HYUNDAI AUTOEVER

AUTOSAR Rte User Manual

DOC. NO

SCOPE OF APPLICATION All Project/Engineering
Responsibility: Classic AUTOSAR Team

File Name: RTE_UM.pdf

Creation SJ Noh 2022/12/01
Check HM Kim 2022/12/01
Approval JH Jung 2022/12/01

Edition Date: 2022/12/01
Document Management System

HYUNDAI AutoEver

Document Change History

Date (YYYY-MM-DD)	Ver.	Editor	Chap	Description (before -> after revision)
2015-01-02	0.1	Seongmin Kim		• Draft Version of document
2015-01-30	1.0.0	Daehwan. Lim		• Release Initial Version of document
2015-05-22	1.1.0	Daehwan. Lim		• Update for release 1.5.0 version of RTE generator
2015-06-19	1.2.0	Seongmin Kim		• Update for release 1.5.1 version of RTE generator
2015-07-29	1.3.0	JeongSu Lim		• Update for release 1.5.2 version of RTE generator
2015-08-31	1.3.1	Daehwan Lim		• Update for release 1.5.3 version of RTE generator
2015-09-30	1.3.2	Daehwan Lim		• Update for release 1.5.4 version of RTE generator . Add design decision for Rte_Mode . Add the description for behavior of partition start among multi partitions. . Remove parameter regarding OsInteraction (Vendor Specific Parameter) and add the description to handle the synchronized offset for OsAlarm based on Specific Counter . Add the description regarding configuration in RteOsInteraction and introduce the new configuration parameter (Vendor Specific) . Add the new error message on generation
2015-10-20	1.3.3	Seongmin Kim		• Update for release 1.5.5 version of RTE generator . Add a new parameter RteEndToEndProtectionWrapperMode . Add new error messages on generation . Add ChangeLog for release 1.5.5 version
2015-12-16	1.3.4	JeongSu Lim		• Update for release 1.5.6 version of RTE generator . Add ChangeLog for release 1.5.6 version
2016-01-21	1.3.5	JeongSu Lim		• Update for release 1.5.7 version of RTE generator . Add ChangeLog for release 1.5.7 version
2016-04-28	4.1.1	Daejun Park	4.3	• Update for release 4.1.1 version of RTE generator
			7.2	• Update Generator Error Message
2016-05-11	4.1.2	Daejun Park	4.3	• Update for release 4.1.2 version of RTE generator

2016-05-30	4.2.0	JeongSu Lim		• Update for release 4.2.0 version of RTE generator
2016-06-03	4.2.1	JeongSu Lim		• Update for release 4.2.1 version of RTE generator
2016-08-16	4.2.2	Daejun Park	4.3.1, 5.4.1	• Add the description of RteUsedOsEventRef and RteBswUsedOsEventRef parameters
2016-08-19	4.3.0	Daejun Park JeongSu Lim	4.3	• Update for release 4.3.0 version of RTE generator
			7.2	• Update Generator Error Message
			4.4, 5.1	• Support HandleOutOfRange
2016-10-21	4.3.1	JeongSu Lim Daejun Park	4.1	• Add description of limitation for using RTE API.
			4.3	• Update for release 4.3.1 version of RTE generator
			7.1	• Add description of generator option for Rte_MemMap.h File.
			7.2	• Add description of generator error messages
2016-12-01	4.3.2	JeongSu Lim Daejun Park	4.3	• Update for release 4.3.2 version of RTE generator
			5.5	• Add description of limitation for using SynchronizedOffset
			7.1	• Add description of generator option for IOC::GenEcudOs
			7.2	• Add description of generator error messages
2017-02-24	4.3.3	Daejun Park Daehwan Lim	4.3	• Update for release 4.3.3 version of RTE generator
			8.6	• Add descrition of Synchronized Offset
2017-06-09	4.4.0	JeongSu Lim Seongmin Kim	4.3	• Change Log updated
			4.4 5.3 5.4	• Exclusive Area unsupported items added
			7.1	• Generator Option added
			7.2	• Generator Error Message added
2017-08-21	4.4.1	KiYoung Yun Seongmin Kim	4.3	• Change Log updated
			4.4 4.5	• Exclusive Area unsupported items added • Regarding Limitation & Deviation, validation error numbers added and some unsupported items changed
			5.3 5.4	• Exclusive Area option descriptions added
			7.1	• Option descriptions about unsupported features (Generation Phase) arranged
2017-11-09	4.4.2	Seongmin	4.4	• Regarding Extended Task, event units and

		Kim	8.1	task units restrictions specified • Interrupt Decoupling Guide revised
2017-12-26	4.5.0	KiYoung Yun Seongmin Kim	2	• Rte 4.2.2 added
			4.3	• Change Log updated
			4.4.2 4.5.2	• Data Transformation details added
			8.7	• Guide for RteEvent added
2018-01-26	4.5.1	KiYoung Yun	4.3	• Change Log updated
2018-03-22	4.6.0	KiYoung Yun	2	• Rte 4.3.1 added
			4.3	• Change Log updated
			4.4 4.5	• Limitation and Deviation related to ApplicationPrimitiveDataType of categories COM_AXIS, CURVE, and MAP added
2018-06-29	4.7.0	KiYoung Yun	4.3	• Change Log updated
			4.4.2	• Limitation related to ApplicationPrimitiveDataType changed
			4.4.2 6.1.1 6.3.1	• Data Transformation details added
			8.4	• Initial Mode Runnable execution details within SchM_Init and Rte_Start added
2018-10-31	4.7.1	KiYoung Yun	4.3	• Change Log updated
2019-05-08	4.8.0	KiYoung Yun	4.3	• Change Log updated and format changed
			4.4.2.3.1	• Disabled Mode details added
2019-06-28	4.8.1	KiYoung Yun	4.3	• Change Log updated
			4.4.2.7.2	• Limitation related to FUNCTION_REFERENCE added
2019-10-18	4.9.0.0	KiYoung Yun	4.3	• Change Log updated
			4.4.2.8.1 4.5.2 5.1	• VFB Tracing details changed
			5.6 5.10 6.3.3	• Manual and PDF configuration items changed
2020-04-24	4.10.0.0	KiYoung Yun	4.3	• Change Log updated
			4.4.2.2.3	• Standard Trigger communication restrictions added
			7.1	• -NoRteReceiverPullCB option descriptions added
2020-06-26	4.10.1.0	KiYoung Yun	4.3	• Change Log updated
			4.4.2.2.2	• Client Server communication restrictions added
			7.2	• Generator Error Message added

2020-09-25	4.10.2.0	KiYoung Yun	4.3	• Change Log updated
			4.4.2.2.1	• DATA_REFERENCE restrictions added
			4.5.2.2	• Deviation related to SetRelAlarm callouts added
2020-12-21	4.10.3.0	KiYoung Yun	4.3	• Change Log updated
			7.1	• ParameterDirection options added
			4.4.2.2.1	• Validation Error numbers added to Limitation • Implicit Sender Receiver communication restrictions added
			4.4.2.3.1	• ExtendedTask configuration restrictions added
2021-04-07	4.10.4.0	Gongbin Lim	4.3	• Change Log updated
2021-07-28	4.10.5.0	KiYoung Yun	4.3	• Change Log updated
			5.5.1	• RteExpectedActivationPosition AlarmSetMethodIsRelative descriptions improved
			6.3.1 6.3.3	• Unsupported return value descriptions removed • Transformer-related return value descriptions added
			8.6.1	• Descriptions related to Synchronized Offset added
2021-11-23	4.10.6.0	KiYoung Yun	4.3	• Change Log updated
			All	• AUTOEVER format changed
2021-12-28	4.10.7.0	KiYoung Yun	4.3	• Change Log updated
2022-06-24	4.11.0.0	KiYoung Yun	4.3	• Change Log updated
			4.4.2.2.1	• REPLACE-BY-TIMEOUT-SUBSTITUTION-VALUE restrictions added
			4.4.2.2.3	• 1:N Standard Trigger Limitation removed
			4.5.2.2	• REPLACE-BY-TIMEOUT-SUBSTITUTION-VALUE Deviation added
			7.2	• Generator Error Message changed and added
2022-08-11	4.11.1.0	KiYoung Yun	4.3	• Change Log updated
2022-12-01	4.11.1.1	SeungJin Noh	4.3	• Change Log updated

Table of Contents

1. Overview

This document explains configuration and various matters for code generation, which is RTE (Runtime Environment) Layer based on the application design document (Software Component Description arxml) that the AUTOSAR standard development methodology designed. It is created based on the AUTOSAR standard SRS/SWS, and for more functional details when using the module, please see the reference documents below.

Each configuration category is defined as follows.

