os configuration along with IOC configuration generated by RTE Code Generator. This arxml should be used by Os generator as an input. The file is only generated in case of multi partition scenario.

**Table 6: Rte Xml Files**

## 6.3 CLI Checks

The CLI checks are performed on the command line options to ensure correctness of information given in command line. Following are checks to ensure correctness.

1. Proper file name with .arxml extension is not provided in the file input option.
2. ARXML File name contains spaces in the provided file input paths, which is not allowed.
3. File paths are not separated with the spaces in the file input option.
4. Check whether EcucValueCollection module configuration value is provided.
5. Either of the -project\_root, -msn, -input\_files, -out command line option values are not provided.
6. Either of the -project\_root, -mod, -input\_files command line option values are not provided.
7. If -postbuild command line options is not provided, the input files will be treated as RTE input without post-build features.

## 6.4 CLI Usage

This section describes the different command line options supported by Rte code generator. The Command line options are:

**Table 7: Command Line Options**

Command line option	Command Line Value	Description
-project_root	Mandatory	This command line option value will provide the root project path.
-generate	NA	This command line option will be used for generation of files in case contents of input files are same.
-input_files	Mandatory	This command line option value will provide the input file paths for which generation tool need to run (and dependent modules file). ECU Description file and system description file form the input files. In case of Multi Partition the RTE Code Generator also expects Os Parameter Definition File to be passed as input.
-logfile	NA	This command line option will be used for printing error logs of the code generator.
-moduleparams <options>	Mandatory	<i>strictInitValChk</i>



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
97 / 146

Command line option	Command Line Value	Description
		This option enables the strictInitialValueCheck specified by AUTOSAR Specification.
		<i>strictUnconnRPortChk</i>  This option enables the strictUnconnectedRportCheck specified by AUTOSAR Specification.
		<i>suppSchemaVal</i>  This option suppresses the Schema Validation Warning message during RTE Code Generation.
		<i>RTE_COMPILER_ &lt;GHS/TASKING/ARM/GCC&gt;</i>  This option generates compiler specific memory directives in Memmap.h generated by RTE. The linker sections used by RTE in the Memmap.h is also provided in the <i>Rte_Info.txt</i> .
		<i>handleIDPrefix</i>  This option generates the Prefix for Com configuration parameters (Signal/SignalGroup) used by Rte.
		<i>genlocOnly</i>  This option enables generating of only IOC Configuration if applicable in a multi-partition scenario instead of generating IOC alongwith existing OS configuration.
		<i>skipGenSWCMemMap</i>  This option disables generation of <SWC>_MemMap.h for all SwComponents used.
		<i>generateNoASIL</i>  This option disable to generate ASIL_D suffix in the memory

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
98 / 146

Command line option	Command Line Value	Description
		partition. If this options is not involed to operate the <i>Rte</i> , ASIL_D suffix will be added in the generated code
		<i>checkCSTimeout</i> This option enables to check timeout configuration for Inter-Partition Synchronous Client Server Communication
		<i>skipOsTaskSection</i> <i>This option enables to make a compatability with R40 modules. (Before using this option, please contact to AUTOEVER)</i>
-msn	Mandatory	<i>Rte</i> option should be used for RTE Module Generation.
-out	Mandatory	This command line option accepts the folder path to place the generated files. This option value will be output path of the generator where the generated code needs to be placed. This option is mandatory.
-validate	NA	This command line option will be used for validation of the provided input file. Output files will not be generated in this case. If user is not providing this option, then validation and generation will be performed.
-version	NA	This command line option will print the code generator version.
-ignoregeneration	NA	This command line option will be used to ignore the generation if the configuration or the code generator is not present.
help	NA	This command line option will print the help messages for all the above commands.
-genmode <options>	Mandatory	This command line option will be used to support various generation modes in RTE.  Options:  -genmode <b>RteCon</b> : RTE module generates SWC & Type Header files (Rte contract phase)  -genmode <b>SchMCon</b> : RTE module generates SchM header files (SchM contract phase)  -genmode <b>SchMGen</b> : RTE module generates SchM files without

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
99 / 146

Command line option	Command Line Value	Description
		SWC functionality (SchM generation phase)  By default, the RTE generator is in RTE Generation mode where it generates all the .c and .h files required by SWC and BSW. -genmode command is not needed for this mode (Rte generation phase)
-apiargqualifier	NA	Switch to generate argument qualifier type (IN, INOUT, OUT) macros for the RTE API Argument
-postbuild	Optional	This command line option is mandatory and used only if PB variant configured.

For RTE generation following command line inputs are **mandatory**:

1. -project\_root
2. -input\_files
3. -msn
4. -out

Following is an example of providing a command for code generation on command line without multi partition scenario:

**SafeRTE.exe -out E:\WTestApplication\W RTE\_ExArea\WConfig -msn Rte -project\_root E:\WTestApplication\W RTE\_ExArea -input\_files E:\WTestApplication\W RTE\_ExArea\WConfig\Wxml\W RTE\_ExArea.arxml -generate -logfile**

Following is an example of providing a command for code generation on command line with multi partition scenario:

**SafeRTE.exe -out E:\WTestApplication\W RTE\_ExArea\WConfig -project\_root E:\WTestApplication\W RTE\_ExArea -msn Rte -moduleparams RTE\_COMPILER\_<GHS/TASKING/GCC> -input\_files E:\WTestApplication\W RTE\_ExArea\WConfig\Wxml\W RTE\_ExArea.arxml E:\WTestApplication\W OsPDF -generate -logfile**

1. To run the code generation in standalone mode user must specify above command line option on command window.

2. For code generation through make environment, user must enter the following command in command prompt.
  - A. `make -f <makefile> appl = < Application Name> gencfg RTE`
3. Consistency errors and warnings in the ECU description file, will be generated in “*log\_<msn>.txt*” file if -logfile option is provided and it is displayed on the console window. “*log\_<msn>.txt*” file will be placed in the project path.
4. The code generator will stop or show error in the following scenarios.
  - A. ECU Description File name is not provided for -input\_files command line option.
  - B. If BswMd File or System Description or Software Description File is not present at provided file input paths given in -input\_files command.
  - C. If -project\_root, -out, -msn or -input\_files commands are not given in command line.
  - D. If .arxml extension is missing from input configuration files.
  - E. If Configuration Files are not present in the project path.
  - F. If File paths are not separated with spaces.
  - G. If File names contains spaces in the provided input file paths.
  - H. If module consists of unresolved consistency error(s).
  - I. If Module generator is not found.
5. For a Multi Partition scenario, RTE shall be generated first. RTE Code Generator shall generate Os\_loc.arxml which shall be given as an input to OS Generator.

## 6.5 Set CLI and Java Heap Memory in SCons

### 6.5.1 Fixed options

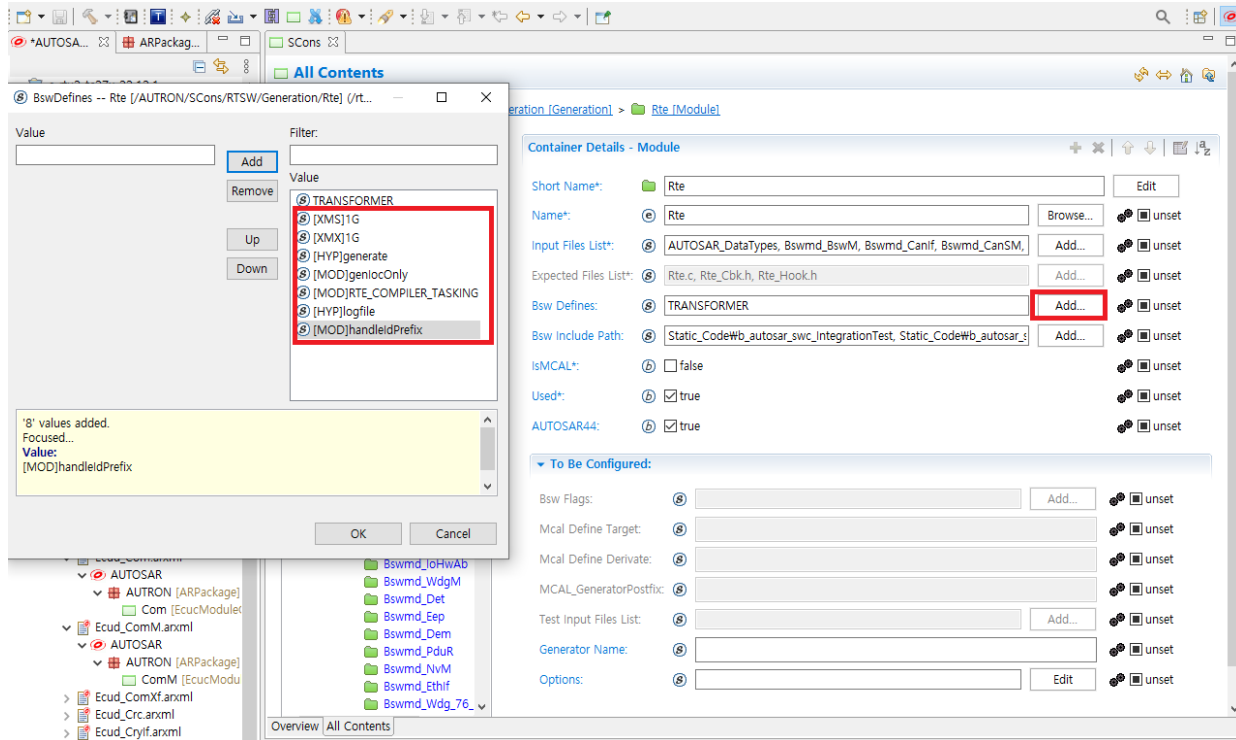
The following options are fixed in the Rte.bat file. It is not recommended to modify these options, but if you need to, you can do so by editing the Rte.bat file directly.

```
-project_root
-inputfiles
-msn
-out
```

### 6.5.2 Modifiable options

Modifiable options can be set in SCons. The following image shows the screen for setting options for the

Rte.bat file in SCons.



**Figure – Set CLI and Java Heap Memory in Scons**

### 6.5.2.1 Prefix

Each option has a prefix associated with it. For example, adding [HYP] generate will create -generate.

In the case of -moduleparams and -genmode, multiple options can be added, so use the prefixes [MOD] and [GEN] respectively.

Finally, use the prefixes [XMS] and [XMX] to set the JAVA Heap Memory.

### 6.5.2.2 [HYP]

[HYP] is used to set options that do not have additional options in CLI Usage. For example, it is used to add -generate and -option.

### 6.5.2.3 [MOD]

[MOD] is used to add -moduleparams. -moduleparams is an option that has additional options, which are added under -moduleparams.

**Example input:**

[MOD]genlocOnly

[MOD]RTE\_COMPILER\_TASKING

[MOD]handleIDPrefix

**Example output:**

-moduleparams

genlocOnly

RTE\_COMPILER\_TASKING

handleIDPrefix

### 6.5.2.4 [GEN]

[GEN] is used to add -genmode. The usage is the same as [MOD], so refer to the [MOD] section.

### 6.5.2.5 [XMS] and [XMX]

[XMS] and [XMX] are used to modify the JAVA Heap Memory options. The default value is 1G for both XMS and XMX.

**Example input:**

[XMS]1G

[XMX]2G

**Example output:**

-Xms1G -Xmx2G

## 7. Debugging and Troubleshooting

This section provides information about RTE feature intended for debugging and troubleshooting. Safe RTE code generator shall include validation checks to ensure that the provided inputs are correct to generate appropriate code. Safe RTE code will be generated only after the validation checks in the input arxml files are corrected. Configuration problems are detected by code generators and they are detailed in the below section.

## 7.1 Code Generator Consistency Checks

This section provides all supported error and warning messages displayed during execution of the code generator. The RTE Code Generator displays error/warning messages when user has configured incorrect inputs. The RTE Code Generator error log consists of the following details:

1. Error Number (SAFERTE\_ERR\_XXXX) or Warning Number (SAFERTE\_WARN\_XXXX)
2. Error Number SAFERTE\_ERR\_MSWXXX for Mode Switch Interface, SAFERTE\_ERR\_CLSXXX for Client Server Interface and SAFERTE\_ERR\_TRGXXX for Trigger Interface.
3. Error Message or Warning Message
4. Object Path
5. Error Number SAFERTE\_ERR\_NVDXXX for Nv Data Interface, SAFERTE\_ERR\_MPCXXX for Multi Partition and SAFERTE\_ERR\_MCXXX for Calibration.
6. Error Number SAFERTE\_ERR\_SRCXXX for Sender Receiver Interface and SAFERTE\_ERR\_XFRXXX for Data Transformation.
7. Error Number SAFERTE\_ERR\_PIMXXX for PortInterfaceMapping.
8. Warning Number SAFERTE\_WARN\_CLSXXX for Client Server Interface, SAFERTE\_WARN\_MCXXX for Calibration, SAFERTE\_WARN\_TRGXXX for Trigger Interface and SAFERTE\_WARN\_PIMXXX for PortInterfaceMapping.
9. Error Number SAFERTE\_ERR\_GEN\_XXXX for input data verification before generator operation.

Given below is an example when RTE Code Generator runs and displays error log:

Module: Rte

Code Generator is loading all the required files...

Code Generator file loading is completed.

Module Generation validation is started for Rte

SAFERTE\_ERR\_0042: NATIVE-DECLARATION should be configured within the SW-BASE-TYPE which is mapped to IMPLEMENTATION-DATA-TYPE.

/Path: /ARPackage0/ImplementationDataType0

RTE Module Configuration(s) consist of unresolved Consistency Error(s).

RTE module Generation is not successful.

## 7.1.1 Common / Generic Error Messages

This section explains Common / Generic errors implemented by RTE Code Generator. The checks also include Sender Receiver checks.

**Table 8: Common / Generic Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_0001	RTE Generator shall throw an error when the configuration parameter or references is not configured in OS module.
2	SAFERTE_ERR_0002	RTE Generator shall throw an error when the configuration parameter or references is not configured in Com or LdCom module.
3	SAFERTE_ERR_0003	RTE Generator shall throw an error when the value of BIT-POSITION is greater than 31.
4	SAFERTE_ERR_0004	RTE Generator shall throw an error when Outer Port and Inner Port of a DelegationSwConnector are connected to different PortPrototypes.
5	SAFERTE_ERR_0005	RTE Generator shall throw an error when P-PORT is connected to P-PORT or R-PORT is connected to R-PORT with an AssemblySwConnector.
6	SAFERTE_ERR_0006	RTE Generator shall throw an error when HANDLE-OUT-OF-RANGE-STATUS is configured for Variable Access in the role DATA-RECEIVE_POINT-BY-VALUE.
7	SAFERTE_ERR_0007	RTE Generator shall throw an error when HANDLE-OUT-OF-RANGE configured within the QUEUED-RECEIVER-COM-SPEC is set to "INVALID".
8	SAFERTE_ERR_0008	RTE Generator shall throw an error when HANDLE-OUT-OF-RANGE configured within the QUEUED-SENDER-COM-SPEC is set to "DEFAULT".
9	SAFERTE_ERR_0009	RTE Generator shall throw an error when HANDLE-OUT-OF-RANGE configured within the QUEUED-SENDER-COM-SPEC is set to "INVALID".
10	SAFERTE_ERR_0010	RTE Generator shall throw an error when HANDLE-OUT-OF-RANGE-STATUS configured within the QUEUED-RECEIVER-COM-SPEC is set to "INDICATE".
11	SAFERTE_ERR_0011	RTE Generator shall throw an error when HANDLE-OUT-OF-RANGE configured within the QUEUED-RECEIVER-COM-SPEC/QUEUED-SENDER-COM-SPEC is set to "EXTERNAL-REPLACEMENT".
12	SAFERTE_ERR_0012	RTE Generator shall throw an error when INIT-VALUE is not configured for the External Replacement Data Element and HANDLE-OUT-OF-RANGE is set to "EXTERNAL-REPLACEMENT".
13	SAFERTE_ERR_0013	RTE Generator shall throw an error, when Activation of runnable entity and wakeup of wait point is combined (i.e. a WaitPoint references a DataReceivedEvent that references a runnable entity).
14	SAFERTE_ERR_0014	RTE Generator shall throw an error when QUEUE-LENGTH attribute configured in QUEUED-RECEIVER-COM-SPEC is less than 1.
15	SAFERTE_ERR_0018	RTE Generator shall throw an error when DATA-SEND-COMPLETED-EVENT that references a RUNNABLE-ENTITY is associated with a WAIT-POINT.
16	SAFERTE_ERR_0020	RTE Generator shall throw an error when RunnableEntity has its attribute CAN-BE-INVOKED-CONCURRENTLY set to "TRUE" and MINIMUM-START-INTERVAL greater than zero.
17	SAFERTE_ERR_0021	RTE Generator shall throw an error, when RUNNABLE-ENTITY has its CAN-BE-INVOKED-CONCURRENTLY attribute set to "FALSE" and mapped to different OS tasks



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
105 / 146

Sl. No.	Error ID	Error Messages
18	SAFERTE_ERR_0023	RTE Generator shall throw an error when both RteUsedOsAlarmRef and RteUsedOsSchTblExpiryPointRef are configured in the RteEventToTaskMapping.
19	SAFERTE_ERR_0024	RTE Generator shall throw an error when RteUsedOsAlarmRef or RteUsedOsSchTblExpiryPointRef is configured and the RTE Event referred is other than TIMING-EVENT or BACKGROUND-EVENT.
20	SAFERTE_ERR_0025	RTE Generator shall throw an error, when OsTask associated with BACKGROUND-EVENT does not have lowest priority on the core.
21	SAFERTE_ERR_0026	RTE Generator shall throw an error when RteEventRef referencing a BACKGROUND-EVENT is mapped to RteUsedOsAlarmRef or RteUsedOsSchTblExpiryPointRef and doesn't have the lowest OsTask priority.
22	SAFERTE_ERR_0030	RTE Generator shall throw an error when Variable Access in the role DATA-RECEIVE-POINT-BY-VALUE references a VARIABLE-DATA-PROTOTYPE whose SW-IMPL-POLICY is set to "QUEUED".
23	SAFERTE_ERR_0031	RTE Generator shall throw an error when the WaitPoint that references a DataReceivedEvent has the reference to VariableDataPrototype with data semantics.
24	SAFERTE_ERR_0032	RTE Generator shall throw an error when VariableAccess in the role dataReadAccess is referenced by a R-PORT or PR -PORT with the enableUpdate attribute enabled.
25	SAFERTE_ERR_0034	RTE Generator shall throw an error when unique component type SYMBOL is not maintained across the components.
26	SAFERTE_ERR_0035	RTE Generator shall throw an error when multiple Runnable Entities share the same entry point.
27	SAFERTE_ERR_0039	RTE Generator shall throw an error when APPLICATION-DATA-TYPE is not mapped to an IMPLEMENTATION-DATA-TYPE.
28	SAFERTE_ERR_0040	RTE Generator shall throw an error when VariableAccess in the role dataReadAccess is referenced by a VARIABLE-DATA-PROTOTYPE whose SW-IMPL-POLICY is set to "QUEUED".
29	SAFERTE_ERR_0041	RTE Generator shall throw an error when SW-IMPL-POLICY is set to "QUEUED" for VARIABLE-DATA-PROTOTYPE referred in INVALIDATION-POLICY.
30	SAFERTE_ERR_0042	RTE Generator shall throw an error when NATIVE-DECLARATION is not configured within the SW-BASE-TYPE which is mapped to IMPLEMENTATION-DATA-TYPE.
31	SAFERTE_ERR_0043	RTE Generator shall throw an error when no IMPLEMENTATION-DATA-TYPE-ELEMENT in the role subElements is configured for the IMPLEMENTATION-DATA-TYPE of category STRUCTURE or ARRAY.
32	SAFERTE_ERR_0044	RTE Generator shall throw an error when DataPrototype that is used in an AtomicSwComponentType has set the ADDITIONAL-NATIVE-TYPE-QUALIFIER attribute to "enum".
33	SAFERTE_ERR_0046	RTE Generator shall throw an error when ARRAY-SIZE is not configured for the IMPLEMENTATION-DATA-TYPE of category ARRAY.
34	SAFERTE_ERR_0047	RTE Generator shall throw an error when attribute SYMBOL/VT element/SHORT-LABEL of the COMPU-SCALE is not configured for the COMPU-METHOD of category TEXTTABLE.
35	SAFERTE_ERR_0048	RTE Generator shall throw an error when attribute SYMBOL/VT element/SHORT-LABEL of the COMPU-SCALE is not configured for the COMPU-METHOD of category SCALE_LINEAR_AND_TEXTTABLE.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
106 / 146

Sl. No.	Error ID	Error Messages
36	SAFERTE_ERR_0049	RTE Generator shall throw an error when attribute SYMBOL/VT element/SHORT-LABEL of the COMPU-SCALE is not configured for the COMPU-METHOD of category SCALE_RATIONAL_AND_TEXTTABLE.
37	SAFERTE_ERR_0050	RTE Generator shall throw an error when attribute SYMBOL/VT element/SHORT-LABEL of the COMPU-SCALE is not configured for the COMPU-METHOD of category BITFIELD_TEXTTABLE.
38	SAFERTE_ERR_0051	RTE Generator shall throw an error when TIME-OUT value of a WaitPoint associated with a DATA-SEND-COMPLETED-EVENT is not same as the corresponding value of TIME-OUT in TRANSMISSION-ACKNOWLEDGEMENT.
39	SAFERTE_ERR_0052	RTE Generator shall throw an error when the SHORT-NAMES of PortInterfaces are not unique within a software component for each set of PPortPrototypes and RPortPrototypes, if the software component supports multiple instantiation or if the indirectAPI attribute is set to "TRUE" for at least one require or provide port.
40	SAFERTE_ERR_0053	RTE Generator shall throw an error when SYMBOL is not configured for the IMPLEMENTATION-DATA-TYPE with equal SHORT-NAME and mapped to different base type.
41	SAFERTE_ERR_0055	RTE Generator shall throw an error when WAIT-POINT is not associated with DataReceivedEvent, DataSendCompletedEvent, ModeSwitchedAckEvent and AsynchronousServerCallReturnsEvent.
42	SAFERTE_ERR_0056	RTE Generator shall throw an error when different RunnableEntities of Software Component mapped to different OsTasks are associated with WAIT-POINT referring to the same DataReceivedEvent.
43	SAFERTE_ERR_0057	RTE Generator shall throw an error when multiple DataReceivedEvents that reference the same VariableDataPrototype with SW-IMPL-POLICY set to "QUEUED" and triggers different runnable entities are mapped to different OsTasks.
44	SAFERTE_ERR_0058	RTE Generator shall throw an error when IMPLEMENTATION-DATA-TYPE of category "ARRAY" whose ARRAY-SIZE-SEMANTICS is not set to "VARIABLE-SIZE" and referenced by a VARIABLE-DATA-PROTOTYPE whose SW-IMPL-POLICY is set to "QUEUED".
45	SAFERTE_ERR_0059	RTE Generator shall throw an error if a dataElement that contains an ImplementationDataType with subElements with arraySizeSemantics equal to variableSize resolves to another type than uint8[n].
46	SAFERTE_ERR_0060	RTE Generator shall throw an error when, IMPLEMENTATION-DATA-TYPE-ELEMENT of category ARRAY and ARRAY-SIZE-SEMANTICS set to "VARIABLE-SIZE" does not have the DYNAMIC-ARRAY-SIZE-PROFILE defined.
47	SAFERTE_ERR_0061	RTE Generator shall throw an error when BswSchedulableEntity is referred in more than one MemorySection in Resource Consumption container.
48	SAFERTE_ERR_0062	RTE Generator shall throw an error when OsTask mapping is not provided for any RteEvent.
49	SAFERTE_ERR_0064	RTE Generator shall throw an error when BswModuleEntry has service arguments (unless SchM_ActivatingEvent is enabled) and return value.
50	SAFERTE_ERR_0065	RTE Generator shall throw an error when I-SIGNALS mapped to SIGNAL-GROUP does not have the same HANDLE-OUT-OF-RANGE values.
51	SAFERTE_ERR_0066	RTE Generator shall throw an error when identical CompuScale Values and symbolic names are configured for the COMPU-METHOD of category TEXTTABLE which is mapped to a VARIABLE-DATAP-ROTOTYPE or IMPLEMENTATION-DATA-TYPE or APPLICATION-PRIMITVE-DATATYPE.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
107 / 146

Sl. No.	Error ID	Error Messages
52	SAFERTE_ERR_0067	RTE Generator shall throw an error when identical CompuScale Values and symbolic names are configured for the COMPU-METHOD of category SCALE_LINEAR_AND_TEXTTABLE which is mapped to an VARIABLE-DATAP-ROTOTYPE or IMPLEMENTATION-DATA-TYPE or APPLICATION-PRIMITVE-DATATYPE.
53	SAFERTE_ERR_0068	RTE Generator shall throw an error when identical CompuScale Values and symbolic names are configured for the COMPU-METHOD of category SCALE_RATIONAL_AND_TEXTTABLE which is mapped to an VARIABLE-DATAP-ROTOTYPE or IMPLEMENTATION-DATA-TYPE or APPLICATION-PRIMITVE-DATATYPE.
54	SAFERTE_ERR_0069	RTE Generator shall throw an error when identical CompuScale Values and symbolic names are configured for the COMPU-METHOD of category BITFIELD_TEXTTABLE which is mapped to a VARIABLE-DATAP-ROTOTYPE or IMPLEMENTATION-DATA-TYPE or APPLICATION-PRIMITVE-DATATYPE.
55	SAFERTE_ERR_0074	RTE Generator shall throw an error when NETWORK-REPRESENTATION-PROPS for the respective I-SIGNAL is not specified and DATA-TYPE-POLICY of an I-SIGNAL is set to "OVERRIDE" or "LEGACY".
56	SAFERTE_ERR_0075	RTE Generator shall throw an error when multiple RUNNABLES are mapped to Basic/Extended Task and RtePositionInTask is not configured and it does not have unique position number.
57	SAFERTE_ERR_0076	RTE Generator shall throw an error when same RTE Event instance which can start a RUNNABLE-ENTITY is referenced to multiple task mappings.
58	SAFERTE_ERR_0077	RTE Generator shall throw an error when INVALIDATION-POLICY is configured and Rte_COMCbKInv_<sn> call back is not configured.
59	SAFERTE_ERR_0078	RTE Generator shall throw an error when TRANSMISSION-ACKNOWLEDGMENT is configured and Rte_COMCbKTack_<sn> call back is not configured.
60	SAFERTE_ERR_0079	RTE Generator shall throw an error when ALIVE-TIMEOUT is configured and Rte_COMCbKRxTOut_<sn> call back is not configured.
61	SAFERTE_ERR_0080	RTE Generator shall throw an error when DataElement of SW-IMPL-POLICY set to "STANDARD" and PORT mapped to System Signal and used in PORT-API-OPTION is not set to "SupportsBufferLocking", when mapped Com I-PDU with ComIPduType is equal to TP.
62	SAFERTE_ERR_0081	RTE Generator shall throw an error when DataElement of SW-IMPL-POLICY set to "STANDARD" and PORT mapped to System Signal and used in PORT-API-OPTION is not set to "SupportsBufferLocking", when mapped LdCom I-PDU with LdComAPIType is equal to TP.
63	SAFERTE_ERR_0082	RTE Generator shall throw an error when INIT-VALUE is not configured and HANDLE-NEVER-RECEIVED is set to "TRUE".
64	SAFERTE_ERR_0083	RTE Generator shall throw an error when section initialization policy is configured in SwAddrMethod and not configured in RteInitializationStrategy.
65	SAFERTE_ERR_0084	RTE Generator shall throw an error when External configuration switch StrictInitialValuesCheck is enabled and SwAddrMethod has a sectionInitializationPolicy set to "INIT" and no INIT-VALUES are not configured either on the sender or receiver side.
66	SAFERTE_ERR_0087	RTE Generator shall throw an error when INVALID-VALUE is not configured neither in VARIABLE-DATA-PROTOTYPE referred in INVALIDATION-POLICY nor in the DATA-TYPE mapped.
67	SAFERTE_ERR_0088	RTE Generator shall throw an error when INIT-VALUE is equal to INVALID-VALUE and HANDLE-INVALID is set to "REPLACE".

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
108 / 146

Sl. No.	Error ID	Error Messages
68	SAFERTE_ERR_0089	RTE Generator shall throw an error when INIT-VALUES are configured for IMPLEMENTATION-DATA-TYPE/ IMPLEMENTATION-DATA-TYPE-ELEMENT of category "UNION".
69	SAFERTE_ERR_0091	RTE Generator shall throw an error when ROOT-SW-COMPOSITION is not configured.
70	SAFERTE_ERR_0092	RTE Generator shall throw an error when multiple COMPOSITION-SW-COMPONENT-TYPE are configured.
71	SAFERTE_ERR_0093	RTE Generator shall throw an error when the number of transformation chains configured is more than 255.
72	SAFERTE_ERR_0094	RTE Generator shall throw an error when ComNotification callback is not configured for Queued Communication.
73	SAFERTE_ERR_0095	RTE Generator shall throw an error when BswSchedulableEntity which references a BswModuleEntry in the role IMPLEMENENTED-ENTRY has IS-RENTRANT attribute set to "FALSE" and this BswSchedulableEntity is mapped to different tasks which can pre-empt each other.
74	SAFERTE_ERR_0096	RTE Generator shall throw an error when PORT associated with Implicit Variable Access is referred in PORT-API-OPTION and INDIRECT-API is set to "TRUE".
75	SAFERTE_ERR_0097	RTE Generator shall throw an error when runnable entity consisting of implicit data access is mapped to different pre-emption areas.
76	SAFERTE_ERR_0098	RTE Generator shall throw an error when all the configured R-PORT are unconnected and StrictRPortCheck is enabled.
77	SAFERTE_ERR_0101	RTE Generator shall throw an error when VARIABLE-DATA-PROTOTYPE referred in ReplaceWith is same as the VARIABLE-DATA-PROTOTYPE referred in ComSpec.
78	SAFERTE_ERR_0102	RTE Generator shall throw an error when configured INIT-VALUES are not within the range.
79	SAFERTE_ERR_0103	RTE Generator shall throw an error when Invalid value is not configured for each data type referred in INVALIDATION-POLICY.
80	SAFERTE_ERR_0104	RTE Generator shall throw an error when Synchronized Runnables and Schedulable Entities are not consistent.
81	SAFERTE_ERR_0105	RTE Generator shall throw an error when the same software component type uses ImplementationDataTypes and ApplicationPrimitiveDataTypes referencing two or more CompuMethods that both contain a CompuScale with a different point range, identical <literalPrefix>es or no literalPrefix and an identical CompuScale symbolic names.
82	SAFERTE_ERR_0106	RTE Generator shall throw an error when RTE-TIMING-EVENTS and other RTE EVENTS are mapped to same Basic Task.
83	SAFERTE_ERR_0107	RTE Generator shall throw an error when PERIOD value configured in RTE-TIMING-EVENTS is less than 0.
84	SAFERTE_ERR_0108	RTE Generator shall throw an error when same OsEvent is not configured in OsTask mapped in RteMappedToTaskRef and RteUsedOsEventRef.
85	SAFERTE_ERR_0112	RTE Generator shall throw an error when External Replacement VARIABLE-DATA-PROTOTYPE is not configured in any one of the components.
86	SAFERTE_ERR_0113	RTE Generator shall throw an error when OsAlarm or ScheduleTable is not configured for the BACKGROUND-EVENT.
87	SAFERTE_ERR_0114	RTE Generator shall throw an error when either OsTask or RteUsedInitFnc is not configured and INIT-EVENT exists.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
109 / 146

Sl. No.	Error ID	Error Messages
88	SAFERTE_ERR_0115	RTE Generator shall throw an error when multiple VariableAccess within a Runnable Entity inside a SW Component consisting of different Implicit and Explicit communication are mapped to same VARIABLE-DATA-PROTOTYPES and PortPrototype.
89	SAFERTE_ERR_0117	RTE Generator shall throw an error when same OsAlarm is configured for different tasks.
90	SAFERTE_ERR_0118	RTE Generator shall throw an error when VARIABLE-DATA-PROTOTYPE referred in ReplaceWith is used in SENDER-RECIEVER-TO-SIGNAL-MAPPING/SENDER-RECIEVER-TO-SIGNAL-GROUP-MAPPING.
91	SAFERTE_ERR_0120	RTE Generator shall throw an error when WaitPoint applied to a RunnableEntity is used by the ExclusiveArea in the role runsInsideExclusiveArea.
92	SAFERTE_ERR_0121	RTE Generator shall throw an error when reference to RTEEvent is not provided in the RteEventToTaskMapping container.
93	SAFERTE_ERR_0122	RTE Generator shall throw an error when reference to BswImplementation is not provided in the RteBswModuleInstance container.
94	SAFERTE_ERR_0123	RTE Generator shall throw an error when reference to BswEvent is not provided in the RteBswEventToTaskMapping container.
95	SAFERTE_ERR_0124	RTE Generator shall throw an error when reference to RteMappedToTaskRef is not provided for RTEEvents other than RteInitEvent in the RteEventToTaskMapping container.
96	SAFERTE_ERR_0125	RTE Generator shall throw an error when reference to RteBswExclusiveAreaOsResourceRef is not provided in the RteBswExclusiveAreaImpl container for the exclusive area having RteExclusiveAreaImplMechanism set to OS_RESOURCE.
97	SAFERTE_ERR_0126	RTE Generator shall throw an error when reference to RteExclusiveAreaOsResourceRef is not provided in the RteExclusiveAreaImplMechanism container for the exclusive area having RteExclusiveAreaImplMechanism set to OS_RESOURCE.
98	SAFERTE_ERR_0127	RTE Generator shall throw an error when reference to VariableDataPrototype is not provided in SenderReceiverToSignalMapping/SenderReceiverToSignalGroupMapping container In the System.
99	SAFERTE_ERR_0128	RTE Generator shall throw an error when Category is not configured for ImplementationDataTypeElement in the ImplementationDataType having category "ARRAY" or "STRUCTURE" or "UNION".
100	SAFERTE_ERR_0129	RTE Generator shall throw an error when reference to RteUsedOsAlarmRef or RteUsedOsSchTblExpiryPointRef is not provided in the RteEventToTaskMapping when TimingEvent is configured in the RTE.
101	SAFERTE_ERR_0130	RTE Generator shall throw an error when reference to RteUsedOsAlarmRef or RteUsedOsSchTblExpiryPointRef is not provided in the RteBswEventToTaskMapping when BswTimingEvent is configured in the RTE.
103	SAFERTE_ERR_0131	RTE Generator shall throw an error when reference to OsAlarmCounterRef is not provided in OsAlarm container and RteEventToTaskMapping is configured.
104	SAFERTE_ERR_0132	RTE Generator shall throw an error when reference to OsScheduleTableCounterRef is not provided in the OsScheduleTable used in RteEventToTaskMapping container.
105	SAFERTE_ERR_0133	RTE Generator shall throw an error when reference to OsAlarmCounterRef is not provided in OsAlarm and RteBswEventToTaskMapping container.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
110 / 146