- Changeable (C): Items that can be configured by users
- Fixed (F): Items that cannot be changed by users
- NotSupported (N): Items that are not used

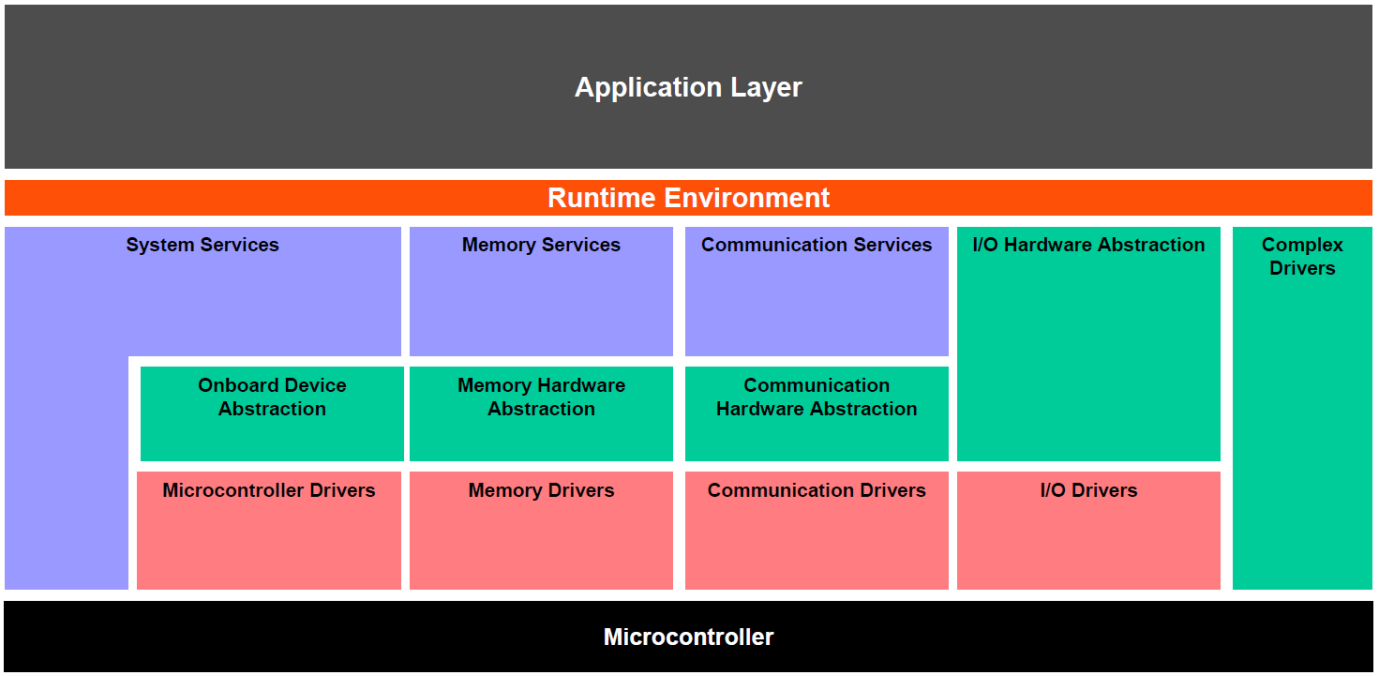
2. Reference

Sl. No.	Title	Version
1.	AUTOSAR SWS RTE.pdf	3.2.0
2.	AUTOSAR TPS SoftwareComponentTemplate.pdf	4.2.0
3.	AUTOSAR TPS SystemTemplate.pdf	4.2.0
4.	AUTOSAR EXP InterruptHandlingExplanation.pdf	1.0.2
5.	AUTOSAR SWS RTE.pdf	4.2.2
6.	AUTOSAR SWS RTE.pdf	4.3.1

3. AUTOSAR System

3.1 Overview of Software Layers

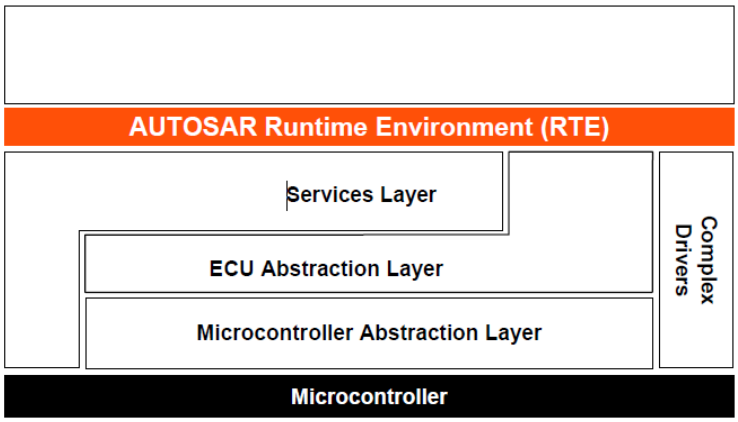
The AUTOSAR platform is a layered architecture as illustrated below. The AUTOSAR platform can be divided into Service Layer, ECU Abstraction Layer, Complex Device Drivers and Microcontroller Abstraction Layer.



3.2 AUTOSAR Runtime Environment (RTE)

RTE handles communication service between AUTOSAR Software Components and/or AUTOSAR Sensor/Actuator components of application software or between application software and basic software module (i.e., OS, Communication Service, services, etc.) by providing glue lyaer between numerous Application SW-Components (i.e., Application SW-Component and Basic Software module) integrated in a specific ECU.

- Provide a communication infrastructure for software components
- Arrange for real-time scheduling of software components


AUTOSAR Runtime Environment (RTE)
Services Layer
ECU Abstraction Layer
Microcontroller Abstraction Layer
Complex Drivers
Microcontroller

4. Product Release Notes

4.1 Overview

This chapter provides the release information on Hyundai AutoEver RTE products, describing the features and restrictions of different release versions of the RTE software product.

4.2 Scope of the release

Every detail of this document is limited to Hyundai AutoEver RTE Product (Rte.exe) Verison 1.5.7.

※ RTE PDF (Parameter Definition File) Version 1.5.4

4.3 Change Logs

4.3.1 Version 1.5.0

- Changed for Enumeration Define statement of DataType to show only in the relevant Application Type Header
- Changed to remove EcuExtract CommandLine Option and use EcuExtractRef of EcucValueCollection
- Support to generate Mode Definition and Type Definition through IncludedModeDeclarationGroup configuration
- Support to generate Enumeration Define statement of DataType through IncludedDataTypeSet configuration
- Validation Error strengthened related to EcucPartition and ComCallback configuration errors
- Optimization Option related to Rte_MemMap.h Code arranged
- SenderReceiver Logic Refactoring.
- DET about ApplicationError of ClientServer added
- Part of MISRA-C Rule Violation corrected
- Category to support ECUC Configuration Parameter settings restrictions feature added

4.3.2 Version 1.5.1

- Initialization order within Rte_Start_<Partition> function arranged
- Extern declaration of PerInstance Memory RAM parameter generated additionally to a separate header file
- Code readability enhanced through Code Indentation related to Rte Start / Stop
- Rte_MemMap.h arranged to facilitate dividing MemorySection by partition
- Misra C Violation corrected additionally
- Validation Error added and words edited

4.3.3 Version 1.5.2

- Validation processing related to Misra C Violation added (11.4, 8.8).
- Version Check of RTE Header enhanced (Autosar Version, Sw Version)

- Blank Line redundancy removed and Code Indentation arranged
- SenderReceiver Interface RamSize optimization (parameter generation optimization when not using Rte_Read and Rte_IRead)
- ClientServer Interface RamSize optimization (ClientServer Flag arranged during InterPartition)
- Unnecessary Warning removed (Task Mapping Warning removed when Unconnected Server and CanBeInvokedConcurrently are TRUE)
- Validation enhanced (Error occurs if StartOnEventRef of TimingEvent and BswTimingEvent is absent.)
- Unnecessary Memory Section Code generation removed
- Command Line Option arranged
- Feature added to configure and allow E_OK in Application Error
- Type Casting added to Data Parameter part of Com_SendSignal, Com_SendDynSignal, Com_UpdateShadowSignal, Com_ReceiveSignal, Com_ReceiveDynSignal, and Com_ReceiveShadowSignal

4.3.4 Version 1.5.3

- Feature added to immediately notify a waiting client if a server partition terminates when using Client-Server communication (When setting WaitOsEventRef, one more OsEvent with the PartitionTerminated postfix for the configured OsEvent must be created for the client to refer to operating OsTask.)
- Task prioritization for Client-Server communication (see 8.3)

4.3.5 Version 1.5.4

- MemorySection of Rte internal parameter arranged
- Client-Server Communication operation connected with different operations supported
- Validation related to RteWaitOsEventRef for ClientServer communication added
- RTE_E_COM_STOPPED return value processing of Rte_Write API added
- Polyspace RunTime Violation corrected
- RamSize optimization related to Rte_COMCbk Callback function (using parameter stack)
- MCU Dependenct Code of <SWC>_MemMap.h removed

4.3.6 Version 1.5.5

- Validation error which occurs when operation that a port refers to and operation that a server call point refers to do not match in Client-Server communication added
- Polyspace Runtime Violation corrected
- E2E PW/Callout option configuration method changed
- Unnecessary Exclusive Area code of use in Inter-Runnable Variable optimization

4.3.7 Version 1.5.6

- RamSize of SenderReceiver Communication Logic optimization (changed to use common buffer if optimization is available for each RPort to generate buffer for DataElement in case of

1:N Case)
- HandleTimeoutType feature implemented (feature changing DataElement value to initial value when Com Timeout occurs)
- Problem in which redundant code or dummy code related to OsAlarm is generated within SchM_Init function corrected
- Validation error that occurs when using WaitPoint in Asynchronous Client-Service communication enhanced
- Code generation error that occurs when ModeManager and ModeUser are mixed during the processing of Initial Mode of InterPartition-based ModeSwitch corrected

4.3.8 Version 1.5.7

- Code generation error in case of InitValue configuration in InterRunnableVariable using ApplicationDataType (error occurring during code generation after InitValue configuration in InterRunnableVariable with ApplicationDataType corrected to work normally)
- MemorySection of <SWC>_MemMap.h arranged (added to generate <PREFIX>_START_SEC_CONST_<ALIGNMENT> that was not generated previously and to generate BOOLEAN in <ALIGNMENT> that was not processed previously)
- Problem in which Copy_IWrite and Copy_IRead GlueCode are not generated in Task Body when using Implicit SenderReceiver in Runnable operated with OperationInvokedEvent corrected (If runnable callout codes are arranged in rows due to indentation, GlueCode was not generated, but it was corrected for codes to work normally even when arranged in rows.)
- CDS Structure Definition and parameter processing error of ArTypedPerInstanceMemory with Array DataType corrected (as for ArTypedPerInstanceMemory of the existing Array DataType, a pointer for Array address was generated in CDS Structure Definition, but it was corrected to generate a pointer for Array BaseType.)
- Rte_IFeedback to be used in DataWriteCompletedEvent Runnable and Rte_IStatus API to be used in DataReceiveErrorEvent Runnable added (added to generate Rte_IFeedback for DataWriteCompletedEvent Runnable and Rte_IStatus API for DataReceiveErrorEvent Runnable that were not generated before)
- Validation for M:N Connection enhanced (M:N Check was done only for IntraEcu Connection before, but it was enhanced to enable M:N Connection Check to include InterEcu Connection.)

4.3.9 Version 4.0.0

- Changed for GetCoreID Function of Rte_Start and SchM_Init Function code generation conditions and Os GetCoreID Function code generation conditions to match (conditions changed from the OsApplicationCoreAssignment configuration or not to two or more of OsOs NumberOfCores)
- Validation logic added to check whether values of Period and RteActivationOffset are valid (checking whether it is multiples of OsCounter OsSecondsPerTick of Period and RteActivationOffset or belongs to 0 ~ OsSecondsPerTick * OsCounterMaxAllowedValue)
- SenderReceiver Communication performance optimization (performance optimized by removing SuspendOSInterrupts/ResumeOSInterrupts if runnables for write and read in the same partition have the same priority or share one InternalOsResource)
- Code generation error when SenderReceiver Interface name begins with Signal_ or Return_ corrected (changed to add Rte_Prefix to IOC API generation code regardless of interface

name)
- Optimization structure for OsEvent Clear in Extended OsTask applied

4.3.10 Version 4.1.0

- Changed to create ShortLabel Define statement if CompuMethod VT value is Invalid C Identifier
- Applied to process IDENTICAL as default if there is no category in CompuMethod

4.3.11 Version 4.1.1

- Error Message format changed and elaborated
- Validation added when setting SynchronousServerCallPoint for the same Port/Operation in two runnables or more

4.3.12 Version 4.1.2

- Trigger Interface generation error corrected in Extended Task

4.3.13 Version 4.2.0

- Data Consistency guarantee code applied to RteEvent, BswEvent
- User Manual Updated for Generation Error and Limitation

4.3.14 Version 4.2.1

- Event Flag variable related to Sender Receiver separated
- Inter Runnable Variable generation error corrected

4.3.15 Version 4.2.2

- User Manual descriptions of RteUsedOsEventRef and RteBswUsedOsEventRef parameter limitations supplemented

4.3.16 Version 4.3.0

- **New feature**
  - HandleOutOfRange feature of SenderReceiver Interface supported
    Feature checking the range of data value and ignoring (Ignore), changing to Min/Max value (Saturate), changing to Init Value (Default) or changing to Invalid Value according to user settings if the data value is out of the range

- **Improvements**

- MISRA-C 16.10 Rule If a function returns error information, then that error information shall be tested, violation items corrected
- When processing Initial Mode On Entry of ModeSwitch Interface, Direct Function Call supported
- RteTaskComMapping configuration error Validation feature added in No Partition, Single Partition, and Multiple Partition conditions
- Validation Error feature of Trigger Interface Event configuration error added
- Validation feature of configuration error coming from the discrepancy between SwImplPolicy and QueueLength of Trigger Interface added

4.3.17 Version 4.3.1

- **New feature**
  - Rte_MemMap.h feature added for Bolero, Rh850 MCU

Rationale	Rte_MemMap.h feature unsupported in Bolero and RH850 MCU
Impact on Behavior	RTE Memmory Section brought from Rte_MemMap.h, not from MemMap.h
Impact on Settings	GenMemMapOption::Ghs Option must be added as RTE Generator Option to a controller using Bolero and Rh850 MCU
Required ASW actions	None

- **Improvements**
  - Disabled Mode error corrected in Timing Event based on Extended OsTask

Rationale	Relevant code not being generated when setting DisabledMode in TimingEvent based on Extended OsTask
Impact on Behavior	Disabled Mode working normally in Timing Event based on Extended OsTask
Impact on Settings	GenMemMapOption::Ghs Option must be added as RTE Generator Option to a controller using Bolero and Rh850 MCU
Required ASW actions	None

  - Abnormal operation of SynchronizedActivateOffset feature corrected

Rationale	Increment value of SetRelAlarm is used as 0 in a particular case according to counter value at Rte_Start. In this case, alarm does not work due to an error.
Impact on Behavior	Alarm working normally
Impact on Settings	None
Required ASW actions	None

  - Signal Reception Hook function location error corrected

Rationale	Signal Reception Hook is called before ComReceiveSignal. In terms of specifications, Signal Reception Hook must be called after Com_ReceiveSignal.

Impact on Behavior	Signal Reception Hook is called after ComReceiveSignal.
Impact on Settings	None
Required ASW actions	None

4.3.18 Version 4.3.2

- **New feature**
  - ■ New Generator Option –IOC::GenEcudOs added to additionally generate IOC Configuration in Os Ecud Arxml file

Rationale	Freescale OS can receive one Ecud Arxml file with Generator input. The existing IOC Option is unavailable as it generates IOC Configuration as a separate file. A new option is added to integrate and generate OsConfiguration in one file.
Impact on Behavior	None
Impact on Settings	–IOC::GenEcudOs must be added as Rte Generator option when using the new feature
Required ASW actions	None

- **Improvements**
  - ■ Feature added to check CPU Byte Order and RTE Byte Order when using E2E feature

Rationale	A problem could occur if Generator Option for RTE ByteOrder is incorrect when using E2E Serialize/Deserialize. Feature was added to early detect a problem by checking CPU Byte Order.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Improved probability that code can be generated in a changed order despite the same settings during generation

Rationale	The probability was that API Code of Rte_Write would be generated in a changed order despite the same settings. Feature was added so that it can be output in the same order through sorting.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ InitialMode related to OnEntry Event generation error improved

Rationale	When setting DirectFunctionCall related to InitialMode OnEntry Event, generation function name error occurred.

Impact on Behavior	Compile error occurred
Impact on Settings	None
Required ASW actions	None

■ Abnormal operation of SynchronizedActivateOffset feature improved

Rationale	Calculation error could occur if interrupts occur during SynchronizedActivateOffset calculation.
Impact on Behavior	In case of excessive interrupts, there could be a difference from the time when task offset is configured.
Impact on Settings	None
Required ASW actions	None

■ Some unnecessary warning messages requiring task mapping removed

Rationale	To remove some unnecessary warning message (WRN 58, 66) outputs requiring task mapping
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.19 Version 4.3.3

- **Improvements**

■ Corrected a problem where SetEvent could be called before ActivateTask when using non-periodic Extended Task

Rationale	If Alarm set before ActivateTask in Rte_Start expires before ActivateTask, SetEvent is called and generates ErrorHook.
Impact on Behavior	ErrorHook possibility removed
Impact on Settings	None
Required ASW actions	None

■ Corrected a problem where Task Body for RTE processing based on Extended Task omits OsEvent

Rationale	If several OsEvents are connected to one OsTask, OsEvent that has not occurred yet could be cleared, which could omit processing OsEvent that has actually occurred.
Impact on Behavior	Possibility of omitting OsEvent processing removed
Impact on Settings	None

Required ASW actions	None

■ Corrected error that could occur when setting TextValueSpecification to InitValue of ExplicitInterRunnableVariable

Rationale	Not processed when InitValue type is TextValueSpecification
Impact on Behavior	None
Impact on Settings	Configuration available when InitValue type is TextValueSpecification
Required ASW actions	None

■ Validation Error added for RteBswRequiredModeGroupConnection configuration error

Rationale	RteBswRequiredModeGroupConnection configuration error not checked
Impact on Behavior	None
Impact on Settings	Need to change configuration if generation error occurs
Required ASW actions	None

■ Validation Error added for a case where redundant SystemSignalRef exists in SenderRecRecordElementMapping of SenderReceiverToSignalGroupMapping

Rationale	Redundancy not checked for SystemSignalRef of SenderRecRecordElementMapping
Impact on Behavior	None
Impact on Settings	Need to change configuration if generation error occurs
Required ASW actions	None

■ Generation c code improved to define functions in one code section

Rationale	To improve all functions to be defined in one code section, away from functions being divided into each code section to be defined
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Changed to Rte Vendor Id 76

Rationale	To change to Vendor Id 76 in Rte.h
Impact on Behavior	None
Impact on Settings	Needed to change to /Rte/CommonPublishedInformation/VendorId = 76
Required ASW	None

actions	

4.3.20 Version 4.4.0

- **Improvements**
  - IOC support for InterPartition of TriggerInterface

Rationale	To improve IOC use method for effective operation of IOC when Trigger is executed through IOC during inter-partition trigger use
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - Added RteImmediateBufferUpdate False feature among Implicit Sender Receiver features

Rationale	To add feature to allow using a method of actually sending data before and after task in addition to a method of actually sending data before and after the existing runnable
Impact on Behavior	None
Impact on Settings	In case of using a newly added feature, –ImmediateBufferUpdate=false must be added as RTE Generator Option. No configuration change if maintaining the existing method.
Required ASW actions	None

  - Added statement banning RTE_ALLOW_CROSS_HEADER_INCLUSION symbol

Rationale	To add statement banning RTE_ALLOW_CROSS_HEADER_INCLUSION symbol, which is unavailable in the application, as an annotation in Rte_<SWC>.h
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - Partition Lifecycle API improved

Rationale	To improve Lifecycle API code generation method in Generator for code arrangement and smooth maintenance
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.21 Version 4.4.1

- **New feature**
  - ■ Generator Validation added for unsupported features

Rationale	Generator could generate a file without an error and work wrong for unsupported features. Generation limited through validation
Impact on Behavior	None
Impact on Settings	Configuration change needed in case of an error (unsupported feature configuration removed)
Required ASW actions	None

- **Improvements**
  - ■ Rte Generator execution speed improved

Rationale	To improve ineffective operation method (API complexity) in Generator
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Alignvar removed to prevent compile warning

Rationale	For Generator not to output alignvar in Rte_MemMap.h when using Green Hills compiler
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ PerInstanceMemory API improved

Rationale	To improve PerInsatnceMemory API code generation method in Generator for code arrangement and smooth maintenance
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Modified E2EPW_Read/Write to address Com_PackAndValidateSigGroup API prototype changes after Com Version 2.2.0

Rationale	Compile error occurred as the number of factors of Com_PackAndValidateSigGroup API changed after Com Version 2.2.0
Impact on Behavior	None
Impact on Settings	None

Required ASW actions	None

■ Corrected a problem of Out of memory in case of self-reference in settings

Rationale	In case of self-reference due to wrong settings, accurate error message is not returned and out of memory occurs.
Impact on Behavior	None
Impact on Settings	Need to change configuration if error occurs
Required ASW actions	None