Sl. No.	Error ID	Error Messages
106	SAFERTE_ERR_0134	RTE Generator shall throw an error when reference to OsScheduleTableCounterRef is not provided in the OsScheduleTable used in RteBswEventToTaskMapping container.
107	SAFERTE_ERR_0135	RTE Generator shall throw an error when OsCounterTick attribute is not configured for OsCounter used by OsAlarm.
108	SAFERTE_ERR_0136	RTE Generator shall throw an error when OsCounterTick attribute is not configured for OsCounter used by ScheduleTable.
109	SAFERTE_ERR_0137	RTE Generator shall throw an error when OsCounterTick attribute is not configured for OsCounter used by OsAlarm in RteBswEventToTaskMapping container.
110	SAFERTE_ERR_0138	RTE Generator shall throw an error when OsCounterTick attribute is not configured for OsCounter used by ScheduleTable in RteBswEventToTaskMapping container.
111	SAFERTE_ERR_0139	RTE Generator shall throw an error when RteSoftwareComponentInstanceRef is not configured in ECU description file.
112	SAFERTE_ERR_0140	RTE Generator shall throw an error when SWC-INTERNAL-BEHAVIOUR is not configured within these components ApplicationSwComponentType, ServiceSwComponentType, ComplexDeviceDriverSwComponentType, EcuAbstractionSwComponentType, SensorActuatorSwComponentType or ServiceProxySwComponentType.
113	SAFERTE_ERR_0141	RTE Generator shall throw an error when ImplementationDataType is configured with category DATA-REFERENCE or FUNCTION-REFERENCE.
114	SAFERTE_ERR_0142	RTE Generator shall throw an error when DataFilter is configured with category ONE_EVERY_N.
115	SAFERTE_ERR_0143	RTE Generator shall throw an error when reference to ImplementationDataType is not provided for ImplementationDataType with category TYPE-REFERENCE.
116	SAFERTE_ERR_0144	RTE Generator shall throw an error when category is not provided for an ImplementationDataType.
117	SAFERTE_ERR_0145	RTE Generator shall throw an error when reference to BaseType is not provided for ImplementationDataTypeElement with category VALUE.
118	SAFERTE_ERR_0146	RTE Generator shall throw an error when reference to TargetDataelement or PortPrototype is not provided in ReceiverComSpec.replaceWith container.
119	SAFERTE_ERR_0147	RTE Generator shall throw an error when InitValue specification does not match with the datatype configured.
120	SAFERTE_ERR_0148	RTE Generator shall throw an error when InitValue specification is not configured for each element of Array/Record DataType.
121	SAFERTE_ERR_0149	RTE Generator shall throw an error when Symbol is not provided for Runnable Entity.
122	SAFERTE_ERR_0150	RTE Generator shall throw an error when the VariableDataPrototype mapped in TargetDataPrototype and VariableDataPrototype mapped in ReceiverComSpec.replaceWith referring to the different Implementation data type.
123	SAFERTE_ERR_0152	RTE Generator shall throw an error when multiple ComSpecs are configured for same PortPrototype and DataElement.
124	SAFERTE_ERR_0153	RTE Generator shall throw an error when contextDataPrototype or rootVariableDataPrototype or autosarVariableInImplDatatype or localVariables are configured for VariableAccess.
125	SAFERTE_ERR_0154	RTE Generator shall throw an error when RunnableEntity using a waitpoint is activated by multiple Timing Events.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
111 / 146

Sl. No.	Error ID	Error Messages
126	SAFERTE_ERR_0155	RTE Generator shall throw an error when the OsEvent referred in RteUsedWaitPointOsEventRef is not referred in the OsTaskEventRef for the OsTask which is mapped to the RunnableEntity via the TimingEvent
127	SAFERTE_ERR_0156	RTE Generator shall throw an error when the OsEvent mapped to the expiry of OsAlarm referred from RteUsedWaitPointTOutOsAlarmRef is not referred in the OsTaskEventRef for the OsTask which is mapped to the RunnableEntity via the TimingEvent.
128	SAFERTE_ERR_0157	RTE Generator shall throw an error when the OsEvent mapped to the expiry of OsAlarm referred from RteUsedWaitPointTOutOsAlarmRef is same as RteUsedWaitPointOsEventRef or RteUsedOsEventRef.
129	SAFERTE_ERR_0158	RTE Generator shall throw an error when the Timeout period of a WaitPoint is greater than the period of the TimingEvent triggering that RunnableEntity.
130	SAFERTE_ERR_0159	RTE Generator shall throw an error when the timeout period of a WaitPoint is configured as zero.
131	SAFERTE_ERR_0160	RTE Generator shall throw an error when the RunnableEntity is configured with multiple WaitPoints.
132	SAFERTE_ERR_0161	RTE Generator shall throw an error when DataReference is not configured in DataReceiveErrorEvent.
133	SAFERTE_ERR_0162	RTE Generator shall throw an error when DataFilters are applied to data types that cannot be interpreted as C language unsigned integer types.
134	SAFERTE_ERR_0163	RTE Generator shall throw an error when OsAlarm referred for waitpoint timeout is not having expiry action as set event.
135	SAFERTE_ERR_0164	RTE Generator shall throw an error when the OsTask mapped to the expiry of OsAlarm referred from RteUsedWaitPointTOutOsAlarmRef is not same as RteMappedToTaskRef which is mapped to the RunnableEntity via the TimingEvent.
136	SAFERTE_ERR_0165	RTE Generator shall throw an error when receiver application is configured for Inter-Intra communication.
137	SAFERTE_ERR_0166	RTE Generator shall throw an error when WaitPoints are configured in Inter-Intra Communication.
138	SAFERTE_ERR_0168	RTE Generator shall throw an error when COMBased or SOMEIP transformer is not configured as the first transformer in the chain on sender side.
139	SAFERTE_ERR_0169	RTE Generator shall throw an error when COMBased or SOMEIP transformer is not configured as the last transformer in the chain on receiver side.
140	SAFERTE_ERR_0171	RTE Generator shall throw an error when bufferProperties are not configured in TransformationTechnology.
141	SAFERTE_ERR_0172	RTE Generator shall throw an error when Xfrm1SignalGroupRefChoice is not configured for the COMBased transformer.
142	SAFERTE_ERR_0173	RTE Generator shall throw an error when compuScaleContents and bufferComputation are not configured in bufferProperties.
143	SAFERTE_ERR_0174	RTE Generator shall throw an error when DYNAMIC-ARRAY-SIZE-PROFILE VSA_FULLY_FLEXIBLE is configured.
144	SAFERTE_ERR_0175	RTE Generator shall throw an error when Implementation Data Type with category Array having a subElement with arraySizeSemantics set to VariableSize is configured.
145	SAFERTE_ERR_0176	RTE Generator shall throw an error when ArraySizeHandling, ArraySizeSemantics and ArraySize is configured for ImplementationDataTypeElement of category ARRAY.
146	SAFERTE_ERR_0177	RTE Generator shall throw an error when HandleOutOfRange is configured for Variable Size Arrays.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
112 / 146

Sl. No.	Error ID	Error Messages
147	SAFERTE_ERR_0178	RTE Generator shall throw an error when Data Conversion is configured for Variable Size Arrays.
148	SAFERTE_ERR_0179	RTE Generator shall throw an error when HandleOutOfRange is configured in Inter-Intra communication.
149	SAFERTE_ERR_0180	RTE Generator shall throw an error when EXTERNAL_REPLACEMENT for HandleOutOfRange and Invalidation is configured.
150	SAFERTE_ERR_0182	RTE Generator shall throw an error when INVALID and DEFAULT HandleOutOfRange policy is configured for Queued Inter-ECU Communication.
151	SAFERTE_ERR_0183	RTE Generator shall throw an error when Initial values configured in Provided ComSpecs and Receiver ComSpecs are not same in case of 1: N and N:1 communication.
152	SAFERTE_ERR_0184	RTE Generator shall throw an error when the variableAccess for the same VDP referred in RteVariableReadAccessRef/RteVariableWriteAccessRef in RteImplicitCommunication container are not mapped to same OSTask.
153	SAFERTE_ERR_0185	RTE Generator shall throw an error when RteCoherency parameter not set to true, if RteImplicitCommunication container is configured.
154	SAFERTE_ERR_0186	RTE Generator shall throw an error when two or more RunnableEntities that own either dataSendPoint or dataWriteAccess point to the same VariableDataPrototype and PortPrototype pair where attribute TransmissionAcknowledgementRequest exist for SenderComSpec and same/different portPrototype is referenced by dataSendPoints or dataWriteAccesses.
155	SAFERTE_ERR_0187	RTE Generator shall throw an error when DATA-RECEIVE-ERROR-EVENT is configured on receiver and INVALIDATION-POLICY is set to 'REPLACE' or 'EXTERNAL-REPLACEMENT' in case of Intra-ECU communication.
156	SAFERTE_ERR_0188	RTE Generator shall throw an error when Different HandleOutOfRange Policy is configured for ISignals in case of Fan-In or Fan-Out Communication.
157	SAFERTE_ERR_0191	RTE Generator shall throw an error when Initial value is specified, and the implementation policy is set to QUEUED.
158	SAFERTE_ERR_0192	RTE Generator shall throw an error when BswModuleEntity which is not a BswSchedulableEntity uses ExclusiveArea in the role runsInsideExclusiveArea.
159	SAFERTE_ERR_0193	RTE Generator shall throw an error when the value of UPPER-LIMIT and LOWER-LIMIT is not same for the COMPU-METHOD of CATEGORY (TEXTTABLE, SCALE_LINEAR_AND_TEXTTABLE, BITFIELD_TEXTTABLE, SCALE_RATIONAL_AND_TEXTTABLE).
160	SAFERTE_ERR_0194	RTE Generator shall throw an error when an LdComIPdu is mapped to two or more SWC using one VariableDataPrototype or a SWC and VDP pair is mapped to multiple LdComIPdu's.
161	SAFERTE_ERR_0195	RTE Generator shall throw an error when Provided/Requester ComSpec container is not configured in PORT-PROTOTYPE.
162	SAFERTE_ERR_0196	RTE Generator shall throw an error when Data Element reference is not configured in PortPrototype.
163	SAFERTE_ERR_0197	RTE Generator shall throw an error when Provided/Required Interface reference is not configured in PortPrototype.
164	SAFERTE_ERR_0198	RTE Generator shall throw an error when supportsMultipleInstantiation attribute is set to TRUE.
165	SAFERTE_ERR_0199	RTE Generator shall throw an error when PORTPROTOTYPE and Data Element reference is not provided in the Variable Access.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
113 / 146

Sl. No.	Error ID	Error Messages
166	SAFERTE_ERR_0201	RTE Generator shall throw an error when PRPortProtoType is set to QUEUED Communication.
167	SAFERTE_ERR_0202	RTE Generator shall throw an error when shortName configured in IMPLEMENTATION-DATATYPE-ELEMENT is same as the StandardPlatformDataTypes.
168	SAFERTE_ERR_0203	RTE Generator shall throw an error when OsResource mapped to an ExclusiveArea using the RteExclusiveAreaImplMechanism container is not mapped to the OsTask mapped to the Runnable Entity using the ExclusiveArea.
169	SAFERTE_ERR_0204	RTE Generator shall throw an error when SystemSignal reference is not configured in ISignal.
170	SAFERTE_ERR_0205	RTE Generator shall throw an error when SystemSignalGroup reference is not configured in ISignalGroup.
171	SAFERTE_ERR_0206	RTE Generator shall throw an error when attribute VT value of the COMPU-SCALE is not set to valid C identifier for the COMPU-METHOD of category TEXTTABLE.
172	SAFERTE_ERR_0207	RTE Generator shall throw an error when attribute VT value of the COMPU-SCALE is not set to valid C identifier for the COMPU-METHOD of category SCALE_LINEAR_AND_TEXTTABLE.
173	SAFERTE_ERR_0208	RTE Generator shall throw an error when attribute VT value of the COMPU-SCALE is not set to valid C identifier for the COMPU-METHOD of category SCALE_RATIONAL_AND_TEXTTABLE.
174	SAFERTE_ERR_0209	RTE Generator shall throw an error when attribute VT value of the COMPU-SCALE is not set to valid C identifier for the COMPU-METHOD of category BITFIELD_TEXTTABLE.
175	SAFERTE_ERR_0210	RTE Generator shall throw an error when M: N sender receiver communication is configured for a VariableDataPrototype.
176	SAFERTE_ERR_0211	RTE Generator shall throw an error when TRANSFORMATION-TECHNOLOGY is not referred in TRANSFORMATION-I-SIGNAL-PROPS of I-SIGNAL.
177	SAFERTE_ERR_0212	RTE Generator shall throw an error when XfrmTransformationTechnologyRef is not configured in XfrmImplementationMapping.
178	SAFERTE_ERR_0213	RTE Generator shall throw an error when XfrmSignal or XfrmSignalChoice is not configured in XfrmImplementationMapping.
179	SAFERTE_ERR_0214	RTE Generator shall throw an error when XfrmInvTransformerBswModuleEntryRef is not configured on receiver side transformer.
180	SAFERTE_ERR_0215	RTE Generator shall throw an error when XfrmTransformerBswModuleEntryRef is not configured on sender side transformer.
181	SAFERTE_ERR_0216	RTE Generator shall throw an error when XfrmISignalRefChoice is not configured for the Somelp transformer.
182	SAFERTE_ERR_0218	RTE Generator shall throw an error when Activation Reason BitPosition is not unique.
183	SAFERTE_ERR_0219	RTE Generator shall throw an error when RTEEvent that is referenced by a WaitPoint in the role trigger references ExecutableEntityActivationReason.
184	SAFERTE_ERR_0220	RTE Generator shall throw an error when Events mapped to the runnable have not configured ExecutableEntityActivationReason and ExecutableEntityActivationReason is configured for runnable.
185	SAFERTE_ERR_0221	RTE Generator shall throw an error when Variable Access in the role DATA-RECEIVE-POINT-BY-VALUE refer to Composite Data Types.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
114 / 146

Sl. No.	Error ID	Error Messages
186	SAFERTE_ERR_0222	RTE Generator shall throw an error when ImplementedEntry is not provided in the BswModuleEntity container.
187	SAFERTE_ERR_0223	RTE Generator shall throw an error when SenderReceiverCompositeElementToSignalMapping is configured.
188	SAFERTE_ERR_0224	RTE Generator shall throw an error when Composite DataTypes are mapped to SenderReceiverToSignalMapping and Primitive DataTypes are mapped to SenderReceiverToSignalGroupMapping and COM is configured.
189	SAFERTE_ERR_0225	RTE Generator shall throw an error when Primitive DataTypes are mapped to LdCom and transformers are not configured.
190	SAFERTE_ERR_0226	RTE Generator shall throw an error when Ports are connected by incompatible interfaces.
191	SAFERTE_ERR_0228	RTE Generator shall throw an error when BswExclusiveAreaImplMechanism is not configured for each exclusive area.
192	SAFERTE_ERR_0229	RTE Generator shall throw an error when RteExclusiveAreaImplMechanism is not configured for each exclusive area.
193	SAFERTE_ERR_0230	RTE Generator shall throw an error when BswBehavior reference is not configured for BswImplementation.
194	SAFERTE_ERR_0232	RTE Generator shall throw an error when combination of Port and VariableDataPrototype referred from DataReceiveErrorEvent does not match the ComSpec configuration.
195	SAFERTE_ERR_0233	RTE Generator shall throw an error when combination of Port and VariableDataPrototype referred from DataReceivedEvent does not match the ComSpec configuration.
196	SAFERTE_ERR_0234	RTE Generator shall throw an error when System Configuration Reference from EcucValueCollection is missing or reference not found in inputs.
197	SAFERTE_ERR_0235	RTE Generator shall throw an error when SenderRecArrayElementMapping configuration is either missing or incomplete.
198	SAFERTE_ERR_0237	RTE Generator shall throw an error when configured LowerLimit/UpperLimit value is beyond the physical range of a dataType OR one of the dataTypeElements for the corresponding Data Prototype.
199	SAFERTE_ERR_0238	RTE Generator shall throw an error when timing events with non-multiple period of each other are not mapped to different Basic tasks.
200	SAFERTE_ERR_0239	RTE Generator shall throw an error when DataTransformation is not referred in the role dataTransformation inside ISignal handled by LdCom with Transformer.
201	SAFERTE_ERR_0240	RTE Generator shall throw an error when RunnableEntity for which WaitPoints are configured the attribute CAN-BE-INVOKED-CONCURRENTLY is set to TRUE.
202	SAFERTE_ERR_0241	RTE Generator shall throw an error when multiple server operations are invoked by a single client component request (1: N communication where n > 1).
203	SAFERTE_ERR_0242	RTE Generator shall throw an error when in Client Server communication, if multiple operations are mapped to the same RunnableEntity and requires a call serialization, the operation invoked events are not mapped to same task and do not have the same position in the task.
204	SAFERTE_ERR_0243	RTE Generator shall throw an error when eventSource reference is not provided in the RTEEvent.
205	SAFERTE_ERR_0244	RTE Generator shall throw an error when the server task does not have the higher priority than the client task in case of Synchronous Client Server communication unless timeout is configured.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
115 / 146

Sl. No.	Error ID	Error Messages
206	SAFERTE_ERR_0246	RTE Generator shall throw an error when same timeout values are not specified for the AsynchronousServerCallPoint and for the WaitPoint associated with the AsynchronousServerCallReturnsEvent for this AsynchronousServerCallPoint.
207	SAFERTE_ERR_0247	RTE Generator shall throw an error when QueueLength attribute of a ServerComSpec is not greater than 0.
208	SAFERTE_ERR_0248	RTE Generator shall throw an error when AsynchronousServerCallReturnsEvent that references a runnable entity is referenced by a WaitPoint.
209	SAFERTE_ERR_0251	RTE Generator shall throw an error when ROperationInAtomicSwcInstanceRef is not configured in ServerCallPoint.
210	SAFERTE_ERR_0252	RTE Generator shall throw an error when Target Required Operation is not configured in ServerCallPoint.
211	SAFERTE_ERR_0253	RTE Generator shall throw an error when Context <u>Rport</u> is not configured in ServerCallPoint.
212	SAFERTE_ERR_0254	RTE Generator shall throw an error when ClientServerOperation in a PPortPrototype is defined but no RunnableEntity is triggered by an OperationInvokedEvent that references the ClientServerOperation.
213	SAFERTE_ERR_0255	RTE Generator shall throw an error when ClientServerOperations triggering the same RunnableEntity are not compatible in terms of PortDefinedArguments, Possible Errors, ArgumentDataPrototypes and if ArgumentDataPrototype is of type Composite then referred ImplementationDataType is not same for all the operations.
214	SAFERTE_ERR_0256	RTE Generator shall throw an error when Operation is not referred in ClientComSpec.
215	SAFERTE_ERR_0257	RTE Generator shall throw an error when Operation is not referred in ServerComSpec.
216	SAFERTE_ERR_0258	RTE Generator shall throw an error when Type is not configured for ArgumentDataPrototype in ClientServerOperation.
217	SAFERTE_ERR_0259	RTE Generator shall throw an error when ApplicationErrors do not have value between 1 to 63.
218	SAFERTE_ERR_0260	RTE Generator shall throw an error when AsynchronousServerCallResultPoint is not configured in case of Asynchronous Client Server Communication.
219	SAFERTE_ERR_0262	RTE Generator shall throw an error when both Synchronous and Asynchronous ServerCallPoints for a given ClientServerOperation are configured.
220	SAFERTE_ERR_0263	RTE Generator shall throw an error when datatype configured is other than PrimitiveDataType for PortDefinedArgumentValues.
221	SAFERTE_ERR_0264	RTE Generator shall throw an error when ValueType is not referred in PortDefinedArgumentValues.
222	SAFERTE_ERR_0265	RTE Generator shall throw an error when Init value is not configured for the applicable data Prototype.
223	SAFERTE_ERR_0266	RTE Generator shall throw an error when SwComponentType has PortPrototypes typed by different PortInterfaces with equal short name but conflicting ApplicationErrors.
224	SAFERTE_ERR_0267	RTE Generator shall throw an error when startOnEvent is not configured for OperationInvokedEvent.
225	SAFERTE_ERR_0269	RTE Generator shall throw an error when ServerArgumentImplPolicy uses void in case of primitive IN arguments.
226	SAFERTE_ERR_0270	RTE Generator shall throw an error when WaitPoints configured for Server runnable are invoked via synchronous client server communication.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
116 / 146