■ Improved Division by zero error generation error if TimingEvent Period is 0

Rationale	To add validation to a case where error occurs when TimingEvent Period is 0
Impact on Behavior	None
Impact on Settings	Need to change configuration if error occurs
Required ASW actions	None

■ Validation added to a case where supportsAsynchronousModeSwitch is false when using Mode Switch Interface

Rationale	To add error code for a case where supportsAsynchronousModeSwitch is false and Task Mapping is not available when using Mode Switch Interface
Impact on Behavior	None
Impact on Settings	Need to change configuration if error occurs
Required ASW actions	None

4.3.22 Version 4.4.2

- **Improvement**

■ Manual update

Rationale	- Interrupt Decoupling Guide improved (see 8.1) - Extended Task support range specified (see 4.4)
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.23 Version 4.5.0

- **New feature**
  - ■ Transformation feature developed

Rationale	Data Transformation feature development to address E2E feature
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

- **Improvements**
  - ■ Implicit Sender-Receiver compile error corrected

Rationale	- Copy_IRead API definition is generated more than one when using DataFilter and HandleTimeoutType. - Redundant prototypes of the same Copy_IRead API occur because API redundancy examination method is base on SWC. - Codes that are both using and not using Suspend/Resume Interrupt are generated because context of other SWC is not considered during critical section distinction.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Solved a problem of inserting multiple GlueCodes according to Runnable Symbol in NonQueuedImplicitSenderReceiver

Rationale	When Generator inserts GlueCode, GlueCode is unintentionally inserted into a runnable with a similar name.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Improved abnormal code generation that occurred when multiple PPorts used the same External Trigger Interface in SWC and their sink was mapped in the same task

Rationale	The relevant Runnable is implemented by adding a flag for RteEvent
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Supported GroupSignal with UINT8_N ComSignalType

Rationale	Compile error occurred when setting GroupSignal with UINT8_N ComSignalType
Impact on	None

Behavior	
Impact on Settings	None
Required ASW actions	None

■ Compile Warning improved

Rationale	- Compile warning occurred due to code generation even when not using Rte_GaaEventFlag variable. - Compile warning occurred as Rte_Mode API the same code regardless of Enhanced API and created unnecessary variables. - Compile warning occurred as compiler optimization was applied to state variables for preventing reentry during mode switch in Rte_Switch and SchM_Switch.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Error information elaborated in Validation

Rationale	- In case of BswSchedulableEntity, not Direct Function Call, not being RteBswEventToTaskMapping - Target Mode Declaration of SwcModeSwitchEvent not configured - M:N communication connected - SenderReceiverToSignalGroupMapping for Union Data Type unsupported - PositionInTask configuration error in case of Mode Switch Event of Direct Function Call method
Impact on Behavior	None
Impact on Settings	Need to change configuration if error occurs
Required ASW actions	None

4.3.24 Version 4.5.1

- **Improvements**

■ Header File Inclusion of Transformer module changed

Rationale	- To change Header Inclusion of Rte.c to <module name>.h
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Transmission Acknowledge configuration without Timeout value supported

Rationale	When setting Transmission Acknowledge, Transmission Acknowledge

	is not recognized if Timeout value is not designated.
Impact on Behavior	None
Impact on Settings	Transmission Acknowledge configuration without Timeout value available
Required ASW actions	None

■ Error information elaborated in Validation

Rationale	- When InitValue is undefined in PerInstanceParameter
Impact on Behavior	None
Impact on Settings	Need to change configuration if error occurs
Required ASW actions	None

4.3.25 Version 4.6.0

- **New features**

■ Transformation Profile 11 support

Rationale	Data Transformation feature development to address E2E Profile 11
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Support for categories COM_AXIS, CURVE, and MAP of ApplicationPrimitiveDataType

Rationale	Support for categories COM_AXIS, CURVE, and MAP of ApplicationPrimitiveDataType
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

- **Improvements**

■ When setting TRANSFORMER-ERROR-HANDLING, Transformer error value is sent.

Rationale	When setting TRANSFORMER-ERROR-HANDLING, RTE_E_HARD_TRANSFORMER_ERROR / RTE_E_SOFT_TRANSFORMER_ERROR value, transformerError parameter of RTE API, is saved.
Impact on Behavior	When setting TRANSFORMER-ERROR-HANDLING,Transformer error value is saved in transformerError parameter of RTE API.
Impact on Settings	None
Required ASW	As Transformer error value, not RTE_E_HARD_TRANSFORMER_ERROR /

actions	RTE_E_SOFT_TRANSFORMER_ERROR value, is saved in transformerError parameter of RTE API when setting TRANSFORMER-ERROR-HANDLING, ASW logic needs to be changed accordingly.

■ During 1:N Sender Receiver communication, missing buffer declaration used by RTE API improved

Rationale	If using 1:N Sender Receiver communication mixed with Implicit / Explicit Sender Receiver API, buffer declaration used in RTE API could be omitted.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ API code related to Inter-Runnable Variable improved

Rationale	To improve Inter-Runnable Variable API code generation method in Generator for code arrangement and smooth maintenance
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.26 Version 4.7.0

- **New features**

■ Support for category BOOLEAN of ApplicationRecordElement in ApplicationRecordDataType

Rationale	Category BOOLEAN of ApplicationRecordElement in ApplicationRecordDataType unsupported
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Support for categories COM_AXIS, CURVE, and MAP of ApplicationPrimitiveDataType

Rationale	Support for categories COM_AXIS, CURVE, and MAP of ApplicationPrimitiveDataType for PerInstanceParameter in BswModuleDescription and Validation added
Impact on Behavior	None
Impact on Settings	Need to change configuration if error occurs
Required ASW actions	None

- **Improvements**

■ When using Transformer, Init Value supported

Rationale	When using Transformer, Init Value unsupported
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ A problem of not generating Trigger API corrected

Rationale	Some Rte_Trigger API is not generated when ShortName of PPortPrototype and Trigger ShortName of Trigger Interface are equal in other SWC. Some Rte_IrTrigger is not generated when ShortName of Runnable and ShortName of InternalTriggeringPiont are equal in other SWC.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ In case of unused transformation feature, compile error corrected when error handling value is set to TRANSFORMER-ERROR-HANDLING

Rationale	If error handling value is set to TRANSFORMER-ERROR-HANDLING but transformation feature is not used, code using a variable undeclared in Rte API is generated.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ During 1:N NonQueued Sender Receiver communication, compile error generation corrected in case of different Init Value of RPort

Rationale	During 1:N NonQueued Sender Receiver communication, compile error occurs as a variable with the same name is declared in different initial value several times in case of different Init Value of RPort.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Validation corrected

Rationale	- In case of Init Value settings through Application Value Specification, SwArraysize is required for all categories. - In case of XfrmImplementationMappingSet settings, Data Mapping

	configuration is required. - In case of TransformationISignal in Signal/SignalGroup, ComSignal / ComSignalGroup is required for the relevant Signal/SignalGroup.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ ERR 0000 message partly improved

Rationale	In case of error due to configuration error, ERR 0000 message that users find difficult to identify occurs.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.27 Version 4.7.1

- **New feature**

■ Support for executeDespiteDataUnavailability option for Rte_Receive API

Rationale	Support for executeDespiteDataUnavailability option for Rte_Receive API
Impact on Behavior	In case of executeDespiteDataUnavailability activation and non-blocking Rte_Receive API, all transformers of Transformer chain are executed, inputBuffer value of all transformers is NULL, and dataLength is 0.
Impact on Settings	None
Required ASW actions	None

- **Improvements**

■ When setting Init Value in Unsigned data type using Text Value Specification, compile error corrected

Rationale	When setting Init Value in Unsigned data type through Text Value Specification, compile error occurs as variable is initialized to a value combined with the Rte.c initial value designated by Text Value Specification and Suffix U.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Corrected a problem of DisabledMode Runnable not operating even when mode is changed

Rationale	If not using Enhanced Mode API or only using OnEntry or OnExit in Mode Switch Event, a variable, which is declared as global variable in a conditional statement for DisabledMode examination in Task but not used, is used. When global variable is initialized to 0 and TargetMode value used in DisabledMode includes 0, the conditional statement is always false and Runnable is not executed.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Application Header and Rte_Hook.h code generation revised to be consistent

Rationale	In case of the same details but a changed order of Arxml, the order of Application Header and Rte_Hook.h file details are also changed.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ During 1:N Sender Receiver communication, warning message displayed if Init Value is not designated in some ComSpec

Rationale	During 1:N Sender Receiver communication, Init Value is designated for most RPorts, but if Init Value is not designated for some RPorts, warning messages are displayed.
Impact on Behavior	None
Impact on Settings	In case of warning, Init Value is designated in ComSpec of the relevant Port.
Required ASW actions	None

■ Rte_Enter, Rte_Exit, SchM_Enter, and SchM_Exit of Rte Generator generation code improved

Rationale	To enhance maintenance by improving Rte_Enter, Rte_Exit, SchM_Enter, and SchM_Exit of Rte Generator generation code
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ During Sender Receiver communication, Validation added for DATA_REFERENCE type

Rationale	To prevent wrong memory reference during Sender Receiver communication using ImplementationDataType of DATA_REFERENCE type
Impact on	None

Behavior	
Impact on Settings	Configuration change if error occurs
Required ASW actions	None

4.3.28 Version 4.8.0

- **Improvements**
  - ■ Corrected a problem of executing SchedulableEntity/Runnable that configured Disabled Mode when it enters or exits the Disabled Mode

Rationale	A logic that examines Disabled Mode before SchedulableEntity/Runnable execution must examine both before and after the mode, but a logic that examines either before or after the mode is created.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Corrected a problem of abnormally calling runnables when using External Triggering Event with the same Queued Trigger

Rationale	If using the same Trigger Interface as PPort in several SWCs and setting runnables of multiple Sink (reception) SWCs to External Trigger Event by using the same Queued Trigger of the relevant Interface, Provided-Required connection of AssemblyConnector is ignored and all SWCs using the relevant Trigger are recognized as Required. When declaring macro and variable and creating task statements related to Queued Trigger, code is written to call all runnables using the relevant Trigger, not just RPort runnables linked to PPort.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Correction and validation processing related to Misra-2012 C Violation

Rationale	Misra C Violation generated.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.29 Version 4.8.1

- **Improvement**
  - During 1:N Mode communication, Rte_Switch/SchM_Switch API that calls Runnable with Direct Function Call revised to call Runnable/ScheduleableEntity in PositionInTask order

Rationale	During 1:N Mode communication, Rte_Switch/SchM_Switch API that calls Runnable with Direct Function Call does not call Runnable/ScheduleableEntity in PositionInTask order but calls randomly.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.30 Version 4.9.0.0

- **New feature**
  - Runnable Entity Trace Events feature added

Rationale	Runnable Entity Trace Events feature unsupported
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

- **Improvements**
  - Corrected a problem of compile error as HASH(0XXXXXXXXX) is displayed in Rte.c / Rte_<Partition>.c file

Rationale	Compile error occurs as HASH(0xXXXXXXXX) is displayed in Rte.c / Rte_<Partition>.c file
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - Manual and PDF configuration items changed

Rationale	Manual, PDF and generator operation inconsistent
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.31 Version 4.10.0.0

- **New feature**
  - -NoRteReceiverPullCB option newly added

Rationale	During Sender Receiver InterPartition communication, frequent data reception degrades performance as Rte_ReceiverPullCB is conducted in Interrupt Context. New option and feature are added to allow data reception without using Rte_ReceiverPullCB in a particular Usecase.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

- **Improvements**
  - ■ Corrected a problem that Extended Task comprised of Timing Event does not execute Runnable/BswScheduleEntity according to PositionInTask settings

Rationale	When generating Extended Task comprised of Timing Event, the execution order of Runnable/SchedulableEntity is not arranged as expected since PositionInTask value is not perceived as numbers but as letters before being arranged (e.g., arranged in an order of '1', '22', '34', and '5', not 1, 5, 22, and 34).
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Corrected a problem that Implicit Sender Receiver communication is not executed because glue code is not created in case of Argument in Server Runnable

Rationale	If Server Runnable operated by Operation Invoked Event has Argument, is task mapped, and configured to Data Read Access or Data Write Access, Implicit Sender Receiver communication is not executed because glue code is not created before and after Server Runnable execution in the Task.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Generator improved for large capacity arxml support

Rationale	Out of Memory occurs when using large capacity arxml. Improvement of memory usage amount by removing <ADMIN-DATA> tag irrelevant to Rte code generation at the parsing stage
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ NmPdu / ISignalToIPduMapping tag recognition support

Rationale	NmPdu/ISignalToIPduMapping tag recognition support for PartialNetwork support
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ CompuMethod Compu Phys to Internal tag recognition support

Rationale	Generator error occurs when CompuMethod Compu Direction is Compu Phys to Internal.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ 1:N Standard Trigger communication restrictions added

Rationale	When setting 1:N Standard Trigger communication operated in Os Task method, triggered runnables may not operate.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.32 Version 4.10.1.0

- **Improvements**

■ Corrected a problem that Implicit Sender Receiver and Implicit Inter Runnable Variable are not executed because glue code is not created normally in case of Return value in Server Runnable

Rationale	Data reception of Implicit Sender Receiver and Implicit Inter Runnable Variable does not work normally as glue code is not created before Server Runnable execution if Server Runnable that is operated by Operation Invoked Event and returns Application Error has configuration related to Implicit Sender Receiver or Implicit Inter Runnable Variable.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Corrected a problem that Application Error is not delivered to Rte_Call and Rte_Result return value if Server Runnable is task mapped and Application Error is returned but Argument is absent during Client Server Intra Partition communication

Rationale	Application Error is not delivered to Client as code for delivering Application Error is not created if Server Runnable is task mapped and Application Error is returned but Argument is absent during Client Server Intra Partition communication.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Corrected a problem that Application Error is not delivered to Rte_Call and Rte_Result return value if Server Runnable returns Application Error during Client Server Inter Partition communication

Rationale	Application Error is not delivered to Client as code for delivering Application Error is not created in Rte_Call, Rte_Result, and Rte_ReceiverPullCB_CsResult API if Server Runnable returns Application Error during Client Server Inter Partition communication.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ N:1 Client Server restrictions added for InterPartition communication

Rationale	N:1 Inter Partition and Intra-Inter Partition Client Server communication unsupported
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Local variable LddMaskType in Extended Task initialized to RTE_ZERO

Rationale	MISRA Violation occurs as local variable LddMaskType is used without initialization
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Error information elaborated in Validation

Rationale	- Input file checking statement in case of error message output - In case of no designated module in BswModuleDependencies - In case of ProvidedEntries referring to the same BswModuleEntry repeatedly
Impact on	None

Behavior	
Impact on Settings	Need to change configuration if error occurs
Required ASW actions	None

4.3.33 Version 4.10.2.0

- **Improvements**
  - ■ Generator error corrected in case of using DATA_REFERENCE in Sender Receiver communication

Rationale	Generator error occurs if ImplementationDataType category is DATA_REFERENCE or includes DATA_REFERENCE and uses it in Sender Receiver communication.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ Corrected to prevent Validation error in case of ComSignalType being UINT8_N, UINT8_DYN and no value in ComBitSize

Rationale	Validation error occurs if ComSignalType is UINT8_N, UINT8_DYN and has not value in ComBitSize.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - ■ MISRA and Run-Time Violation validation and some codes improved

Rationale	MISRA and Run-Time Violation occur.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.34 Version 4.10.3.0

- **Improvements**
  - ■ Corrected a phenomenon where Overlayed Errors are lost in Rte_Read, Rte_IStatus, and Rte_Receive API return value when using Transformer

Rationale	Overlayed Errors are lost as not OR operator but an assignment operator is used for variable to save return value when using

	Transformer.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ ParameterDirection option added to use IN/OUT/INOUT macros

Rationale	Basic operation changed to deactivate IN/OUT/INOUT macros and prevent them from being generated in Rte API/Runnable. Addition of –ParameterDirection option to use the relevant macros.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ -Validate option basic application

Rationale	Correction of generator for -Validate generator option to basically operate
Impact on Behavior	-Validate option basic operation
Impact on Settings	None
Required ASW actions	None

■ Validation added

Rationale	- Inter Ecu Inter Partition communication unsupported - DataReadAccess, DataWriteAccess configuration unsupported for RunnableEntity called by Direct Function Call - Configuration of several RteEvent referring to the same OsEvent in ExtendedTask unsupported
Impact on Behavior	None
Impact on Settings	Configuration change in case of error
Required ASW actions	None

■ Changed for Category of ImplementationDataType designated in impldataTypes_ioc.arxml

Rationale	Category of ImplementationDataType in impldataTypes_ioc.arxml is not designated.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Gc.exe and Rte.bat file in RTe folder removed

Rationale	Gc.exe and Rte.bat that are deployed with Rte Generator to arrange the indentation of files generated by Rte are not used.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.35 Version 4.10.4.0

- **Improvement**

■ 64-bit Rte.exe deployed

Rationale	Out of Memory occurs in case of building a project with large DB.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.36 Version 4.10.5.0

- **Improvements**

■ Added Interrupt Disable / Enable function redefinition macros related to Synchronized Offset

Rationale	Different Interrupt Disable / Enable function names for each compiler
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Suspend/Resume Interrupt code added to prevent Race Condition in Rte_Result API

Rationale	Race Condition may occur in case of global variable access within function
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Validation typo related to Misra C Violation corrected

Rationale	Validation typo related to Misra C Violation
Impact on Behavior	None

Impact on Settings	None
Required ASW actions	None

■ Manual update

Rationale	- Unsupported API return value description removed and return value related to Transformer added - RteExpectedActivationPosition and AlarmSetMethodIsRelative descriptions improved - Descriptions of Synchronized Offset added
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.37 Version 4.10.6.0

- **Improvements**

■ Corrected compile error in case of one Argument and INOUT Direction during InterPartition Client Server communication

Rationale	Compile error occurs as two conditional functions of if statement in Rte_Call API do not have parentheses surrounding them in case of one Argument and INOUT Direction during InterPartition Client Server communication.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Changed to designate a section for NO_INIT area variable using farnoclear keyword when using Tasking compiler

Rationale	NO_INIT area variable is allocated to UNDEFINED_RAMDATA when designating a section with #pragma noclear and farbss.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ AUTOEVER format changed

Rationale	Company name changed
Impact on Behavior	None
Impact on Settings	None

Required ASW actions	None

4.3.38 Version 4.10.7.0

- **Improvements**
  - Buffer deployed to INIT area when using initial value in Implicit Sender Receiver communication

Rationale	Buffer deployed to CLEARED area when using initial value in Implicit Sender Receiver communication
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

  - Security coding improved to comply with the UNECE Cyber Security regulations

Rationale	Violation of the UNECE Cyber Security regulations occurred.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.39 Version 4.11.0.0

- **New feature**
  - Supports REPLACE-BY-TIMEOUT-SUBSTITUTION-VALUE of HandleTimeoutType

Rationale	Supports SUBSTITUTE function of ComRxDataTimeoutAction in Com module
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

- **Improvement**
  - Improve the phenomenon that some Runnables are not called when 1:N Standard Trigger Communication.

Rationale	For Standard Trigger communication, only one Event Flag is generated based on the source, and Sink operating as Async is not called
Impact on Behavior	None
Impact on Settings	None

Required ASW actions	None

■ Improves Mode Switch Init processing and SchM Lifecycle-related compilation errors when only one partition is set.

Rationale	Mode Switch Init and SchM Lifecycle logic processing only when two or more partitions are set.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Improve the problem of some SchM_ActMainFunction APIs not being created when more than three ActivationPoints are set up within BswEntity.

Rationale	Error recognizing number of Activation Points in Generator Code
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

■ Improve error creating Count macro on SchM Trigger Queued 1:N.