Sl. No.	Error ID	Error Messages
227	SAFERTE_ERR_0271	RTE Generator shall throw an error when multiple operation invoked events are mapped to same task and are invoking different runnable entities and having same position in the task configured.
228	SAFERTE_ERR_0272	RTE Generator shall throw an error when different timeout values in ServerCallPoints are configured referencing the same instance of ClientServerOperation in a RPortPrototype.
229	SAFERTE_ERR_0275	RTE Generator shall throw an error when Local variable is not configured in the InterRunnableVariable VariableAccess.
230	SAFERTE_ERR_0276	RTE Generator shall throw an error when AutosarDataType is not referred in InterRunnableVariable.
231	SAFERTE_ERR_0277	RTE Generator shall throw an error when init value is not configured for InterRunnableVariable.
232	SAFERTE_ERR_0278	RTE Generator shall throw an error when Rte_Receive API does not have Non-Blocking characteristics and executeDespiteDataUnavailability is enabled in DataTransformation.
233	SAFERTE_ERR_0279	RTE Generator shall throw an error when executeDespiteDataUnavailability in DataTransformation is not set to true for a transformer chain using E2E.
234	SAFERTE_ERR_0280	RTE Generator shall throw an error when ImplementationDataTypeElement with category UNION is not configured for Wrapped Union Data type.
235	SAFERTE_ERR_0281	RTE Generator shall throw an error when ImplementationDataType configured for Wrapped Union is configured with more than two ImplementationDataTypeElements.
236	SAFERTE_ERR_0282	RTE Generator shall throw an error when ImplementationDataType is configured with Category UNION.
237	SAFERTE_ERR_0283	RTE Generator shall throw an error when initValue of the MemberSelector element of the Wrapped Union data type is not configured as 1.
238	SAFERTE_ERR_0284	RTE Generator shall throw an error when "memberSelector" with the category TYPE_REFERENCE does not points to category VALUE for Wrapped Union data type.
239	SAFERTE_ERR_0285	RTE Generator shall throw an error when TYPE_REFERENCE is configured as ImplementationDataTypeElement for "payload" of a Wrapped Union Data Type.
240	SAFERTE_ERR_0286	RTE Generator shall throw an error when ArraySizeSemantics is not set to VARIABLE_SIZE or ArraySize is not configured for subElement of ImplementationDataTypeElement of category ARRAY for VSA_LINEAR.
241	SAFERTE_ERR_0287	RTE Generator shall throw an error when SOMEIP Transformer is not configured and dataType used is either VSA_SQUARE or VSA_RECTANGULAR.
242	SAFERTE_ERR_0288	RTE Generator shall throw an error when ArraySizeHandling is not set to INHERITED_FROM_ARRAY_ELEMENT_TYPE_SIZE, or category is not ARRAY, or ArraySizeSemanticsEnum is not set to VARIABLE_SIZE or ArraySize is configured for intermediate ImplementationDataTypeElement of category ARRAY for VSA_SQUARE.
243	SAFERTE_ERR_0289	RTE Generator shall throw an error ArraySizeSemantics is not set to VARIABLE_SIZE or ArraySize is not configured for the terminating ImplementationDataTypeElement for VSA_SQUARE.
244	SAFERTE_ERR_0290	RTE Generator shall throw an error when ArraySizeHandling is not set to ALL_INDICES_SAME_ARRAY_SIZE, or category is not ARRAY, or ArraySizeSemanticsEnum is not set to VARIABLE_SIZE or ArraySize is not configured for intermediate ImplementationDataTypeElement of category ARRAY for VSA_RECTANGULAR.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
117 / 146

Sl. No.	Error ID	Error Messages
246	SAFERTE_ERR_0291	RTE Generator shall throw an error when ArraySizeSemantics is not set to VARIABLE_SIZE or ArraySize is not configured for the terminating ImplementationDataTypeElement for VSA_RECTANGULAR.
247	SAFERTE_ERR_0292	RTE Generator shall throw an error when a transformer chain consists of multiple serialLizer or safety transformers.
248	SAFERTE_ERR_0293	RTE Generator shall throw an error when Rte_IStatus is not configured, PortAPIOption is configured and transformerErrorHandling is set to TRANSFORMER-ERROR-HANDLING.
249	SAFERTE_ERR_0294	RTE Generator shall throw an error when Ticks configured for OsAlarm/OsScheduleTable does not have Nonzero value, if RteEventToTaskMapping is configured.
250	SAFERTE_ERR_0295	RTE Generator shall throw an error when Ticks configured for OsAlarm/OsScheduleTable does not have Nonzero value, if RteBswEventToTaskMapping is configured
251	SAFERTE_ERR_0296	RTE Generator shall throw an error when RunnableEntities/RteEventToTaskMappings mapped to the same basic task are not mapped to a single ScheduleTableExpiryPoint and mapped to a Schedule Table.
252	SAFERTE_ERR_0297	RTE Generator shall throw an error when any Extended Task activated via a ScheduleTableExpiryPoint is not referred using OsScheduleTableEventSetting and the OsEvent, OsTask referred in both OsScheduleTableEventSetting and RteEventToTaskMapping are not same.
253	SAFERTE_ERR_0298	RTE Generator shall throw an error when any Basic Task activated via a ScheduleTableExpiryPoint is not referred using OsScheduleTableTaskActivation and the OsTask referred in both OsScheduleTableTaskActivation and RteEventToTaskMapping is not same.
254	SAFERTE_ERR_0300	RTE Generator shall throw an error when OsScheduleTableActivateTaskRef is not provided in OsScheduleTableTaskActivation and RteUsedOsSchTblExpiryPointRef reference is configured in RteEventToTaskMapping for basic task.
256	SAFERTE_ERR_0301	RTE Generator shall throw an error when OsScheduleTableSetEventRef and OsScheduleTableSetEventTaskRef is not provided in OsScheduleTableEventSetting and RteUsedOsSchTblExpiryPointRef reference is configured in RteEventToTaskMapping for extended task.
257	SAFERTE_ERR_0302	RTE Generator shall throw an error when SystemSignal/SystemSignalGroup reference is not provided in SenderReceieverToSignalMapping/SenderReceiverToSignalGroupMapping.
258	SAFERTE_ERR_0303	RTE Generator shall throw an error when SchedulableEntities/RteBswEventToTaskMappings mapped to the same basic task are not mapped to a single ScheduleTableExpiryPoint and mapped to a Schedule Table.
259	SAFERTE_ERR_0304	RTE Generator shall throw an error when any Extended Task activated via a ScheduleTableExpiryPoint is not referred using OsScheduleTableEventSetting and the OsEvent, OsTask referred in both OsScheduleTableEventSetting and RteBswEventToTaskMapping are not same.
260	SAFERTE_ERR_0305	RTE Generator shall throw an error when any Basic Task activated via a ScheduleTableExpiryPoint is not referred using OsScheduleTableTaskActivation and the OsTask referred in both OsScheduleTableTaskActivation and RteBswEventToTaskMapping is not same.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
118 / 146

Sl. No.	Error ID	Error Messages
261	SAFERTE_ERR_0307	RTE Generator shall throw an error when RteExpectedActivationOffset and RteExpectedTickDuration are not provided in RteUsedOsActivation and RteOsInteraction container is configured.
262	SAFERTE_ERR_0308	RTE Generator shall throw an error when OsScheduleTableActivateTaskRef is not provided in OsScheduleTableTaskActivation and RteUsedOsSchTblExpiryPointRef reference is configured in RteBswEventToTaskMapping for basic task.
263	SAFERTE_ERR_0309	RTE Generator shall throw an error when OsScheduleTableSetEventRef and OsScheduleTableSetEventTaskRef is not provided in OsScheduleTableEventSetting and RteUsedOsSchTblExpiryPointRef reference is configured in RteBswEventToTaskMapping for extended task.
264	SAFERTE_ERR_0310	RTE Generator shall throw an error when Event configured in the component is not configured in RteEventToTaskMapping Container.
265	SAFERTE_ERR_0311	RTE Generator shall throw an error when reference to trigger is not given for WaitPoints configured.
266	SAFERTE_ERR_0312	RTE Generator shall throw an error when SwBaseType reference is not provided for ImplementationDataType of category VALUE and non-Platform Type shortName.
267	SAFERTE_ERR_0313	RTE Generator shall throw an error when LdComApiType and LdComIPduDirection is not configured in LdComIPdu and LdCom is configured.
268	SAFERTE_ERR_0314	RTE Generator shall throw an error when CompuMethod of category LINEAR is configured for ImplementationDataType.
269	SAFERTE_ERR_0315	RTE Generator shall throw an error when ApplicationRecordElement /ApplicationArrayElement is not configured for composite application datatypes and reference is not provided for ApplicationRecordElement and ApplicationArrayElement under the parameter type.
270	SAFERTE_ERR_0316	RTE Generator shall throw an error when ApplicationDataType is not mapped to ImplementationDataType in dataMapping.
271	SAFERTE_ERR_0317	RTE Generator shall throw an error when MaxNumberOfElements is not configured for ApplicationArrayElement or when the MaxNumberOfElements is not matched with the mapped implementationDataType Element.
272	SAFERTE_ERR_0318	RTE Generator shall throw an error when the compu methods of category TEXTTABLE and SCALE_LINEAR_TEXTTABLE are provided only on the network side.
273	SAFERTE_ERR_0319	RTE Generator shall throw an error when DataConversion is configured for following scenarios: 1. VDP configured with Data Invalidation and Data Transformation 2. Transformation of Data with type Record and 2D array 3. LdCom without Transformation
274	SAFERTE_ERR_0320	RTE Generator shall throw an error when CompuMethod is configured as a combination of LINEAR and TEXTTABLE.
278	SAFERTE_ERR_0321	RTE Generator shall throw an error when CompuScale Point ranges are not matched with the range provided in mapping under PortInterfaceMapping.
279	SAFERTE_ERR_0322	RTE Generator shall throw an error when for Dynamic DataType, ComSignalType is not configured as UINT8_DYN.
280	SAFERTE_ERR_0323	RTE Generator shall throw an error when RteVfbTraceEnabled is not configured.
281	SAFERTE_ERR_0325	RTE Generator shall throw an error when Data Conversion is configured for Inter Intra communication.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
119 / 146

Sl. No.	Error ID	Error Messages
282	SAFERTE_ERR_0326	RTE Generator shall throw an error when the Sub Element shortName of actual data type is not same as Sub Element shortName of network data type when PortInterfaceMapping is involved.
283	SAFERTE_ERR_0327	RTE Generator shall throw an error EcucPartitionQmBswModuleExecution, EcucPartitionBswModuleExecution and PartitionCanBeRestarted parameters are not configured and partition is involved.
284	SAFERTE_ERR_0328	RTE Generator shall throw an error when the value of V0 of bufferComputation of a TransformationTechnology which has the protocol attribute set to COMBased does not have the same value as the length attribute of the ISignalIPdu to which the ISignalGroup is mapped.
285	SAFERTE_ERR_0329	RTE Generator shall throw an error when BswEvent configured in the BswInternalBehavior is not configured in RteBswEventToTaskMapping Container.
286	SAFERTE_ERR_0330	RTE Generator shall throw an error when RTEEvent is not mapped to a basic OsTask referred in RteMappedToTaskRef and Virtual Task Mapping is present for it.
288	SAFERTE_ERR_0331	RTE Generator shall throw an error when more than one RTEEvent are mapped to the OsTask referred in RteMappedToTaskRef and Virtual Task Mapping is present for it.
289	SAFERTE_ERR_0332	RTE Generator shall throw an error when the priority of the Task referred in RteMappedToTaskRef is less than the task referred in RteVirtuallyMappedToTaskRef.
290	SAFERTE_ERR_0334	RTE Generator shall throw an error when VirtualTaskMapping is configured for BackGroundEvent/OperationInvokedEvent.
291	SAFERTE_ERR_0335	RTE Generator shall throw an error when Symbol for AtomicSoftwareComponent is not configured and SymbolProps is configured.
292	SAFERTE_ERR_0336	RTE Generator shall throw an error when StartsOnEvent is not configured in BswEvent.
293	SAFERTE_ERR_0337	RTE Generator shall throw an error when either of the RteActivationOsAlarmRef or RteActivationOsSchTblRef and RteActivationOsTaskRef is not configured and RteUsedOsActivation is configured.
294	SAFERTE_ERR_0338	RTE Generator shall throw an error when AUTOSAR version schema of file is below 4.0.3. Minimum supported version is 4.0.3.
295	SAFERTE_ERR_XFR001	RTE Generator shall throw an error when ISignal is referred from Xfrm ECU configuration but does not have any DataTransformations configured.
296	SAFRTE_ERR_XFR002	RTE Generator shall throw an error when the value of BufferProperties.headerLength is not 0 and ComXf and E2EXf transformer chain is involved.
297	SAFERTE_ERR_0339	RTE Generator shall throw an error when Runnable Entity with WaitPoint is mapped to multiple RTEEvents.
298	SAFERTE_ERR_0340	RTE Generator shall throw an error when RteUsedWaitPointTOutOsAlarmRef and RteUsedWaitPointOsEventRef are not configured and WaitPoint is configured for a RunnableEntity.
299	SAFERTE_ERR_0341	RTE Generator shall throw an error when the category of the ImplementationDataTypeElement or ImplementationDataType referring an ImplementationDataType through SwDataDefProps is not set to TYPE_REFERENCE.
300	SAFERTE_ERR_0342	RTE Generator shall throw an error when ComTimeout Parameter value is not non-zero and ALIVE-TIMEOUT is configured.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
120 / 146

Sl. No.	Error ID	Error Messages
301	SAFERTE_ERR_0343	RTE Generator shall throw an error when OsCounterType is not configured.
302	SAFERTE_ERR_0344	RTE Generator shall throw an error when ContextComponent Type referred in DataMapping is not mapped to either Software Composition configured in System container or AtomicSwComponentType.
303	SAFERTE_ERR_0345	RTE Generator shall throw an error when ContextPort referred in DataMapping is not referred from either Software Composition configured in System container or from AtomicSwComponentType.
304	SAFERTE_ERR_0346	RTE Generator shall throw an error when RteSwComponentInstance is not configured for every SwComponentPrototype in the CompositionSwComponentType except ParameterSwComponentType.
305	SAFERTE_ERR_0348	RTE Generator shall throw an error when for an RteEventToTaskMapping using RteUsedOsAlarmRef, the extended OsTask and the OsEvent from OsAlarm.OsAlarmAction.OsAlarmSetEvent do not match the OsTask and OsEvent configured in RteEventToTaskMapping.RteMappedToTaskRef and RteEventToTaskMapping.RteUsedOsEventRef respectively.
306	SAFERTE_ERR_0349	RTE Generator shall throw an error when for an RteEventToTaskMapping using RteUsedOsAlarmRef, the basic OsTask from OsAlarm.OsAlarmAction.OsAlarmActivateTask does not match the OsTask configured in RteEventToTaskMapping.RteMappedToTaskRef.
307	SAFERTE_ERR_0350	RTE Generator shall throw an error when for an RteBswEventToTaskMapping using RteBswUsedOsAlarmRef, the extended OsTask and the OsEvent from OsAlarm.OsAlarmAction.OsAlarmSetEvent do not match the OsTask and OsEvent configured in RteBswEventToTaskMapping.RteBswMappedToTaskRef and RteBswEventToTaskMapping.RteBswUsedOsEventRef respectively.
308	SAFERTE_ERR_0351	RTE Generator shall throw an error when for a RteBswEventToTaskMapping using RteBswUsedOsAlarmRef, the basic OsTask from OsAlarm.OsAlarmAction.OsAlarmActivateTask does not match the OsTask configured in RteBswEventToTaskMapping.RteBswMappedToTaskRef.
309	SAFERTE_ERR_0352	RTE Generator shall throw an error when for a SwcInternalBehavior, PortPrototype reference is not provided in PortAPIOption or PortPrototype contains proxy.
310	SAFERTE_ERR_0353	RTE Generator shall throw an error when same OsEvent is not configured in OsTask mapped to RteBswMappedToTaskRef and RteBswUsedOsEventRef.
311	SAFERTE_ERR_0354	RTE Generator shall throw an error when RteBswUsedOsEventRef is empty and OsEvent is mapped in OsTaskEventRefs of OsTask container.
312	SAFERTE_ERR_0355	RTE Generator shall throw an error when value is not configured for NumericalValueSpecification.
313	SAFERTE_ERR_0356	RTE Generator shall throw an error when ComNotification name does not contain the name of its particular ComSignalGroup name in it.
314	SAFERTE_ERR_0357	RTE Generator shall throw an error when ComNotification name does not contain the name of its particular ComSignal name in it.
315	SAFERTE_ERR_0358	RTE Generator shall throw an error when CompuMethods with category TEXTTABLE, SCALE_LINEAR_AND_TEXTTABLE, SCALE_RATIONAL_AND_TEXTTABLE, or BITFIELD_TEXTTABLE which are referred by ImplementationDataTypes and ApplicationPrimitiveDataTypes of the same Basic Software Module contain a CompuScale with a different point range, identical <literalPrefix>es or no literalPrefix and an identical CompuScale symbolic names.
316	SAFERTE_ERR_0359	RTE Generator shall throw an error when RteMappedToTaskRef is not configured but RteVirtuallyMappedToTaskRef is configured.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
121 / 146