Rationale	When creating a Count macro within the generator code, refer to the QueueLength value, not the CountLength value
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.40 Version 4.11.1.0

- **Improvements**

■ Security coding improved to comply with the UNECE Cyber Security regulations

Rationale	Violation of the UNECE Cyber Security regulations occurred.
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.3.41 Version 4.11.1.1

- Improvements
  - UM English version support

Rationale	UM English version support
Impact on Behavior	None
Impact on Settings	None
Required ASW actions	None

4.4 Limitations

4.4.1 General Description

This chapter describes limitations on features and operations provided by RTE module.

4.4.2 Limited Function Description in Rte Module

4.4.2.1 Common Limitation in Overall Rte Module

- **Only Generation Phase is supported in Rte Generation Process.**
- **Minimum Start Interval is not available.**
- **In Instantiation, 'Multiple Instance' is not supported and only single instance is possible.**
- **Periodic RTE Event (i.e. Timing Event / Background Event) can not be mapped into the OsTask which other RTE Events (except for the periodic Rte Event. eg. Operation Invoked Event) mapped also.**
- **RunnableEntity shall be scheduled by only RteEvent. Direct function call from application code is not available.**
- **RTE API shall be called only in the RunnableEntity which has Data/ParameterAccess for DataElement.**

4.4.2.2 Limitation in Rte Interface

4.4.2.2.1 Sender Receiver Interface

- **Compatibility between Implicit Non Queued Sender Receiver and Explicit Non Queued Sender Receiver are not available.**

- **Compatibility between Sender Receiver Interface and Other Interface is not available.**

- **In Implicit Non Queued type Sender Receiver interface, Coherency Group is not available.**

- **In Sender Receiver interface, the following functionalities are not available.**
  Data Conversion and Filter

- **Inter Ecu Communication through Inter Partition is not available.**

- **Rte_IWrite API shall be called every time during an execution of the RunnableEntity if the RunnableEntity has DataWriteAccess.**

- **The following functionalities are not available in case an ImplementationDataType with category DATA_REFERENCE is used in Sender Receiver communication**
  - Inter Partition Communication, Inter ECU Communication, Init Value, Invalidation Policy- Replace and Keep, Handle Out Of Range and Etc.

- **Data Read Access or Data Write Access is not available in a RunnableEntity activated by a Direct Function Call.**

4.4.2.2.2 Client Server Interface

- **Inter-Ecu Communication is not available.**

- **The following functionalities are not available.**

- **N:1(Multiple clients - one server) Inter Partition / Intra-Inter Partition are not supported.**

- In case of inter partition communication, if the partition of server is stopped or restarting during the operation of the server call, the client cannot be notified.

- If queuing feature has been enabled for server operation, basically the queue mechanism in Rte is operated with Task Ready Queue in OsTask. That is, if during server operation, to handle the additional request has been invoked from client, OsTask which the Server Runnable has been mapped into would be activated in Rte_Call API body. And in Os, for additional activation request from Rte, the Queue counter for this OsTask will be increased because this OsTask has been already Ready state (Queue for this OsTask in OS, should be configured). If the queue in Server operation is needed for application design, please contact to Hyundai AUTOEVER.

4.4.2.2.3 Trigger Interface

- The communication based on the Trigger interface between partition with Queue, is not available

- In Trigger interface, synchronized trigger is not available

- If queuing feature has been enabled for Trigger Sink, basically the queue mechanism in Rte is operated with Task Ready Queue in OsTask. That is, if during Trigger Sink operation, to handle the additional request has been invoked from Trigger Source, OsTask which the Sink Runnable has been mapped into would be activated in Rte_Trigger API body. And in Os, for additional activation request from Rte, the Queue counter for this OsTask will be increased because this OsTask has been already Ready state (Queue for this OsTask in OS, should be configured). If the queue in Trigger Sink is needed for application design, please contact to Hyundai AUTOEVER.

- 1:N Trigger Communication is not supported when one of the following cases is satisfied:

  1. The triggered executable entities are activated via Os Task activation and the SwImplPolicy of the Trigger is STANDARD.
  2. The triggered executable entities are activated via Os Task activation and the

SwImplPolicy of the Internal Trigger Point is set to STANDARD

3. The triggered executable entities are activated via Os Task activation and the SwImplPolicy of the BSW Internal Trigger Point is set to STANDARD.

4.4.2.2.4 Mode Switch Interface

- **In Mode Switch Acknowledgement, the following functionalities are not available**
  - Interaction between multi partitions

- **Common Mode Instance**

4.4.2.3 Limitation in Rte Events

4.4.2.3.1 General Limitation in Rte Events

- **The following Rte Events cannot be mapped to Extended Tasks.**
  BackgroundEvent, DataWriteCompletedEvent, OperationInvokedEvent, AsynchronousServerCallReturnsEvent, SwcModeSwitchEvent, ModeSwitchedAckEvent, BswBackgroundEvent, BswModeSwitchEvent, BswModeSwitchedAckEvent

- **Extended Tasks cannot have different kind of Rte Events.**

- **Following RteEvents in an ExtendedTask cannot reference the same OsEvent.**

- **DataReceivedEvent and DataReceivedErrorEvent for implicit Sender-Receiver communication cannot be mapped to Extended Tasks.**

- **ExternalTriggerOccurredEvent, InternalTriggerOccurredEvent, BswExternalTriggerOccurredEvent, and BswInternalTriggerOccurredEvent for QUEUED triggers cannot be mapped to Extended Tasks.**

- **Runnables in Extended Task cannot have DataWriteAccess, or DataReadAccess.**

- Runnables in Extended Task cannot have ReadLocalVariables or WriteLocalVariables for ImplicitInterRunnableVariables.
- When Rte Events are mapped to Extended Tasks, they cannot have Disabled Modes.
- Rte Events cannot have ModeDeclaration in the role disabledMode in the ModeGroup of the unconnected RPortPrototype.
- Rte API with wait point (Blocked Type of Rte API) is not available.

4.4.2.3.2 Timing Event

- Timing Event based on the ScheduleTable, is not supported

4.4.2.3.3 Back Ground Event

- Back Ground Event is not supported.

4.4.2.4 Limitation in Rte APIs

4.4.2.4.1 Rte_Feedback API

- Rte_Feedback with blocked option is not available.

4.4.2.4.2 Rte_SwitchAck API

- Rte_SwitchAck with blocked option is not available.
- Following values are not returned

4.4.2.4.3 Rte_Receive API

- API with Blocking option is not available.

4.4.2.4.4 Rte_Call, Rte_Result API

- APIs for Inter-Ecu Communication is not available.

4.4.2.5 Limitation in Rte LifeCycle and Call-back APIs

4.4.2.5.1 Rte APIs regarding Partition

- Rte_PartitionTerminated, Rte_PartitionRestarting and Rte_RestartPartition APIs are not supported

4.4.2.5.2 Rte APIs regarding NvM Service Callback

- Rte_SetMirror, Rte_GetMirror, Rte_NvMNotifyJobFinished and Rte_NvMNotifyInitBlock APIs are not supported

4.4.2.6 Limitation on Data Transformation

4.4.2.6.1 General Limitation on Data Transformation

- Data Transformation for Client-Server and Trigger Communication is not supported.

- Data Transformation for Intra-ECU Communication is not supported.

4.4.2.6.2 TRANSFORMER-ERROR-HANDLING

- ErrorHandling set to TRANSFORMER-ERROR-HANDLING in PortAPIOption is not supported without using functionality of data transformation.

- Rte_DRead, Rte_Call, Rte_Result and Rte_Trigger don't support the optional OUT parameter transformerError.

4.4.2.7 Limitation in data type

4.4.2.7.1 ApplicationPrimitiveDataType

- ApplicationPrimitiveDataType with category VAL_BLK, STRING and RES_AXIS is not

supported.

- ApplicationPrimitiveDataType with category COM_AXIS, CURVE, MAP cannot be mapped to ImplementationDataType except category ARRAY or STRUCTURE.

- ApplicationPrimitiveDataType with the category COM_AXIS, CURVE and MAP can be only used in parameter interface and sender receiver interface for Intra-ECU communication, PerInstanceParameter, SharedParameter, ArTypedPerInstanceMemory in SWC, PerInstanceParameter in BSW and parameter access within limited function.

- ApplicationPrimitiveDataType with the category COM_AXIS, CURVE and MAP is only initialized with ApplicationValueSpecification of InitValue or ConstantValueSpecification.

- Initialization using ValueGroup in SwValuePhys of SwValueCont or SwAxisCont of ApplicationValueSpecification is not supported.

- Validation or Creation from SwRecordLayout to ImplementationDataType is not supported.

- Functions by configuration in SwDataDefProps about ApplicationPrimitiveDataType with the category COM_AXIS, CURVE, MAP are not supported

4.4.2.7.2 ImplementationDataType

- A category FUNCTION_REFERENCE of ImplementationDataType is not supported.

4.4.2.8 Limitation in Etc. functionality

4.4.2.8.1 Tracing in VFB Trace

- VFB Trace functionality for RTE API, COM and OS is not supported.

- Client Prefix for multiple clients is not supported.

4.4.2.8.2 Monitoring of runnable execution time

- RteVirtuallyMappedToTask is not available

4.4.2.8.3 DET Error reporting

- Development Errors (DET) is not supported.

4.4.2.8.4 Variant Handling

- Variant Handling is not supported

4.4.2.8.5 Calibration

- Double Pointered Method in Calibration, is not available

4.4.2.8.6 Exclusive Area

- Cooperative Runnable Placement can not be configured
- USER_DEFINED_MACRO can not be configured
- When using RenesasOS for RH850, both internal resource and standard resource can not be used in the same task.
- Run Inside Exclusive Area can not be configured

4.5 Deviations

4.5.1 General Description

This chapter describes differences in implementation method and unsupported features compared with AUTOSAR RTE standard specifications (version 4.0.3).