Sl. No.	Error ID	Error Messages
317	SAFERTE_ERR_0360	RTE Generator shall throw an error when for the given OsAlarm, RteActivationType is configured as SYNCHRON.
319	SAFERTE_ERR_0361	RTE Generator shall throw an error when for the given OsScheduleTable with RteActivationType configured as SYNCHRON, OsScalabilityClass is set to either SC1 or SC3.
320	SAFERTE_ERR_0362	RTE Generator shall throw an error when Configured SwComponentPrototype is typed with same SwComponentType as other SwComponentPrototypes.
321	SAFERTE_ERR_0363	RTE Generator shall throw an error when AutosarDataType is not configured for the given VariableDataPrototype.
322	SAFERTE_ERR_0364	RTE Generator shall throw an error when AutosarDataType is not configured for the given SchM VariableDataPrototype.
323	SAFERTE_ERR_0365	Rte Generator shall throw an error when CompuNominatorDenominator does not adhere to the following rules: 1. NumericalValueVariationPoint is not configured. OR 2. NumericalValueVariationPoint.Vs value is not provided or configured as 0.
324	SAFERTE_ERR_XFR003	RTE Generator shall throw an error when the header length in bits specified by bufferProperties.headerLength is not a multiple of 8.
328	SAFERTE_ERR_XFR004	RTE Generator shall throw an error when the attribute inPlace is not set to false for the first transformer in a transformerChain on the sending side.
329	SAFERTE_ERR_XFR005	RTE Generator shall throw an error when the attribute inPlace is not set to false for the last transformer in a transformerChain on the receiving side.
330	SAFERTE_ERR_0366	RTE Generator shall throw an error when an ISignalToIPduMapping is mapped to multiple ComSignals/ComSignalGroups/ComGroupSignals/LdComIPdus.
331	SAFERTE_ERR_0367	RTE Generator shall throw an error when a CompuMethod with category SCALE_LINEAR_AND_TEXTTABLE does not have more than one CompuScales configured and the CompuScale with CompuScaleRationalFormula configured has same lower and upper limits.
332	SAFERTE_ERR_0368	RTE Generator shall throw an error when RteEventToTaskMapping is not configured for the RTEEvent which is used to resolve a WaitPoint.
333	SAFERTE_ERR_0369	RTE Generator shall throw an error OsTask is not non-preemptable in case of RteOsSchedulePoint is configured.
334	SAFERTE_ERR_0370	RTE Generator shall throw an error OsTask is not non-preemptable in case of RteBswOsSchedulePoint is configured.
335	SAFERTE_ERR_0371	If a given RunnableEntity owns RunnableEntityArguments in the role argument, their number shall be identical to the number of applicable portArgValues of the PortAPIOption that references the PortPrototype that in turn is referenced by the OperationInvokedEvent that references the RunnableEntity plus the number of ArgumentDataPrototypes aggregated in the role argument by the ClientServerOperation referenced by said OperationInvokedEvent.
336	SAFERTE_ERR_GEN0001	Input arxml files have wrong reference
337	SAFERTE_ERR_GEN0002	No instance for [instance] in the Rte input files list.
338	SAFERTE_ERR_GEN0003	Module parameter option that cannot be used.
339	SAFERTEPB_ERR_0001	RTE Generator shall throw an error when the value assigned to postBuildVariantCriterionValues is out of range of the referenced

Sl. No.	Error ID	Error Messages
		CompuMethod.
340	SAFERTEPB_ERR_0002	RTE Generator shall throw an error when the value assigned to postBuildVariantConditions is out of range of the referenced CompuMethod.
341	SAFERTEPB_ERR_0003	RTE Generator shall throw an error when different values are assigned to the same PostBuildVariantCriterion.
342	SAFERTEPB_ERR_0004	RTE Generator shall throw an error when {0} PostBuildCriterion is configured for {1} and the same PostBuildCriterion is not available for {2}.
343	SAFERTEPB_ERR_0005	RTE Generator shall throw an error when the same variation point condition is not configured for the ISignalIPdu referring the Isignal/IsignalGroup.

## 7.1.2 Sender Receiver Error Messages

This section explains Sender Receiver error checks implemented by RTE Code Generator.

**Table 9: Sender Receiver Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_SRC001	RTE Generator shall throw an error when referred VariableDataPrototype of the given VariableAccess is not referred from any SenderComSpec of the referred PortPrototype OR PortPrototype reference from VariableAccess is missing.
2	SAFERTE_ERR_SRC002	RTE Generator shall throw an error when Port referred by a VariableAccess and having a ComSpec for the respective VariableDataPrototype does not refer to any PortInterface.
3	SAFERTE_ERR_SRC003	RTE Generator shall throw an error when Port referred by a VariableAccess and having a ComSpec for the respective VariableDataPrototype does not refer to a SenderReceiverInterface.
4	SAFERTE_ERR_SRC004	RTE Generator shall throw an error when Port referred by a VariableAccess and having a ComSpec for the respective VariableDataPrototype does not refer to the SenderReceiverInterface which owns the referred VariableDataPrototype.
5	SAFERTE_ERR_SRC005	RTE Generator shall throw an error when referred VariableDataPrototype of the given VariableAccess is not referred from any ReceiverComSpec of the referred PortPrototype OR PortPrototype reference from VariableAccess is missing.
6	SAFERTE_ERR_SRC006	RTE Generator shall throw an error when in given SenderReceiverToSignalMapping, VariableDataPrototype is not referred in ComSpec's of the Port.
7	SAFERTE_ERR_SRC007	RTE Generator shall throw an error when VDP reference is not provided in BswDataReceivedEvent.

Sl. No.	Error ID	Error Messages
8	SAFERTE_ERR_SRC008	RTE Generator shall throw an error when VariableDataPrototype reference is not provided in the BswVariableAccess.
9	SAFERTE_ERR_SRC009	RTE Generator shall throw an error when RteBswProvidedDataModInstRef reference is not provided for a given RteBswRequiredSenderReceiverConnection.
10	SAFERTE_ERR_SRC010	RTE Generator shall throw an error when RteBswProvidedVariableDataPrototypeRef reference is not provided for a given RteBswRequiredSenderReceiverConnection.
11	SAFERTE_ERR_SRC011	RTE Generator shall throw an error when RteBswRequiredVariableDataPrototypeRef reference is not provided for a given RteBswRequiredSenderReceiverConnection.
12	SAFERTE_ERR_SRC012	RTE Generator shall throw an error when SW-IMPL-POLICY is not set to QUEUED for a given providedData or requiredData.
13	SAFERTE_ERR_SRC013	RTE Generator shall throw an error when Initial Value is not configured for a given providedData or requiredData.
14	SAFERTE_ERR_SRC014	RTE Generator shall throw an error for a given BswQueuedDataReceptionPolicy when queueLength attribute length is configured less than or equal to zero.
15	SAFERTE_ERR_SRC015	RTE Generator shall throw an error when for a RteBswRequiredSenderReceiverConnection, the connected Sender and Receiver are called from multiple contexts.
16	SAFERTE_ERR_SRC016	RTE Generator shall throw an error when M: N communication is configured for SchM Sender Receiver Communication.
17	SAFERTE_ERR_SRC017	RTE Generator shall throw an error when context Limitation is not provided for a BswVariableAccess involved in a MultiPartition environment.
18	SAFERTE_ERR_SRC018	RTE Generator shall throw an error when SwImplPolicyEnum for a DataPrototype is QUEUED but referred in NonqueuedSenderComSpec instead of QueuedSenderComSpec.
19	SAFERTE_ERR_SRC019	RTE Generator shall throw an error when SwImplPolicyEnum for a DataPrototype is QUEUED but referred in NonqueuedReceiverComSpec instead of QueuedReceiverComSpec.

## 7.1.3 Client Server Error Messages

This section explains Client Server error checks implemented by RTE Code Generator.

**Table 10: Client Server Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_CLS001	RTE Generator shall throw an error when two or more OperationInvokedEvents reference a single RunnableEntity, the value of the ServerComSpec attribute queueLength should be identical for all ServerComSpecs owned by PPortPrototypes of the enclosing SwComponentType.
2	SAFERTE_ERR_CLS002	RTE Generator shall throw an error when PortDefinedArgumentValue is not defined for AbstractProvidedPortPrototype typed by a ClientServerInterface.
3	SAFERTE_ERR_CLS003	RTE Generator shall throw an error when the AsynchronousServerCallPoint is not referenced by at most one AsynchronousServerCallResultPoint.
4	SAFERTE_ERR_CLS004	RTE Generator shall throw an error when an OperationInvokedEvent has a reference to a ModeDeclaration in the role disabledMode.
5	SAFERTE_ERR_CLS005	RTE Generator shall throw an error when for inter ECU Client Server communication ServerCallPoint Timeout is not configured.
6	SAFERTE_ERR_CLS006	RTE Generator shall throw an error when for Client Server communication Rte_COMCbkTack_<sn>, Rte_COMCbkTErr_<sn>, Rte_COMCbkInv_<sn>, Rte_COMCbkTxTOut_<sn> are configured.
7	SAFERTE_ERR_CLS007	RTE Generator shall throw an error when a ClientServerToSignalMapping does not have a callSignal, returnSignal and OperationInSystemInstanceRef defined.
8	SAFERTE_ERR_CLS008	RTE Generator shall throw an error when ClientServerOperation owned by an RPortPrototype is referenced by more than one ClientServerToSignalMapping.
9	SAFERTE_ERR_CLS009	RTE Generator shall throw an error when COM Signals does not have ComSignalType as UINT8_N or UINT8_DYN when mapped to callSignal or returnSignal.
10	SAFERTE_ERR_CLS010	RTE Generator shall throw an error when a ComSignal/LdComIpdu is referenced by ClientServerToSignalMapping and a PortDefinedArgumentValue exist for PPortPrototype.
11	SAFERTE_ERR_CLS011	RTE Generator shall throw an error when a ComSignal/LdComIpdu is referenced by more than one ClientServerToSignalMappings mapped to different ClientServerOperations.
12	SAFERTE_ERR_CLS013	RTE Generator shall throw an error when all OperationInvokedEvents belonging to a ClientServerOperation mapped to an extended OsTask does not re-use the same OsEvent.
13	SAFERTE_ERR_CLS014	RTE Generator shall throw an error when in the context of the definition of a ClientIdDefinition, the reference clientServerOperation.contextPort does not refer to an RPortPrototype.
14	SAFERTE_ERR_CLS015	RTE Generator shall throw an error when Somelp transformer is not configured for client server communication.
15	SAFERTE_ERR_CLS016	RTE Generator shall throw an error when all TransformerHardErrorEvents belonging to a ClientServerOperation mapped to an extended OsTask does not re-use the same OsEvent.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
125 / 146

Sl. No.	Error ID	Error Messages
16	SAFERTE_ERR_CLS017	RTE Generator shall throw an error when multiple TransformerHardErrorEvents referencing same ClientServerOperation does not have identical StartOnEvent.
17	SAFERTE_ERR_CLS018	RTE Generator shall throw an error when all ISignals that refer to callSignal or to returnSignal of the same ClientServerToSignalMapping does not have the same TransformationTechnology.protocol and TransformationTechnology.version defined.
18	SAFERTE_ERR_CLS019	RTE Generator shall throw an error when in TransformationISignalProps the attribute csErrorReaction is not set and the transformerClass is equal to serializer and the ISignal that aggregates the TransformationISignalProps transports a client/server communication.
19	SAFERTE_ERR_CLS020	RTE Generator shall throw an error when referred ClientServerOperation of the given ServerCallPoint is not referred from any ClientComSpec of the referred PortPrototype OR PortPrototype reference from ServerCallPoint is missing.
20	SAFERTE_ERR_CLS023	RTE Generator shall throw an error when given Port referred by a ServerCallPoint and having a ClientComSpec for the respective ClientServerOperation does not refer to the ClientServerInterface which owns the referred ClientServerOperation.
21	SAFERTE_ERR_CLS024	RTE Generator shall throw an error when a ComSignal/LdComIpdu referenced by more than one ClientServerToSignalMapping does not have identical DataTransformationErrorHandlingEnum.
22	SAFERTE_ERR_CLS025	RTE Generator shall throw an error when a ComSignal/LdComIpdu referenced by more than one ClientServerToSignalMapping does not have unique Client Identifier for each DataMapping.
23	SAFERTE_ERR_CLS028	RTE Generator shall throw an error when runnable entity has serverCallPoint communication timeout value greater than its timing event periodicity.
24	SAFERTE_ERR_CLS029	RTE Generator shall throw an error when a possibleError referenced by a ClientServerOperation is not owned by the PortInterface that also owns the ClientServerOperation.
25	SAFERTE_ERR_CLS030	RTE Generator shall throw an error when a RunnableEntity referenced by an InitEvent in the role startOnEvent aggregates an AsynchronousServerCallResultPoint.
26	SAFERTE_ERR_CLS031	RTE Generator shall throw an error when the value of the attribute ArgumentDataPrototype.serverArgumentImplPolicy is not set to useArrayBaseType for an ArgumentDataPrototype that is typed by an AutosarDataType that is (after all TYPE_REFERENCES are resolved) either an ImplementationDataType of category ARRAY or an ApplicationDataType mapped to (after all TYPE_REFERENCES are resolved) an ImplementationDataType of category ARRAY.
27	SAFERTE_ERR_CLS032	RTE Generator shall throw an error when for any given TransformerHardErrorEvent, either the attribute TransformerHardErrorEvent.operation or TransformerHardErrorEvent.trigger does not exist.
28	SAFERTE_ERR_CLS033	RTE Generator shall throw an error when in the context of a RunnableEntity, a given AsynchronousServerCallResultPoint is not referenced by one AsynchronousServerCallReturnsEvent in the role eventSource.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
126 / 146

Sl. No.	Error ID	Error Messages
29	SAFERTE_ERR_CLS036	RTE Generator shall throw an error when given ServerCallPoint shares reference to RPortPrototype and a ClientServerOperation with another ServerCallPoint of different type (Asynchronous in case of Synchronous and vice versa).
30	SAFERTE_ERR_CLS037	RTE Generator shall throw an error when waitPoint referencing an AsynchronousServerCallReturnsEvent as well as an AsynchronousServerCallResultPoint referenced by said AsynchronousServerCallReturnsEvent is not aggregated by the same RunnableEntity.
31	SAFERTE_ERR_CLS038	RTE Generator shall throw an error when any SystemSignal (Call or Return) referred from more than one ClientServerToSignalMapping is not referred with same counterpart SystemSignal (Return or Call).
32	SAFERTE_ERR_CLS040	RTE Generator shall throw an error when OsTask reference is not configured in RteEventToTaskMapping for multiple clients who can preempt each other mapped to the server with CanBeInvokedConcurrently as False.
33	SAFERTE_ERR_CLS041	RTE Generator shall throw an error when isSynchronous or isReentrant Parameter for the mapped Provided and Required Client Server Entry is not identical.
34	SAFERTE_ERR_CLS042	RTE Generator shall throw an error when BswModuleEntry mapped to the Provided and Required Client Server Entry is not identical.
35	SAFERTE_ERR_CLS043	RTE Generator shall throw an error when isSynchronous parameter is not true for the BswModuleClientServerEntry mapped to the BswSynchronousServerCallPoint.
36	SAFERTE_ERR_CLS044	RTE Generator shall throw an error when isSynchronous parameter is true for the BswModuleClientServerEntry mapped to the BswAsynchronousServerCallPoint.
37	SAFERTE_ERR_CLS045	RTE Generator shall throw an error when BswModuleEntry configured in BswOperationInvokedEvent and in mapped BswModuleEntity is not identical.
38	SAFERTE_ERR_CLS046	RTE Generator shall throw an error when a BswOperationInvokedEvent has a reference to a ModeDeclaration in the role disabledInMode.
39	SAFERTE_ERR_CLS047	RTE Generator shall throw an error when requiredClientServerEntry is not configured for the CallPoint.
40	SAFERTE_ERR_CLS048	RTE Generator shall throw an error when BswAsynchronousServerCallPoint is not configured for the BswAsynchronousServerCallResultPoint.
41	SAFERTE_ERR_CLS049	RTE Generator shall throw an error when argument in BswModuleEntry is not mapped to any Data Type.
42	SAFERTE_ERR_CLS050	RTE Generator shall throw an error when Returntype in BswModuleEntry is not mapped to any Data Type.
43	SAFERTE_ERR_CLS051	RTE Generator shall throw an error when BswDistinguishedPartition of all the invoking BSWEvents are not present in the CallPoint of respective BswEntity.
44	SAFERTE_ERR_CLS052	RTE Generator shall throw an error when isReentrant is set to true for clients and server in different EcucPartition (No Direct Function call).

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
127 / 146

Sl. No.	Error ID	Error Messages
45	SAFERTE_ERR_CLS053	RTE Generator shall throw an error when RteBswMappedToTaskRef is not configured for the Server and clients and server are in different EcucPartition.
46	SAFERTE_ERR_CLS054	RTE Generator shall throw an error when Multiple EcucPartitions are not mapped to a single BswEvent.
47	SAFERTE_ERR_CLS055	RTE Generator shall throw an error when Multiple BswOperationInvokedEvent invoking the same BswEntity does not have reference to the identical BswModuleClientServerEntry.
48	SAFERTE_ERR_CLS056	RTE Generator shall throw an error for a BswOperationInvokedEvent when IsReentrant parameter is not set to true for mapped BswModuleClientServerEntry and BswModuleEntry and RteBswMappedToTaskRef is not configured.
49	SAFERTE_ERR_CLS057	RTE Generator shall throw an error when returnType direction of BswModuleEntry is not set to 'OUT'.
50	SAFERTE_ERR_CLS059	RTE Generator shall throw an error when arguments or return value are configured for BswModuleEntry mapped to the BswSchedulableEntity.
51	SAFERTE_ERR_CLS061	RTE Generator shall throw an error when BswAsynchronousServerCallPoint and corresponding BswAsynchronousServerCallResultPoint are not mapped to same EcucPartitions.
52	SAFERTE_ERR_CLS062	RTE Generator shall throw an error when BswModuleEntity having BswAsynchronousServerCallPoint is mapped to multiple EcucPartitions then its respective BswModuleEntity which is containing BswAsynchronousServerCallResultPoint are not mapped to same EcucPartitions.
53	SAFERTE_ERR_CLS063	RTE Generator shall throw an error when BswProvidedClientServerEntry is not configured for BswOperationInvokeEvent.
54	SAFERTE_ERR_CLS064	RTE Generator shall throw an error when CalledEntry is not configured for each BswSynchronousServerCallPoint and BswAsynchronousServerCallPoint.
55	SAFERTE_ERR_CLS065	RTE Generator shall throw an error when Context Limitation is not configured in callPoint.
56	SAFERTE_ERR_CLS066	RTE Generator shall throw an error when EncapsulatedEntry is not configured for each ProvidedClientServerEntries and RequiredClientServerEntries.
57	SAFERTE_ERR_CLS067	RTE Generator shall throw an error for an OperationInvokedEvent when Operation/ContextPort/TargetProvidedOperation is not configured for it.
58	SAFERTE_ERR_CLS068	RTE Generator shall throw an error when BswAsynchronousServerCallReturnsEvent does not refer to a BswAsynchronousServerCallResultPoint in the role eventSource.
59	SAFERTE_ERR_CLS069	RTE Generator shall throw an error when for Inter-ECU client server communication E2E transformer is configured.
60	SAFERTE_ERR_CLS070	RTE Generator shall throw an error when RteUsedWaitPointTOOutOsAlarmRef and RteUsedWaitPointOsEventRef parameters in case of Synchronous Client Server communication are not configured for an RteEventToTaskMapping which is mapped to an OperationInvokedEvent with serverCallPoint.timeout greater than zero.