4.5.2 Deviation Function Description in Rte Module

4.5.2.1 Common Deviation in Rte Module

- Not available to generate the Rte API with “indirect type”.
- Functionality regarding minimum Start Interval is not available in Runnable configuration

4.5.2.2 Detail deviation in each Rte functionality

- NvData Interface does not be supported.
- Rte_Ports, Rte_NPorts and Rte_Port APIs are not supported.
- Initialization of Union Ar-Type and C-Typed PerInstanceMemory Type is not supported.
- Array C-Typed PerInstanceMemory is not supported.
- Rte_Invalidate and Rte_IInvalidate is not supported.
- Port Interface Mapping and Compatibility is not supported
- The number of underscore (‘_’) of shortname for BswModuleDescription is limited upto one.
- For data transformation, it is assumed that NeedsOriginalData is false and DespiteDataUnavailability is true.
- Primitive data types is not supported for Data transformation. Therefore Rte_DRead cannot be used for data transformation.
- Verification of data structure between ApplicationPrimitiveDataType with the category COM_AXIS, CURVE and MAP and ImplementationDataType mapped to it in DataTypeMappingSet is not supported.

- When initialization of ApplicationPrimitiveDataType with category COM_AXIS, CURVE and MAP by using ApplicationValueSpecification, Verification of data structure between ImplementationDataType mapped to the ApplicationPrimitiveDataType and the ApplicationValueSpecification is performed. If ApplicationValueSpecification is configured with SwValueCont only, the category of the ImplementationDataType shall be ARRAY. If ApplicationValueSpecification is configured with SwValueCont and SwAxisCont, the category of the ImplementationDataType shall be STRUCTURE and the STRUCTURE shall contain an ARRAY per an axis or a value each.

- When Synchronize Activate Offset is false, Expected Activation Offset is 0 and Alarm Set Methods is Relative is true, the second parameter of SetRelAlarm API is 1 instead of 0 . SetRelAlarm returns an error when the second parameter is 0.

5. Configuration Guide

Rte configuration parameters described in this chapter are parameters that the AUTOSAR standard presents excluding cases with special comments.

5.1 RteGeneration

General elements related to Rte code generation are configured.

Parameter Name	Value	Categor
RteCalibrationSupport[1)]		C
RteDevErrorDetect[2)]	false	F
RteDevErrorDetectUnInit[3)]	false	F
RteVfbTraceClientPrefix[4)]		N
RteVfbTraceEnabled[5)]		C
RteVfbTraceFunction[6)]		C
RteTaskComMapping[7)]		C
RteGenerationMode	COMPATIBILITY_MODE	F
RtelocInteractionReturnValue	RTE_IOC	F
RteMeasurementSupport	false	F
RteOptimizationMode	RUNTIME	F
RteValueRangeCheckEnabled[8)]		C
RteCodeVendorId	-	N
RteToolChainSignificantCharacters	-	N
RteEndToEndProtectionWrapperMode[9)]	-	N

1) RteCalibrationSupport
Set Calibration Method (None/InitializedRam/SinglePointered/OverlayRam).

2) RteDevErrorDetect
DET feature ON/OFF for Rte module

3) RteDevErrorDetectUnInit
Rte Init check feature ON/OFF when using DET feature

4) RteVfbTraceClientPrefix
Prefix input for using additional VfbTraceFunction in Rte API

5) RteVfbTraceEnabled
Vfb Trace feature ON/OFF for Rte module

6) RteVfbTraceFunction
VfbTrace Function input

- RTE API Trace Events : Rte_<API>Hook_<cts>_<ap>_Start (or Return)
- COM: refer to AUTOSAR RTE SWS chapter 5.11.4
- OS Trace Events:

. Rte_Task_<OsService>_TaskType_<OsTask Name> : In case of service with only OsTask as factor
. Rte_Task_<OsService>_TaskType_<OsTask Name>_EventMaskType_<OsEvent Name> :
In case of service with only OsTask as factor
. Rte_Task_<OsService type>_All : Providing Hook function for all OsTask and OsEvent used in RTE
- Runnable Entity Trace Events : Rte_Runnable_[<client>_]<cts>_<reName>_Start (or Return)
※ AUTOEVER internally defined rules as above since implementation was not available with Spec guide.
※ refer to chapter 4.4.2.8.1

7) RteTaskComMapping
As Main Partition with Com Module in Sub Partition in the Multi Partition conditions, this sets OsTask to be used for Communication Data transfer.

8) RteValueRangeCheckEnabled
RangeCheck(HandleOutOfRange) feature ON/OFF for Rte module

9) RteEndToEndProtectionWrapperMode
See E2E User Manual for E2E Protection Wrapper configuration (Configuration unnecessary if not using E2E)

5.2 RteSwComponentType

Information on SwComponentType for Rte code generation is configured.

Parameter Name	Value	Categor
RteComponentTypeRef[1)]		C
RteImplementationRef	-	N

1) RteComponentTypeRef
Select SwComponentType that needs configuration.

5.2.1 RteComponentTypeCalibration

Configure when using Calibration feature.

Parameter Name	Value	Categor
RteCalibrationSupportEnabled[1)]		C
RteCalibrationSwAddrMethodRef	-	N

1) RteCalibrationSupportEnabled
Set to True when the relevant Software Component is ParamterComponentType or Shared/PerInstance Parameter is set to use Calibration feature

5.3 RteSwComponentInstance

Add this container to each SW-C Prototype for SW-C Prototype code generation.

Parameter Name	Value	Categor
RteSoftwareComponentInstanceRef[1)]		C

1) RteSoftwareComponentInstanceRef
Select SW-C Prototype for code generation

5.3.1 RteEventToTaskMapping

Add this container to each RTE Event as OsTask mapping for Event set to execute Runnable in SW-C.

Parameter Name	Value	Categor
RteEventRef [1)]		C
RteMappedToTaskRef[2)]		C
RtePositionInTask[3)]		C
RteUsedOsAlarmRef[4)]		C
RteWaitOsEventRef[6)] (Vendor Specific)		C
RteUsedOsEventRef[5)]		N
RteReceiverUsedOsEventRef (Vendor Specific)		N
RteVirtuallyMappedToTaskRef	-	N
RteActivationOsAlarmRef (Vendor Specific)	-	N
RteActivationOffset	-	N
RteImmediateRestart	-	N
RteOsSchedulePoint	-	N
RteUsedOsSchTblExpiryPointRef	-	N
RteWaitOsAlarmRef (Vendor Specific)		N

1) RteEventRef
Select RTE Event for configuration.

2) RteMappedToTaskRef
Select OsTask to which Runnable that RTE Event executes is set to belong.

3) RtePositionInTask
Decide the order of runnables to set when several runnables belong to one OsTask (Designate numbers and 0 starts first.).
※ Position In Task must not have redundancy in the same OsTask.

4) RteUsedOsAlarmRef
Select OsAlarm to Activate Task chosen at 2) when RTE Event is Timing Event.
(Thus, as for OsAlaram, OsAlarmAction must be selected as ActivateTask, and OsTask

selected for ActivateTask must be the same as OsTask chosen at 2).)
※ Configuration can be ignored if it is not Timing Event.

5) RteUsedOsEventRef
Select OsEvent set by OsAlarm when RTE Event is Timing Event and Task chosen at 2) is Extended Task.
※ Thus, as for OsAlaram chosen at 4), OsAlarmAction must be selected as SetEvent, and OsEvent selected for SetEvent must be the same as OsEvent selected here.
Select OsEvent set (i.e., see OS configuration) to Activate the relevant OsTask when RTE Event is an event besides Timing Event, and Task chosen at 2) is Extended Task.
※ In mobilgene C Studio, the relevant parameter is a configurable item, but RTE Event implementation based on Extended Task is not basically supported. RTE Event implementation based on Extended Task is configurable only after consultation with Hyundai AutoEver. (Refer to Chapter 4.4.2.1)

6) RteWaitOsEventRef
Select OsEvent used to wait for server execution completion at Rte_Call or Rte_Result during Client-Server communication.

5.3.2 RteExclusiveAreaImplementation

Add this container to each Exclusive Area when using Exclusive Area during SW-C Runnable code development.

Parameter Name	Value	Categor
RteExclusiveAreaRef [1)]		C
RteExclusiveAreaImplMechanism [2)]		C
RteExclusiveAreaOsResourceRef[3)]		C

1) RteExclusiveAreaRef
Select a route of Exclusive Area set in Software Component.

2) RteExclusiveAreaImplMechanism
Determine how to operate the exclusive area.
A. ALL_INTERRUPT_BLOCKING: Use SuspendAllInterrupts and ResumeAllInterrupts API of the OS to block all Interrupts in the area.
B. COOPERATIVE_RUNNABLE_PLACEMENT: Unsupported [selection unavailable]
C. OS_INTERRUPT_BLOCKING: Use SuspendOsInterrupts and ResumeOsInterrupts API of the OS to block Category2 Interrupt in the area.
D. OS_RESOURCE: Use GetResource and ReleaseResource API of the OS to enter Critical Section in the area.
※ See the OS manual for more details of SuspendAllInterrupts, ResumeAllInterrupts, SuspendOsInterrupts, ResumeOsInterrupts, and ReleaseResource API.
※ The RenesasOS for RH850 does not support concurrent configuration of Internal Resource and Standard Resource. Therefore, Exclusive Area in OS_RESOURCE mode is not available in a task using Internal Resource. See the RenesasOS manual for more details (Refer to Chapter 4.4.2.6.6).

3) RteExclusiveAreaOsResourceRef
Select OsResource to be used in Exclusive Area when RteExclusiveAreaImplMechanism in 2) is set to OS_RESOURCE.

5.3.3 RteExternalTriggerConfig

Add this container to RteSwComponentIntance related to Trigger Source SW-C if it is a queued mode when using External Trigger Communication between SW-C.

Parameter Name	Value	Categor
RteSwcTriggerSourceRef[1)]		C
RteTriggerSourceQueueLength[2)]		C

1) RteSwcTriggerSourceRef
Select Trigger Instance to be used in a queued mode.

2) RteTriggerSourceQueueLength
Enter the length of Trigger Instance of 1).
※ To use the Queue feature, the queue value of OsTask at which Sink Entity will be mapped must be set (see the OS manual for descriptions of the settings.).
※ As a platform using the Bolero MCU does not support multiple activation of OsTask, Maximum QueueLength equals 1 if a queue feature based on Task Activation (not a direct function call) is used.