Sl. No.	Error ID	Error Messages
61	SAFERTE_ERR_CLS071	RTE Generator shall throw an error when RteBswUsedWaitPointTOutOsAlarmRef and RteBswUsedWaitPointOsEventRef parameters in case of Synchronous Client Server communication are not configured for an RteBswEventToTaskMapping which is mapped to a runnable through SwcBswRunnableMapping with serverCallPoint.timeout greater than zero.

## 7.1.4 Mode Switch Error Messages

This section explains Mode Switch error checks implemented by RTE Code Generator.

**Table 11: Mode Switch Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_MSW001	RTE Generator shall throw an error when the timeout value of a Wait-Point associated with a ModeSwitchedAckEvent is not equal to the corresponding ModeSwitchedAckRequest.timeout.
2	SAFERTE_ERR_MSW002	RTE Generator shall throw an error when RtePositionInTask/RteBswPositionInTask given for all SwcModeSwitchEvents/BswModeSwitchEvents mapped to the OsTask in case of Synchronous Mode Switch Notification does not follow the given order: On Exit -> On Transition -> On Entry.
3	SAFERTE_ERR_MSW003	RTE Generator shall throw an error when Synchronous Mode Switching related SwcModeSwitchEvents/BswModeSwitchEvents for a ModeDeclarationGroupPrototype are mapped to different OsTasks. Mapping to single OsTask is allowed.
4	SAFERTE_ERR_MSW004	RTE Generator shall throw an error when the Category 2 Runnable (Runnable with WaitPoint) is triggered with SwcModeSwitchEvent or RTEEvent with mode disabling dependency of synchronous mode machine.
5	SAFERTE_ERR_MSW005	RTE Generator shall throw an error when an Os Task implements a synchronous mode machine and priority is greater than all Tasks mapped to RteEvents with mode disabling dependency to this mode machine.
6	SAFERTE_ERR_MSW006	RTE Generator shall throw an error when a SwcModeSwitchEvent referring a ModeDeclaration has a mode disabling dependency on the same ModeDeclaration.
7	SAFERTE_ERR_MSW007	RTE Generator shall throw an error when same ModeDeclarationGroupPrototype and a PortPrototype combination is referred in multiple ModeSwitchPoints N:1 Mode Switch communication.
8	SAFERTE_ERR_MSW008	RTE Generator shall throw an error when RtePositionInTask/RteBswPositionInTask given for all SwcModeSwitchEvents/BswModeSwitchEvents mapped to the OsTask for Asynchronous Mode Switch Notification does not follow the given order: On Exit -> On Transition -> On Entry.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
129 / 146

Sl. No.	Error ID	Error Messages
9	SAFERTE_ERR_MSW009	RTE Generator shall throw an error when no ModeRequestTypeMap is found for the ModeDeclarationGroup of the given ModeDeclarationGroupPrototype for a Software Component.
10	SAFERTE_ERR_MSW010	RTE Generator shall throw an error when Given ModeDeclarationGroup shares its names with other ModeDeclarationGroup's but does not have matching values of ModeDeclarations with them.
11	SAFERTE_ERR_MSW011	RTE Generator shall throw an error when either all or none of the on-entry Runnable Entities of a particular mode machine instance for the initialMode are not mapped to RteInitializationRunnableBatch containers.
12	SAFERTE_ERR_MSW012	RTE Generator shall throw an error when ModeSwitchSenderComSpec is configured with queueLength = 0.
13	SAFERTE_ERR_MSW013	RTE Generator shall throw an error when a SwcModeSwitchEvent with activationkind other than onEntry OR referring to any mode other than initialMode refer to a RteInitializationRunnableBatch via RteUsedInitFnc reference.
14	SAFERTE_ERR_MSW015	RTE Generator shall throw an error when all of the attributes: ModeDeclarationGroup.onTransitionValue and ModeDeclaration.value (for each ModeDeclaration) are not configured for a ModeDeclarationGroup of category EXPLICIT_ORDER.
15	SAFERTE_ERR_MSW016	RTE Generator shall throw an error when none of the attributes: ModeDeclarationGroup.onTransitionValue OR ModeDeclaration.value (for each ModeDeclaration) are configured for a ModeDeclarationGroup of category other than EXPLICIT_ORDER.
16	SAFERTE_ERR_MSW017	RTE Generator shall throw an error when the attributes: ModeDeclarationGroup.onTransitionValue OR ModeDeclaration.value (for each ModeDeclaration) overlap for a ModeDeclarationGroup of category EXPLICIT_ORDER.
17	SAFERTE_ERR_MSW018	RTE Generator shall throw an error when initialMode is not provided for the given ModeDeclarationGroup.
18	SAFERTE_ERR_MSW021	RTE Generator shall throw an error when Modes configuration for the given SwcModeSwitchEvent for ON_TRANSITION does not adhere to the following rules: 1) Two modes are not referred. 2) Mode Declarations referred from both modes does not belongs to same instance (ModeDeclarationGroupPrototype) of ModeDeclarationGroup. 3) Same Mode Declaration are not referred from both modes.
19	SAFERTE_ERR_MSW022	RTE Generator shall throw an error when a BackgroundEvent has mode disabling dependency on a synchronous ModeDeclarationGroupPrototype. BackgroundEvent having mode disabling dependency on a synchronous ModeDeclarationGroupPrototype is not allowed.
20	SAFERTE_ERR_MSW023	RTE Generator shall throw an error when the priority of the OsTask mapped to a SwcModeSwitchEvent involved in an asynchronous mode machine is lower than the OsTask (Runnable) calling the Rte_Switch and any OsTask (Runnable) having mode disabling dependency.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
130 / 146

Sl. No.	Error ID	Error Messages
21	SAFERTE_ERR_MSW024	RTE Generator shall throw an error when ModelInterfaceMapping is configured for any AssemblySwConnector. ModelInterfaceMapping is not supported for any AssemblySwConnector.
22	SAFERTE_ERR_MSW025	RTE Generator shall throw an error when ModeDeclarationGroupPrototype is not configured in Provided and Required ComSpecs.
23	SAFERTE_ERR_MSW026	RTE Generator shall throw an error when RModelInAtomicSwcInstanceRef and its configuration parameters are not configured for SwcModeSwitchEvent.
24	SAFERTE_ERR_MSW027	RTE Generator shall throw an error when PModeGroupInAtomicSwcInstanceRef and its configuration parameters are not configured for ModeSwitchPoint.
25	SAFERTE_ERR_MSW028	RTE Generator shall throw an error when referred ModeDeclarationGroupPrototype of the given ModeSwitchPoint is not referred in any ModeSwitchSenderComSpec of the referred PortPrototype OR PortPrototype reference from ModeSwitchPoint is missing.
26	SAFERTE_ERR_MSW029	RTE Generator shall throw an error when Port referred by a ModeSwitchPoint/ModeAccessPoint and having a ComSpec for the respective ModeDeclarationGroupPrototype does not refer to any PortInterface.
27	SAFERTE_ERR_MSW030	RTE Generator shall throw an error when port referred by a ModeSwitchPoint/ModeAccessPoint and having a ComSpec for the respective ModeDeclarationGroupPrototype does not refer to a ModeSwitchInterface.
28	SAFERTE_ERR_MSW031	RTE Generator shall throw an error when port referred by a ModeSwitchPoint/ModeAccessPoint and having a ComSpec for the respective ModeDeclarationGroupPrototype does not refer to the ModeSwitchInterface owning the referred ModeDeclarationGroupPrototype.
29	SAFERTE_ERR_MSW032	RTE Generator shall throw an error when referred ModeDeclarationGroupPrototype of the given ModeAccessPoint is not referred in any ModeSwitchReceiverComSpec/ModeSwitchSenderComSpec of the referred PortPrototype OR PortPrototype reference from ModeAccessPoint is missing.
30	SAFERTE_ERR_MSW033	RTE Generator shall throw an error when ModelInBswModuleDescriptionInstanceRef and its configuration parameters are not configured for BswModeSwitchEvent.
31	SAFERTE_ERR_MSW034	RTE Generator shall throw an error when ContextModeDeclarationGroup and TargetMode are not configured in ModelInBswModuleDescriptionInstanceRef for a BswModeSwitchEvent.
32	SAFERTE_ERR_MSW035	RTE Generator shall throw an error when ModeDeclarationGroupPrototype Reference is not configured in ManagedModeGroups.
33	SAFERTE_ERR_MSW036	RTE Generator shall throw an error when ModeDeclarationGroupPrototype Reference is not configured in AccessedModeGroups.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
131 / 146

Sl. No.	Error ID	Error Messages
34	SAFERTE_ERR_MSW037	RTE Generator shall throw an error when RteBswPositionInTask given for all BswModeSwitchEvents mapped to the OsTask for Synchronous SchM Mode Switch Notification does not follow the given order: On Exit -> On Transition -> On Entry.
35	SAFERTE_ERR_MSW040	RTE Generator shall throw an error when a BswModeSwitchEvent referring a ModeDeclarationGroupPrototype has a mode disabling dependency on the same ModeDeclarationGroupPrototype.
36	SAFERTE_ERR_MSW042	RTE Generator shall throw an error when the priority of the OsTask mapped to a BswModeSwitchEvent involved in an asynchronous mode machine is lower than the OsTask (ModuleEntity) calling the SchM_Switch and any OsTask (ModuleEntity) having mode disabling dependency.
37	SAFERTE_ERR_MSW043	RTE Generator shall throw an error when no ModeRequestTypeMap is found for the ModeDeclarationGroup of the given ModeDeclarationGroupPrototype for a BSW Module.
38	SAFERTE_ERR_MSW044	RTE Generator shall throw an error when given ModeDeclarationGroup shares its names with other ModeDeclarationGroup's but does not have matching values of ModeDeclarations with them.
39	SAFERTE_ERR_MSW051	RTE Generator shall throw an error when Modes configuration for the given BswModeSwitchEvent for ON_TRANSITION does not adhere to the following rules: 1) Two modes are not referred. 2) Mode Declarations referred from both modes does not belongs to same instance (ModeDeclarationGroupPrototype) of ModeDeclarationGroup. 3) Same Mode Declaration are not referred from both modes.
40	SAFERTE_ERR_MSW052	RTE Generator shall throw an error when a BackgroundEvent has mode disabling dependency on a synchronous ModeDeclarationGroupPrototype. BackgroundEvent having mode disabling dependency on a synchronous ModeDeclarationGroupPrototype is not allowed.
41	SAFERTE_ERR_MSW053	RTE Generator shall throw an error when the attribute QueueLength of BswModeSenderPolicy is configured with value 0.
42	SAFERTE_ERR_MSW054	RTE Generator shall throw an error when ModeDeclarationGroupPrototype reference is not configured in ProvidedModeGroup in BswModeSenderPolicy.
43	SAFERTE_ERR_MSW055	Rte Generator shall throw an error when ModeErrorReactionPolicy for a ModeErrorBehavior is not set to LAST_MODE for a ModeDeclarationGroupPrototype of a provided port connected to ModeDeclarationGroupPrototypes of require ports from more than one partition.
44	SAFERTE_ERR_MSW057	Rte Generator shall throw an error when Default mode is not configured for the ModeErrorBehavior and ModeErrorReactionPolicy is configured as DEFAULT.
45	SAFERTE_ERR_MSW058	RTE Generator shall throw an error when RteBswProvidedModeGrpModInstRef reference is not provided for a given RteBswRequiredModeGroupConnection.
46	SAFERTE_ERR_MSW059	RTE Generator shall throw an error when RteBswProvidedModeGroupRef reference is not provided for a given RteBswRequiredModeGroupConnection.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
132 / 146

Sl. No.	Error ID	Error Messages
47	SAFERTE_ERR_MSW060	RTE Generator shall throw an error when RteBswRequiredModeGroupRef reference is not provided for a given RteBswRequiredModeGroupConnection.
48	SAFERTE_ERR_MSW061	RTE Generator shall throw an error when BswDistinguishedPartition reference is not provided in ContextLimitation for a BswEvent with ModeDisabling configured.
49	SAFERTE_ERR_MSW062	RTE Generator shall throw an error when RteSyncPoint is mapped to more than one OsTask in one core.
50	SAFERTE_ERR_MSW063	RTE Generator shall throw an error when same RteSyncPoint is mapped to different ModeDeclarationGroupPrototype.
51	SAFERTE_ERR_MSW064	RTE Generator shall throw an error when for a ModeSwitchedAckEvent, RteSyncPoint is not configured as predecessor.
52	SAFERTE_ERR_MSW065	RTE Generator shall throw an error when for a RteModeToScheduleTableMapping, OsScheduleTable reference is not provided in RteModeScheduleTableRef.
53	SAFERTE_ERR_MSW066	RTE Generator shall throw an error when for a RteModeToScheduleTableMapping, ModeDeclaration reference is not provided in RteModeSchtblMapModeDeclarationRef.
54	SAFERTE_ERR_MSW067	RTE Generator shall throw an error when for a RteModeToScheduleTableMapping, RteBswModuleInstance reference is not provided in RteModeSchtblMapBsw.RteModeSchtblMapBswInstanceRef.
55	SAFERTE_ERR_MSW068	RTE Generator shall throw an error when for a RteModeToScheduleTableMapping, ModeDeclarationGroupPrototype reference is not provided in RteModeSchtblMapBsw.RteModeSchtblMapBswProvidedModeGroupRef.
56	SAFERTE_ERR_MSW069	RTE Generator shall throw an error when for a RteModeToScheduleTableMapping, RteSwComponentInstance reference is not provided in RteModeSchtblMapSwc.RteModeSchtblMapSwcInstanceRef.
57	SAFERTE_ERR_MSW070	RTE Generator shall throw an error when for a RteModeToScheduleTableMapping, Port reference is not provided in RteModeSchtblMapSwc.RteModeSchtblMapSwcPortRef.
58	SAFERTE_ERR_MSW071	RTE Generator shall throw an error when for a BswModeSwitchedAckEvent, RteSyncPoint is not configured as predecessor.
59	SAFERTE_ERR_MSW072	RTE Generator shall throw an error when an RteEvent which is not a SwcModeSwitchEvent is mapped to a Mode Machine OsTask.
60	SAFERTE_ERR_MSW073	RTE Generator shall throw an error when a BswEvent which is not a BswModeSwitchEvent is mapped to a Mode Machine OsTask.
61	SAFERTE_ERR_MSW074	RTE Generator shall throw an error when for the given RteModeToScheduleTableMapping both RteModeSchtblMapBsw and RteModeSchtblMapSwc are not configured.
62	SAFERTE_ERR_MSW075	Rte Generator shall throw an error when mapping of Schedule Table to multiple Modes is found.
63	SAFERTE_ERR_MSW076	Rte Generator shall throw an error when RtePositionInTask/RteBswPositionInTask is not configured for directly invoked ExecutableEntities mapped to SwcModeSwitchEvents/BswModeSwitchEvents and it does not have Unique Position Number.

Sl. No.	Error ID	Error Messages
64	SAFERTE_ERR_MSW078	Rte Generator shall throw an error when ModeDeclarationGroupPrototype reference is not provided in SwcBswSynchronizedModeGroupPrototype.BswModeGroup for a configured SwcBswSynchronizedModeGroupPrototype.
65	SAFERTE_ERR_MSW079	Rte Generator shall throw an error when ModeDeclarationGroupPrototype/contextPort reference is not provided in SwcBswSynchronizedModeGroupPrototype.SwcModeGroup for a configured SwcBswSynchronizedModeGroupPrototype.
66	SAFERTE_ERR_MSW080	Rte Generator shall throw an error when for a configured SwcBswSynchronizedModeGroupPrototype, either ModeSwitchPoint or ManagedModeGroup is not configured.

## 7.1.5 Trigger Error Messages

This section explains Trigger error checks implemented by RTE Code Generator.

**Table 12: Trigger Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_TRG001	RTE Generator shall throw an error when Trigger referred from a RteExternalTriggerConfig container is not triggered from the SwComponent referred by the parent RteSwComponentInstance.
2	SAFERTE_ERR_TRG002	RTE Generator shall throw an error when Trigger referred from a RteInternalTriggerConfig container is not triggered from the SwComponent referred by the parent RteSwComponentInstance.
3	SAFERTE_ERR_TRG003	RTE Generator shall throw an error when an External Trigger with SwImplPolicy = QUEUED does not have RteExternalTriggerConfig configured along with RteTriggerSourceQueueLength <= 0.
4	SAFERTE_ERR_TRG004	RTE Generator shall throw an error when an Internal Triggering Point with SwImplPolicy = QUEUED does not have RteInternalTriggerConfig configured along with RteTriggerSourceQueueLength <= 0.
5	SAFERTE_ERR_TRG005	RTE Generator shall throw an error when a Trigger is referred from multiple Triggering Points i.e. N:1 Communication.
6	SAFERTE_ERR_TRG006	RTE Generator shall throw an error when a Trigger involved in Inter ECU Trigger Communication has SwImplPolicy = QUEUED.
7	SAFERTE_ERR_TRG007	RTE Generator shall throw an error when an ISignal mapped to Trigger without Data-Transformation has non-zero length.
8	SAFERTE_ERR_TRG008	RTE Generator shall throw an error when a Single Trigger is used from Inter as well as Intra ECU communications.
9	SAFERTE_ERR_TRG009	RTE Generator shall throw an error when an Internal Triggering Point is used as a source to trigger Runnable owning the same Internal Trigger Point.
10	SAFERTE_ERR_TRG010	RTE Generator shall throw an error when a Runnable Entity triggered by an ExternalTriggerOccuredEvent/InternalTriggerOccuredEvent has a WaitPoint.
11	SAFERTE_ERR_TRG012	RTE Generator shall throw an error when an OsTask mapped to Trigger Sink having lower priority than the Trigger Source is not a Basic Task in case of queued Trigger Communication.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
134 / 146

Sl. No.	Error ID	Error Messages
12	SAFERTE_ERR_TRG013	RTE Generator shall throw an error when an OsTask mapped to Trigger Sink having lower priority than the Trigger Source has OsTaskActivation less than the Trigger Queue Length in case of queued Trigger Communication.
13	SAFERTE_ERR_TRG014	RTE Generator shall throw an error when a Basic Task mapped to sinks of a particular Trigger is not mapped to ExternalTriggerOccuredEvents for a single Trigger (i.e. Other Types of RTE Events are not allowed to be mapped this Task) in case of queued Trigger Communication.
14	SAFERTE_ERR_TRG015	RTE Generator shall throw an error when an ISignal mapped to the Trigger belongs to COM (UINT8_N/UINT8_DYN) or LdCom and Transformer configuration is not present for the same in case of Inter ECU Trigger communication
15	SAFERTE_ERR_TRG016	RTE Generator shall throw an error when an ISignal of non-zero length is mapped to the Trigger and Somelp Transformer configuration is not present for it (i.e. only Somelp transformer must be configured for an ISignal of non-zero length) in case of Inter ECU Trigger communication.
16	SAFERTE_ERR_TRG017	RTE Generator shall throw an error when a ComSignal mapped to Trigger with non-zero length is not of type UINT8_N or UINT8_DYN in case of Inter ECU Trigger communication.
17	SAFERTE_ERR_TRG018	RTE Generator shall throw an error when a Basic Task mapped to sinks of a Trigger is not mapped to InternalTriggerOccuredEvents for a single Trigger (i.e. Other Types of RTE Events are not allowed to be mapped to this Task) in case of queued Trigger Communication.
18	SAFERTE_ERR_TRG019	RTE Generator shall throw an error when an OsTask mapped to Trigger Sink having lower priority than the Trigger Source is not a Basic Task in case of queued Trigger Communication for InterRunnable Triggering.
19	SAFERTE_ERR_TRG020	RTE Generator shall throw an error when an OsTask mapped to Trigger Sink having lower priority than the Trigger Source has OsTaskActivation less than the Trigger Queue Length in case of queued Trigger Communication for InterRunnable Triggering.
20	SAFERTE_ERR_TRG021	RTE Generator shall throw an error when ComUpdateBitPosition is not defined for the ComSignal in case of Inter-ECU Trigger Communication without Data Transformation.
21	SAFERTE_ERR_TRG022	RTE Generator shall throw an error when ComNotification is not defined for the ComSignal on Sink Side in case of Inter-ECU Trigger Communication.
22	SAFERTE_ERR_TRG023	RTE Generator shall throw an error when TargetTrigger/Port/ContextComponent is not provided in TriggerToSignalMapping container in the System.
23	SAFERTE_ERR_TRG024	RTE Generator shall throw an error when SystemSignal reference is not provided in TriggerToSignalMapping.
24	SAFERTE_ERR_TRG025	RTE Generator shall throw an error when StartOnEvent/ContextRPort/TargetTrigger is not configured in ExternalTriggerOccurredEvent.
25	SAFERTE_ERR_TRG026	RTE Generator shall throw an error when StartOnEvent/EventSource is not configured in InternalTriggerOccurredEvent.
26	SAFERTE_ERR_TRG027	RTE Generator shall throw an error when ContextPPort/TargetTrigger is not configured in ExternalTriggeringPoint.
27	SAFERTE_ERR_TRG028	RTE Generator shall throw an error when an ExternalTriggeringPoint refers to a Trigger whose parent TriggerInterface is not used by the contextPPort.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
135 / 146

Sl. No.	Error ID	Error Messages
28	SAFERTE_ERR_TRG029	RTE Generator shall throw an error when RteExternalTriggerConfig is configured and RteSwcTriggerSourceRef is not configured for the same.
29	SAFERTE_ERR_TRG030	RTE Generator shall throw an error when RteInternalTriggerConfig is configured and RteSwcTriggerSourceRef is not configured for the same.
30	SAFERTE_ERR_TRG031	RTE Generator shall throw an error when task attribute defined in BswTriggerDirectImplementation and task of ExecutableEntity activated by BswExternalTriggerOccurredEvent associated with connected Trigger of the trigger source are not same.
31	SAFERTE_ERR_TRG032	RTE Generator shall throw an error when IssuedTrigger association and a BswTriggerDirectImplementation is defined for the same releasedTrigger.
32	SAFERTE_ERR_TRG033	RTE Generator shall throw an error when same trigger is referred in multiple BswRequiredTriggerConnections. N:1 Communication is not supported as per AUTOSAR.
33	SAFERTE_ERR_TRG034	RTE Generator shall throw an error when BswInternalTriggeringPoint is used as a source to trigger BswModuleEntity owning the same Internal Trigger Point.
34	SAFERTE_ERR_TRG035	RTE Generator shall throw an error when OsTask mapped to BswEvent of Required Trigger is not a Basic Task in case of queued Trigger Communication.
35	SAFERTE_ERR_TRG036	RTE Generator shall throw an error when OsTask mapped to BswEvent of Required Trigger having lower priority than the Released Trigger has OsTaskActivation less than the Trigger Queue Length in case of queued Trigger Communication.
36	SAFERTE_ERR_TRG037	RTE Generator shall throw an error when BswExternalTriggerOccuredEvent mapped to a BasicTask of a particular Required Trigger is mapped to other types of RTE/BSW Events in case of queued Trigger Communication.
37	SAFERTE_ERR_TRG038	RTE Generator shall throw an error when BswInternalTriggerOccuredEvent mapped to a BasicTask of a particular BswInternalTriggeringPoint is mapped to other types of RTE/BSW Events in case of queued Trigger Communication.
38	SAFERTE_ERR_TRG039	RTE Generator shall throw an error when OsTask mapped to BswInternalTriggerOccurredEvent is not configured as a Basic Task in case of queued Trigger Communication for BswInterRunnableTriggering.
39	SAFERTE_ERR_TRG040	RTE Generator shall throw an error when OsTask mapped to BswInternalTriggerOccurredEvent having lower priority than the one acting as a Source has OsTaskActivation less than the Trigger Queue Length in case of queued Trigger Communication for BswInterRunnableTriggering.
40	SAFERTE_ERR_TRG041	RTE Generator shall throw an error when RteBswTriggerSourceRef is not configured and RteBswInternalTriggerConfig is configured.
41	SAFERTE_ERR_TRG042	RTE Generator shall throw an error when RteBswTriggerSourceRef is not configured and RteBswExternalTriggerConfig is configured.
42	SAFERTE_ERR_TRG043	RTE Generator shall throw an error when Task attribute defined in BswTriggerDirectImplementation is not configured as BasicTask.
43	SAFERTE_ERR_TRG044	RTE Generator shall throw an error when StartsOnEvent/Trigger is not configured in BswExternalTriggerOccurredEvent.
44	SAFERTE_ERR_TRG045	RTE Generator shall throw an error when StartsOnEvent/EventSource is not configured in BswInternalTriggerOccurredEvent.
45	SAFERTE_ERR_TRG046	RTE Generator shall throw an error when Referred Trigger in BswExternalTriggerOccurredEvent does not belong to same BswModuleDescription.

Sl. No.	Error ID	Error Messages
46	SAFERTE_ERR_TRG047	RTE Generator shall throw an error when Referred EventSource in BswInternalTriggerOccurredEvent does not belong to same BswModuleDescription.
47	SAFERTE_ERR_TRG048	Rte Generator shall throw an error if referred Trigger in RteBswTriggerSourceRef does not belong to the BswModuleDescription which is referred in the related RteBswImplementationRef.
48	SAFERTE_ERR_TRG049	Rte Generator shall throw an error if referred BswInternalTriggeringPoint in RteBswTriggerSourceRef does not belong to the BswModuleDescription which is referred in the related RteBswImplementationRef.
49	SAFERTE_ERR_TRG050	Rte Generator shall throw an error when RtePositionInTask is not configured for directly invoked ExecutableEntities mapped to ExternalTriggerOccurredEvent/InternalTriggerOccurredEvent and it does not have Unique Position Number.
50	SAFERTE_ERR_TRG051	Rte Generator shall throw an error when RteBswPositionInTask is not configured for directly invoked ExecutableEntities mapped to BswExternalTriggerOccurredEvent/BswInternalTriggerOccurredEvent and it does not have Unique Position Number.

## 7.1.6 Nv Data Error Messages

This section explains NV Data error checks implemented by RTE Code Generator.

**Table 13: Nv Data Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_NVD001	RTE Generator shall throw an error when VariableDataPrototype of the given VariableAccess is not associated with any NvProvideComSpec of referred PortPrototype OR when PortPrototype reference is missing from the Variable Access.
2	SAFERTE_ERR_NVD002	RTE Generator shall throw an error when given Port referred by VariableAccess having ComSpec for VDP does not refer to any Port Interface
3	SAFERTE_ERR_NVD003	RTE Generator shall throw an error when given Port referred by VariableAccess having ComSpec for VDP does not refer to NvDataInterface.
4	SAFERTE_ERR_NVD004	RTE Generator shall throw an error when VDP referred by given Port and NvDataInterface are not same.
5	SAFERTE_ERR_NVD005	RTE Generator shall throw an error when the VDP of the given VariableAccess is not referred from any NvRequireComSpec of PortPrototype OR PortPrototype reference from VariableAccess is missing.
6	SAFERTE_ERR_NVD006	RTE Generator shall throw an error when RomBlock of the configured NvBlockDescriptor has type missing or has a different type than the RamBlock.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
137 / 146

Sl. No.	Error ID	Error Messages
7	SAFERTE_ERR_NVD007	RTE Generator shall throw an error when Configured port with NvRequireComSpec is unconnected and does not have initValue specified.
8	SAFERTE_ERR_NVD008	RTE Generator shall throw an error when configured NvBlockDescriptor is mapped to NvDatas belonging to different Partitions.
9	SAFERTE_ERR_NVD009	RTE Generator shall throw an error when configured NvBlockDataMapping violate following rules a. The role writtenNvData shall only be used if the corresponding PortPrototype is a RPortPrototype. b. The role writtenReadNvData shall only be used if the corresponding Port-Prototype is a PRPortPrototype. c. The role readNvData shall only be used if the corresponding Port-Prototype is a PPortPrototype.
10	SAFERTE_ERR_NVD010	RTE Generator shall throw an error when configured NvBlockDataMapping does not map RamBlock/part of RamBlock to a compatible nvData.
11	SAFERTE_ERR_NVD011	RTE Generator shall throw an error when configured NvBlockDataMapping refers a nvData that is also referred by another NvBlockDataMapping. A NvData can only be mapped to one NvBlockDataMapping.
12	SAFERTE_ERR_NVD012	RTE Generator shall throw an error when configured SwcServiceDependency violates constr_1404 i.e. all NvDataInterface.nvData of PortPrototypes in the context of this SwcServiceDependency are not mapped to the same NvBlockDescriptor.
13	SAFERTE_ERR_NVD013	RTE Generator shall throw an error when neither Port defined Argument value nor RteNvRamAllocation is configured for NvBlockDescriptor.
14	SAFERTE_ERR_NVD014	RTE Generator shall throw an error when for given descriptor RAM block is not configured.
15	SAFERTE_ERR_NVD015	RTE Generator shall throw an error when for given descriptor configured with modeSwitchEventTriggeredActivities Parameters SwcModeswitchevent and Role is not configured.
16	SAFERTE_ERR_NVD016	RTE Generator shall throw an error when for given descriptor nvBlockDataMappings is not configured.
17	SAFERTE_ERR_NVD017	RTE Generator shall throw an error when for given nvBlockDataMapping parameter nvRamBlockElement is not configured.
18	SAFERTE_ERR_NVD018	RTE Generator shall throw an error when for given descriptor configured with DirtyFlag and storeCyclic as TRUE, Timing Event is not configured.
19	SAFERTE_ERR_NVD019	RTE Generator shall throw an error when for given descriptor with DirtyFlag and storeImmediate set to TRUE, DataReceivedEvent is not configured.
20	SAFERTE_ERR_NVD020	RTE Generator shall throw an error when for given descriptor with DirtyFlag and storeAtShutdown set to TRUE, DataReceivedEvent is not configured.
21	SAFERTE_ERR_NVD021	RTE Generator shall throw an error when Index element is not configured in the NvBlockDataMapping for partially mapped NvData to NvRamBlock Element of category ARRAY.

Sl. No.	Error ID	Error Messages
22	SAFERTE_ERR_NVD022	RTE Generator shall throw an error when for given nvBlockDataMapping, one of parameter amongst readNvData, writtenNvData or writtenReadNvData is not configured.
23	SAFERTE_ERR_NVD023	RTE Generator shall throw an error when RteSwNvBlockDescriptorRef is configured along with RteSwNvRamMappingRef /RteNvmRomBlockLocationSymbol/RteNvmRamBlockLocationSymbol.

## 7.1.7 Calibration Error Messages

This section explains Calibration error checks implemented by RTE Code Generator.

**Table 14: Calibration Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_MC001	RTE Generator shall throw an error when ParameterRequireComSpec is not configured for given ParameterDataPrototype mapped to the ParameterAccess.
2	SAFERTE_ERR_MC002	RTE Generator shall throw an error when ParameterSwComponentType is not referred from an instantiated RteSwComponentType container to configure calibration settings related to the ParameterSwComponentType.
3	SAFERTE_ERR_MC003	RTE Generator shall throw an error when if RteCalibrationSupport or RteMeasurementSupport is configured and FlatMap is not referred in rootSoftwareComposition.
4	SAFERTE_ERR_MC004	RTE Generator shall throw an error when same Calibration parameter is referred by multiple ports in a ParameterSwComponentType.
5	SAFERTE_ERR_MC005	RTE Generator shall throw an error when SwImplPolicy for the ParameterDataPrototype is not set to 'STANDARD'.
6	SAFERTE_ERR_MC006	RTE Generator shall throw an error when initValue is not configured in ParameterRequireComSpec for an unconnected ParameterDataPrototype.
7	SAFERTE_ERR_MC007	RTE Generator shall throw an error when InitValue is not configured in ParameterProvideComSpec or in CalibrationParameterValue inside CalibrationParameterValueSet container.
8	SAFERTE_ERR_MC008	RTE Generator shall throw an error when portPrototype configured in parameter access is not an RPortPrototype of an Application SWC.

Sl. No.	Error ID	Error Messages
9	SAFERTE_ERR_MC009	RTE Generator shall throw an error when Data emulation method is configured and FlatMap is not configured for each ParameterDataPrototype, Shared Parameter and Per Instance Parameter.
10	SAFERTE_ERR_MC010	RTE Generator shall throw an error when SwAddrMethod.SectionType is not set to CALPRM for each calibration parameter.
11	SAFERTE_ERR_MC011	RTE Generator shall throw an error when ParameterDataPrototype accessed through a RPortPrototype for an AtomicSwComponentType provided by multiple ParameterSwComponentTypes.
12	SAFERTE_ERR_MC012	RTE Generator shall throw an error when PPortProtoType is not configured in ParameterSwComponentType.
13	SAFERTE_ERR_MC013	RTE Generator shall throw an error when InitValue is not configured for each Shared Parameter and per Instance Parameter.
14	SAFERTE_ERR_MC014	RTE Generator shall throw an error when AutosarDataPrototype is not configured for the given SchM ParameterDataPrototype.
15	SAFERTE_ERR_MC015	RTE Generator shall throw an error when AutosarDataPrototype is not configured for the given ParameterDataPrototype.
16	SAFERTE_ERR_MC016	RTE Generator shall throw an error when measurement is configured for queue in sender receiver communication.

## 7.1.8 Multi Partition Error Messages

This section explains Multi Partition error checks implemented by RTE Code Generator. The errors will only come when multi partition scenarios are configured in the input arxml.

**Table 15: Multi Partition Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_MPC001	RTE Generator shall throw an error when the handleTerminationAndRestart attribute of a SW-C is not set to canBeTerminatedAndRestarted and this SW-C is mapped on a Partition with the PartitionCanBeRestarted parameter set to TRUE.
2	SAFERTE_ERR_MPC003	RTE Generator shall throw an error when RteInitializationRunnableBatch refers to RunnableEntities belonging to different partitions.
3	SAFERTE_ERR_MPC004	RTE Generator shall throw an error when Runnable Entities of a SW-C are mapped to tasks of different partitions
4	SAFERTE_ERR_MPC005	RTE Generator shall throw an error when configured OsApplication is not mapped to an EcucPartition.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
140 / 146

Sl. No.	Error ID	Error Messages
5	SAFERTE_ERR_MPC006	RTE Generator shall throw an error when PartitionCanBeRestarted parameter is set to true for an EcucPartition, but its corresponding OsApplication does not have the RestartTask configured.
6	SAFERTE_ERR_MPC008	RTE Generator shall throw an error when Os PDF is not provided as an input to RTE Code Gen. The error only occurs in case of Multi-partition scenarios.
7	SAFERTE_ERR_MPC009	RTE Generator shall throw an error when given SW-Component is not mapped to any EcucPartition.
8	SAFERTE_ERR_MPC010	RTE Generator shall throw an error when configured OsApplication does not refer to an EcucCoreDefinition in case of multi-core.
9	SAFERTE_ERR_MPC011	RTE Generator shall throw an error when given EcucPartition is not mapped to any OsApplication.
10	SAFERTE_ERR_MPC012	RTE Generator shall throw an error when configured EcucCoreDefinition is not mapped to any OsApplication.
11	SAFERTE_ERR_MPC013	RTE Generator shall throw an error when no OsApplication is configured even though one or more EcucPartition's are configured.
12	SAFERTE_ERR_MPC014	RTE Generator shall throw an error when there are no EcucPartition with ecucPartitionBswModuleExecution or ecucPartitionQmBswModuleExecution parameter set to true.
13	SAFERTE_ERR_MPC015	RTE Generator shall throw an error when given EcucPartition refers to BswDistinguishshdPartition's but neither ecucPartitionBswModuleExecution nor ecucPartitionQmBswModuleExecution is set to true.
14	SAFERTE_ERR_MPC016	RTE Generator shall throw an error in case of Inter ECU data mapping with Com module, if minimum one BswDistinguishedPartition of Com is not mapped to an EcucPartition.
15	SAFERTE_ERR_MPC017	RTE Generator shall throw an error in case of Inter ECU data mapping with LdCom module, if minimum one BswDistinguishedPartition of LdCom is not mapped to an EcucPartition.
16	SAFERTE_ERR_MPC018	RTE Generator shall throw an error if in a multi-partition configuration with NvBlockSwComponentTypes present, BswDistinguishedPartition belonging to NvM is not mapped to an EcucPartition.
17	SAFERTE_ERR_MPC019	RTE Generator shall throw an error when Task Reference from RteEventToTaskMapping is not given for ModeUser SwcModeSwitchEvent and Mode Manager belonging to different partitions.
18	SAFERTE_ERR_MPC020	RTE Generator shall throw an error when Task Reference is not provided from RteEventToTaskMapping when Trigger Sink SWC and Trigger Source SWC belong to different partitions.
19	SAFERTE_ERR_MPC021	RTE Generator shall throw an error if in case of Client Server Multi Partitioning, direct function call is configured for Operation Invoked Events.
20	SAFERTE_ERR_MPC022	RTE Generator shall throw an error when ComNotification callback is not configured at Receiver side for MultiPartition.
21	SAFERTE_ERR_MPC023	RTE Generator shall throw an error if BswDistinguishedPartitions of EcuM is not mapped to EcucPartitions of all the cores.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
141 / 146

Sl. No.	Error ID	Error Messages
22	SAFERTE_ERR_MPC024	RTE Generator shall throw an error when RunnableEntity is mapped to multiple OsTasks directly/indirectly via multiple RTEEvents and RteEventToTaskMappings.
23	SAFERTE_ERR_MPC025	RTE Generator shall throw an error when general parameter RteCrossCoreServiceTOut is set to a zero value in case of a multi-core configuration. That is the waiting time for an RTE API for any cross-core request before a timeout.
24	SAFERTE_ERR_MPC026	RTE Generator shall throw an error when VariableAccess configured for Transmission involved in an Inter-Intra ECU Sender Receiver Communication with receivers and Com/LdCom mapped to different partitions.
25	SAFERTE_ERR_MPC027	RTE Generator shall throw an error when BswDistinguishedPartition is not configured for Com or LdCom and their EcucModuleConfigurationValues are found when multiple EcucPartitions are configured.
26	SAFERTE_ERR_MPC028	RTE Generator shall throw an error when RteBswExclusiveAreaOsSpinlockRef is not configured and RteExclusiveAreaImplMechanism is configured as OS_SPINLOCK for a RteBswExclusiveAreaImpl container
27	SAFERTE_ERR_MPC029	RTE Generator shall throw an error when RteExclusiveAreaImplMechanism is configured as OS_SPINLOCK for a RteExclusiveAreaImplementation container.
28	SAFERTE_ERR_MPC030	RTE Generator shall throw an error when RteCrossCoreServiceTOut parameter value is not divisible by the OsSecondsPerTick/RteExpectedTickDuration parameter associated with the Hardware Counter available on the core.
29	SAFERTE_ERR_MPC031	RTE Generator shall throw an error when Task Reference in RteBswEventToTaskMapping is not configured as Released Trigger's BSWMD and Required Trigger's BSWMD belong to different partitions.
30	SAFERTE_ERR_MPC032	RTE Generator shall throw an error for a BSW module involved in a MultiPartition environment when BswDistinguishedPartition is not configured and RteBswMappedToTaskRef is also not configured in any of the RteBswEventToTaskMapping(s).
31	SAFERTE_ERR_MPC033	RTE Generator shall throw an error when all Mode Users are not mapped to the same partition in Synchronous Mode Switch.
32	SAFERTE_ERR_MPC034	RTE Generator shall throw an error for a BswEvent involved in SchM Sender Receiver communication when Context Limitation is not provided for the following cases: 1) BswEvent is not a BswBackgroundEvent or a BswTimingEvent. 2) BswEvent is a BswBackgroundEvent or a BswTimingEvent and the referred BswModuleEntity contains BswVariableAccesses.
33	SAFERTE_ERR_MPC035	RTE Generator shall throw an error when a BswEvent involved in SchM Sender Receiver communication is mapped to multiple BswDistinguishedPartition.
34	SAFERTE_ERR_MPC037	RTE Generator shall throw an error when a configured OsTask or an OsAlarm in a Multi Partition scenario is not mapped to any OsApplication.

Sl. No.	Error ID	Error Messages
35	SAFERTE_ERR_MPC038	<p>RTE Generator shall throw an error when BswModuleEntity mapped to multiple EcucPartitions directly/indirectly calls SchM APIs in a Multi-Partition environment via one of the following paths:</p> <ol style="list-style-type: none"> <li>1. n BswEvent -&gt; m BswDistinguishedPartition -&gt; (&gt; 1) EcucPartition.</li> <li>2. n BswEvent -&gt; m OsTask -&gt; (&gt; 1) OsApplication -&gt; (&gt; 1) EcucPartition.</li> <li>3. n BswDirectCallPoint -&gt; m BswDistinguishedPartition -&gt; (&gt; 1) EcucPartition</li> </ol> <p>Mapping to EcucPartition should be unambiguous and single (= 1) for such BswModuleEntity.</p>

### 7.1.9 PortInterfaceMapping Error Messages

**Table 16: PortInterfaceMapping Error Messages**

Sl. No.	Error ID	Error Messages
1	SAFERTE_ERR_PIM001	RTE Generator shall throw an error when DataPrototypeMapping is not configured and VariableAndParameterInterfaceMapping is configured for an AssemblySwConnector.
2	SAFERTE_ERR_PIM002	RTE Generator shall throw an error when both FirstDataPrototype and SecondDataPrototype parameters are not configured for a DataPrototypeMapping and VariableAndParameterInterfaceMapping is configured for an AssemblySwConnector.
3	SAFERTE_ERR_PIM003	RTE Generator shall throw an error when NvDataInterface to ParmeterInterface PortInterfaceMapping is configured.
4	SAFERTE_ERR_PIM004	RTE Generator shall throw an error when ClientServerOperationMapping is not configured and ClientServerInterfaceMapping is configured for an AssemblySwConnector.
5	SAFERTE_ERR_PIM005	RTE Generator shall throw an error when both FirstOperation and SecondOperation parameters are not configured for a ClientServerOperationMapping and ClientServerInterfaceMapping is configured for an AssemblySwConnector.
6	SAFERTE_ERR_PIM006	RTE Generator shall throw an error when SenderReceiverInterface (Sender) to ParmeterInterface (Receiver) PortInterfaceMapping is configured.
7	SAFERTE_ERR_PIM007	RTE Generator shall throw an error when for a DataPrototypeMapping, the firstDataPrototype and SecondDataPrototype belongs to same SenderReceiverInterface.
8	SAFERTE_ERR_PIM008	RTE Generator shall throw an error when for a DataPrototypeMapping, the firstDataPrototype and SecondDataPrototype belongs to same NvDataInterface.
9	SAFERTE_ERR_PIM009	RTE Generator shall throw an error when for a DataPrototypeMapping, the firstDataPrototype and SecondDataPrototype belongs to same ParmeterInterface.
10	SAFERTE_ERR_PIM010	RTE Generator shall throw an error when for a ClientServerOperationMapping, the FirstOperation and SecondOperation belongs to same ClientServerInterface.

Sl. No.	Error ID	Error Messages
11	SAFERTE_ERR_PIM011	RTE Generator shall throw an error when for a ClientServerOperationMapping, firstOperation and secondOperation arguments size is not identical.
12	SAFERTE_ERR_PIM012	RTE Generator shall throw an error when for a ClientServerOperationMapping, argument directions of firstOperation and secondOperation are not identical.
13	SAFERTE_ERR_PIM013	RTE Generator shall throw an error when for a ClientServerOperationMapping, argument serverArgumentImplPolicies of firstOperation and secondOperation are not identical.
14	SAFERTE_ERR_PIM014	RTE Generator shall throw an error when for a ClientServerOperationMapping, the data types mapped to arguments of firstOperation and secondOperation are not compatible.
15	SAFERTE_ERR_PIM015	RTE Generator shall throw an error when for a ClientServerOperationMapping, application errors mapped to firstOperation and secondOperation are not identical.

## 7.1.10 Warning Messages

This section explains about the warning ID and warning message description considered in Safe RTE Code Generator.

**Table 17: Warning Messages**

Sl. No.	Warning ID	Description
1	SAFERTE_WARN_0002	RTE Generator shall throw a warning when RteDevErrorDetectUninit is set to "TRUE" and RteDevErrorDetect Parameter is not configured.
2	SAFERTE_WARN_0003	RTE Generator shall throw a warning when TIMEOUT value configured is less than or equal to the ALIVE-TIMEOUT value in the NonqueuedReceiverComSpec.
3	SAFERTE_WARN_0004	RTE Generator shall throw a warning when ComNotification parameter is configured and TRANSMISSION-ACKNOWLEDGMENT and DATA-SEND-COMPLETED-EVENT is not configured.
4	SAFERTE_WARN_0005	RTE Generator shall throw a warning when TriggerTransmit callback is not configured for LdCom in case of sender application.
5	SAFERTE_WARN_0007	RTE Generator shall throw a warning, if both VariableReadAccRef and VariableWriteAccRef are configured in RteImplicitCommunication container.
6	SAFERTE_WARN_0008	RTE Generator shall throw a warning when HANDLE-NEVER-RECEIVED is set to TRUE and COM notification callback is not configured.
7	SAFERTE_WARN_0009	RTE Generator shall throw a warning when DATA-RECEIVED-EVENT or DATA-RECEIVE-ERROR-EVENT exists, and COM reception Callback is not configured.
8	SAFERTE_WARN_0011	RTE Generator shall throw a warning when FAN-IN configuration exists and COM callback (Rte_ComCbk) is not configured.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
144 / 146

Sl. No.	Warning ID	Description
9	SAFERTE_WARN_0012	RTE Generator shall throw a warning when the server runnables property canBeInvokedConcurrently is set to TRUE and OsTask is referred in RteEventToTaskMapping. Rte generator should ignore the referred task.
10	SAFERTE_WARN_0013	RTE Generator shall throw a warning when canBeInvokedConcurrently is set to TRUE and server queue length is not configured as 1.
11	SAFERTE_WARN_0014	RTE Generator shall throw a warning if the canBeInvokedConcurrently attribute of the server runnable is set to TRUE and TimeOut is configured.
12	SAFERTE_WARN_0015	RTE Generator shall throw a warning when SwImplPolicy is set to queued and QueuedComSpec is not configured or vice versa.
13	SAFERTE_WARN_0016	RTE Generator shall throw a warning when TIMEOUT is configured for LdCom.
14	SAFERTE_WARN_0017	RTE Generator shall throw a warning when data constraint is not configured, and range check is required.
15	SAFERTE_WARN_0019	RTE Generator shall throw a warning when RteExclusiveAreaPolicy is not configured for each exclusive area.
16	SAFERTE_WARN_0020	RTE Generator shall throw a warning when BswExclusiveAreaPolicy is not configured for each exclusive area.
17	SAFERTE_WARN_0021	RTE Generator shall throw a warning when Category of ModeDeclarationGroup is not configured involved in Rte Mode Switch.
18	SAFERTE_WARN_0023	RTE Generator shall throw a warning when Runnable entities configured are not mapped to any of the RTE Event.
19	SAFERTE_WARN_0024	RTE Generator shall throw a warning when an API is called from a runnable which is not mapped to any task. Also, API shall not be able to detect calls from illegal invocation at the runtime.
20	SAFERTE_WARN_0025	RTE Generator shall throw a warning when Rte_Enter/Rte_Exit APIs are called from a runnable which is not mapped to any task. Also, it shall not handle RTE_E_DET_ILLEGAL_NESTED_EXCLUSIVE_AREA during run-time.
21	SAFERTE_WARN_0026	RTE Generator shall throw a warning when Given Mode Switch Point is part of a synchronous mode machine instance and the runnable containing it is not mapped to any OsTask.
22	SAFERTE_WARN_MC002	RTE Generator shall throw a warning when runnable is not mapped to the given ParameterDataPrototype.
23	SAFERTE_WARN_CLS001	RTE Generator shall throw a warning when BswCallType is 'REGULAR' or 'CALLBACK' and BswExecutionContext is 'TASK' for the BswModuleEntry used in the role BswModuleClientServerEntry.encapsulatedEntry.
24	SAFERTE_WARN_CLS002	RTE Generator shall throw a warning when BswCallType is 'SCHEDULED', and BswExecutionContext is 'TASK' for the implementedEntry mapped to the BswSchedulableEntity.
25	SAFERTE_WARN_CLS003	RTE Generator shall throw a warning when BswCallType is 'REGULAR', and BswExecutionContext is 'TASK' for the implementedEntry mapped to the BswCalledEntity.
26	SAFERTE_WARN_CLS004	RTE Generator shall throw a warning when BswModuleEntry Mapped to BswModuleEntity is not unique inside a single BswModuleDescription.
27	SAFERTE_WARN_CLS005	RTE Generator shall throw a warning when the server task does not have the lower priority than the client task in Synchronous Client Server Communication, if timeout is configured.
28	SAFERTE_WARN_0029	RTE Generator shall throw a warning when no RunnableEntities are mapped to an OsTask.
29	SAFERTE_WARN_0030	RTE Generator shall throw a warning when no SchedulableEntities are mapped to an OsTask.



## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
145 / 146

Sl. No.	Warning ID	Description
30	SAFERTE_WARN_0031	RTE Generator shall throw a warning when FlatMap is not configured for a given VariableDataPrototype.
31	SAFERTE_WARN_0032	RTE Generator shall throw a warning when InitValue is not configured for ArTypedPerInstanceMemory.
32	SAFERTE_WARN_0033	RTE Generator shall throw a warning when the period of the Timing Event is not equal to the product of tick duration and Schedule Table Duration. Integrator shall ensure correct run-time behavior by configuring additional ExpiryPoints actions at appropriate offsets in the Schedule Table.
33	SAFERTE_WARN_0034	RTE Generator shall throw a warning when the period of the BSW Timing Event is not equal to the product of tick duration and Schedule Table Duration. Integrator shall ensure correct run-time behavior by configuring additional ExpiryPoints actions at appropriate offsets in the Schedule Table.
34	SAFERTE_WARN_0035	RTE Generator shall throw a warning when RteEventPredecessorSyncPointRef for the RteSyncPoint is not configured.
35	SAFERTE_WARN_0036	RTE Generator shall throw a warning when BswModuleEntity calls SchM APIs but does not conform to any of the following criteria required in a Multi-Partition environment: 1. BswModuleEntity is not mapped to an EcucPartition directly/indirectly via the path BswEvent->BswDistinguishedPartition->EcucPartition or BswEvent->OsTask->OsApplication->EcucPartition. 2. BswModuleEntity's implementedEntry is not referred from a BswDirectCallPoint created inside another BswModuleEntity with a mandatory contextLimitation (BswDistinguishedPartition) reference.
36	SAFERTE_WARN_0037	RTE Generator shall throw a warning when RteSyncPoint is not mapped to any task.
37	SAFERTE_WARN_0038	RTE Generator shall throw a warning when all the OsTasks mapped to the RteSyncPoint are in same core.
38	SAFERTE_WARN_0039	RTE Generator shall throw a warning when RteEventSuccessorSyncPointRef for an RteSyncPoint is not configured.
39	SAFERTE_WARN_0040	RTE Generator shall throw a warning when all the BswOsTasks mapped to RteSyncPoint are in same core.
40	SAFERTE_WARN_0041	RTE Generator shall throw a warning when the ValueSpecification is not TextValueSpecification for a DataPrototype typed by an ImplementationDataType that references a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE.
41	SAFERTE_WARN_0042	RTE Generator shall throw a warning when RteBswEventSuccessorSyncPointRef for an RteSyncPoint is not configured.
42	SAFERTE_WARN_0043	RTE Generator shall throw a warning when RteBswEventPredecessorSyncPointRef for the RteSyncPoint is not configured.
43	SAFERTE_WARN_MC003	RTE Generator shall throw a warning when RteMeasurementSupport is configured for the DataPrototype which is not referred in FlatMap container and SwCalibrationAccess for the DataPrototype is set to NOT_ACCESSIBLE.
44	SAFERTE_WARN_PIM001	RTE Generator shall throw a warning when SubElementMapping is configured in DataPrototypeMapping.
45	SAFERTE_WARN_PIM002	RTE Generator shall throw a warning when ArgumentMapping is configured in ClientServerOperationMapping.
46	SAFERTE_WARN_TRG001	RTE Generator shall throw a warning when for a ReleasedTrigger, the SwImplPolicy is QUEUED and QueueLength is 0.

## Rte User Manual

Document number  
(DOC NO)

SHT/SHTS  
146 / 146

Sl. No.	Warning ID	Description
47	SAFERTE_WARN_TRG002	RTE Generator shall throw a warning when for a given BswInternalTriggeringPoint, the SwImplPolicy is QUEUED and QueueLength is 0.
48	SAFERTE_WARN_0045	Rte Generator shall throw a warning for every connection between SwComponentPrototypes mapped to different partitions, the value of VariableAccess.scope is set to VariableAccessScopeEnum.communicationIntraPartition.
49	SAFERTE_WARN_0046	RTE Generator shall throw a warning when multiple P-PORT consisting of different HANDLE-OUT-OF-RANGE values are connected to an R-PORT.
50	SAFERTE_WARN_0047	RTE Generator shall throw a warning when generated API cannot the Nesting Order as calling BswModuleEntity is not mapped to OsTask in configuration.
51	SAFERTE_WARN_0048	RTE Generator shall throw a warning when MinimumStartInterval is configured for Runnable Entities and Schedulable Entities.
52	SAFERTE_WARN_0049	RTE Generator shall throw a warning when QueueLength parameter under BswModeSenderPolicy is not configured for a given providedModeGroup in case of Asynchronous Mode Switch Communication and mode users are in more than one partition.
53	SAFERTE_WARN_0050	RTE Generator shall throw a warning when QueueLength parameter under BswModeSenderPolicy is not configured for a given providedModeGroup in case of Synchronous Mode Switch Communication.
54	SAFERTE_WARN_CLS006	RTE Generator shall throw a warning when timeout is either not configured or configured as zero for a synchronous client server communication when both client and server are in different partitions.
55	SAFERTE_WARN_0051	RTE Generator shall throw a warning when only single EcucCoreDefinition/EcucPartition is configured RTE.
56	SAFERTEPB_WARN_0001	Safe RTE PB Generator shall throw a warning when only one predefined variant is referred in RtePostBuildUsedPredefinedVariant