5.3.4 RteInternalTriggerConfig

Add this container to RteSwComponentIntance related to TriggerInstance SW-C if it is a queued mode when using Internal Trigger Communication between runnables within SW-C.

Parameter Name	Value	Categor
RteSwcTriggerSourceRef[1)]		C
RteTriggerSourceQueueLength[2)]		C

1) RteSwcTriggerSourceRef
Select Trigger Instance to be used in a queued mode.

2) RteTriggerSourceQueueLength
Enter the length of Trigger Instance of 1).
※ To use the Queue feature, the queue value of OsTask at which Sink Entity will be mapped must be set (see the OS manual for descriptions of the settings.).
※ As a platform using the Bolero MCU does not support multiple activation of OsTask, Maximum QueueLength equals 1 if a queue feature based on Task Activation (not a direct function call) is used.

5.3.5 RteNvRamAllocation

Refer to Chapter 4.5 Deviations

Parameter Name	Value	Categor
RteNvmRamBlockLocationSymbol	-	N
RteNvmRomBlockLocationSymbol	-	N
RteSwNvRamMappingRef	-	N
RteNvmBlockRef	-	N

5.4 RteBswModuleInstance

Add this container to each BSW module during BSW module development.
※ Container that includes configuration information on AUTOSAR BSW module behavior
※ This is what Hyundai AutoEver finalizes and deploys about SWP SRS information and basic item information necessary for each BSW module, and it cannot be changed during Application design and configuration.
※ However, additional settings are allowed in consultation with Hyundai AutoEver when developing CDD Layer directly in the application.

Parameter Name	Value	Categor
RteBswImplementationRef[1)]		C
RteBswModuleConfigurationRef[2)]		C

1) RteBswImplementationRef
Designate Implementation settings route for Bsw module.

2) RteBswModuleConfigurationRef
Designate Configuration settings route for Bsw module.

5.4.1 RteBswEventToTaskMapping

Add this container to each BSW Event as OsTask mapping for Event set to execute Entity in BSW Module Description.

Parameter Name	Value	Categor
RteBswEventRef [1)]		C
RteBswMappedToTask Ref [2)]		C
RteBswPositionInTask [3)]		C
RteBswUsedOsAlarmRef[4)]		C
RteBswUsedOsEventRef[5)]		N
RteBswActivationOffset		N
RteBswImmediateRestart	-	N
RteOsSchedulePoint	-	N
RteBswUsedOsSchTblExpiryPointRef	-	N
RteBswActivationOsAlarmRef	-	N
RteBswTimeOutOsAlarmRef	-	N

1) RteBswEventRef
Designate BSW Event route for Task mapping.

2) RteBswMappedToTaskRef
Designate Task to which Entity that the BSW Event calls will belong.

3) RteBswPositionInTask
Set to decide the order of Entity to be called when Task is executed.
Enter the order of Entity that the BSW Event calls within the mapped task. (Execute from 0.)
※ Position In Task must not have redundancy in the same OsTask.

4) RteBswUsedOsAlarmRef
Designate Alarm to execute Task if BSW Event if Timing Event.
※ Configuration can be ignored if it is not Timing Event.

5) RteBswUsedOsEventRef
Designate Event to execute Task if Task mapped in RTE Event is Extended Task.
※ In mobilgene C Studio, the relevant parameter is a configurable item, but RTE Event implementation based on Extended Task is not basically supported. RTE Event implementation based on Extended Task is configurable only after consultation with Hyundai AutoEver. (Refer to Chapter 4.4.2.1)

5.4.2 RteBswExclusiveAreaImpl

Add this container to each Exclusive Area when using Exclusive Area during BSW Entity code development.

Parameter Name	Value	Categor
RteBswExclusiveAreaRef[1)]		C
RteExclusiveAreaImplMechanism[2)]		C
RteBswExclusiveAreaOsResourceRef[3)]		C
SchMEnableMacroName	-	N
SchMDisableMacroName	-	N

1) RteBswExclusiveAreaRef
Select a route to Exclusive Area set up in the BSW Module Description.

2) RteExclusiveAreaImplMechanism
Determine how to operate the exclusive area.
A. ALL_INTERRUPT_BLOCKING: Use SuspendAllInterrupts and ResumeAllInterrupts API of the OS to block all Interrupts within the zone.
B. COOPERATIVE_RUNNABLE_PLACEMENT: Not supported [not available for selection]
C. OS_INTERRUPT_BLOCKING: Use SuspendOsInterrupts and ResumeOsInterrupts API of the OS to block Category2 Interrupt within the zone.
D. OS_RESOURCE: Use GetResource and ReleaseResource API of the OS to enter Critical Section within the zone.
※ For more details of SuspendAllInterrupts, ResumeAllInterrupts, SuspendOsInterrupts, ResumeOsInterrupts, and ReleaseResource API, see the OS manual.
※ USER_DEFINED_MACRO is not supported.
※ The RenesasOS for RH850 does not support concurrent configuration of Internal

Resource and Standard Resource. Therefore, it is not possible to use Exclusive Area relying on the OS_RESOURCE method within a task that uses Internal Resource. See the RenesasOS manual for more details.(Refer to Chapter 4.4.2.6.6)

3) RteBswExclusiveAreaOsResourceRef
If RteExclusiveAreaImplMechanism in 2) is set to OS_RESOURCE, select the OsResource to be used in Exclusive Area.

5.4.3 RteBswRequiredModeGroupConnection

Add the relevant container when using mode switch communication at the BSW Module Description level.

Parameter Name	Value	Categor
RteBswProvidedModeGroupRef[1)]		C
RteBswRequiredModeGroupRef[2)]		C
RteBswProvidedModeGrpModInstRef[3)]		C

1) RteBswProvidedModeGroupRef
Refer providedModeGroupPrototype. This is connected to the Bsw Module Entity that plays the role of Manager in mode switch communication between BSW.
※ The role of PPort in the mode switch communication between Application SWC

2) RteBswRequiredModeGroupRef
Refer requiredModeGroupPrototype. This is connected to the Bsw Module Entity that plays a user role in mode switch communication between BSW.
※ The role of RPort in the mode switch communication between Application SWC

3) RteBswProvidedModeGrpModInstRef
Select ModeGroupPrototype Instance which will be used by the mode switch connected through the setting in 1) and 2) above.

5.4.4 RteBswRequiredTriggerConnection

When using trigger communication at the BSW Module Description level, add this container to RteBswComponentIntance related to the Release Trigger BSW Module Description.

Parameter Name	Value	Categor
RteBswReleasedTriggerRef[1)]		C
RteBswRequiredTriggerRef[2)]		C
RteBswReleasedTriggerModInstRef[3)]		C

1) RteBswReleasedTriggerRef
Select Released Trigger Instance.

2) RteBswRequiredTriggerRef
Select Required Trigger Instance.

3) RteBswReleasedTriggerModInstRef

Select RteBswComponentIntance that referenced the BSW Module Description with Released Trigger.

5.4.5 RteBswExternalTriggerConfig

When using external trigger communication between the BSW Module Descriptions, add this container to RteBswComponentIntance related to the Release Trigger BSW Module Description if the method is Queued.

Parameter Name	Value	Categor
RteBswTriggerSourceRef [1)]		C
RteBswTriggerSourceQueueLength [2)]		C

1) RteBswTriggerSourceRef
   Select Release Trigger Instance that will be used for the Queued method.
2) RteBswTriggerSourceQueueLength
   Enter the length of Release Trigger instance in 1).
   ※ To use the Queue feature, queue values of OsTask to which Sink Runnable will be mapped should be set up (see the OS manual for information on the setting).
   ※ As a platform that uses the Bolero MCU does not support multiple activation of OsTask, the Maximum QueueLength = 1 if a queue feature that relies on Task Activation (not a direct function call) is used.

5.4.6 RteBswInternalTriggerConfig

When using internal trigger communication between entities in the BSW Module Description, add this container to RteBswComponentIntance related to the BSW Module Description in which Trigger is configured if the method is Queued.

Parameter Name	Value	Categor
RteBswTriggerSourceRef [1)]		C
RteBswTriggerSourceQueueLength [2)]		C

1) RteBswTriggerSourceRef
   Select Release Trigger Instance that will be used for the Queued method.
2) RteBswTriggerSourceQueueLength
   Enter the length of Release Trigger instance in 1).
   ※ To use the Queue feature, queue values of OsTask to which Sink Runnable will be mapped should be set up (see the OS manual for information on the setting).
   ※ As a platform that uses the Bolero MCU does not support multiple activation of OsTask, the Maximum QueueLength = 1 if a queue feature that relies on Task Activation (not a direct function call) is used.

5.5 RteOsInteraction

Configure RteOsInteraction to authorize the Start method and Offset of a specific OsAlarm. To start a specific OsAlarm in AUTOEVER RTE, users can choose between Absolute (the RTE standard feature) and Relative (the Vendor Specific feature) methods. The settings differ by AlarmSetMethodIsRelative (the Vendor Specific Parameter).
※ In general, AUTOSAR RTE standards recommend to use the Absolute method to define operation of the OsAlarm Start through configuration of RteOsInteraction. As for Absolute method, regardless of OsCounter value at the time when current OsAlarm starts, it authorizes the value without additional processing for the Offset. Once OsCounter reaches relevant Offset value, the OsAlarm linked to it starts. On the other hand, in the case of the Relative method additionally provided by AutoEver, the OsAlarm is started after adding the Offset value to the current OsCounter value (that is, after the offset value set at the current time). Additional parameters will be provided for users to distinguish and set this up.
※ As for the OsAlarm whose Alarm Start method is not specifically designated through RteOsInteraction configuration, Offset value is 1 based on RTE and the method of operation is Relative.
In addition, if synchronization of the Offset of Timing Event that operates using a specific OsCounter-based OsAlarm is required, use the following settings (a Vendor Specific feature).
※ Through the setting below, OsTask for Timing Event used in the BSW module and Offset for activation time of the OsTask to which Timing Event used in Application SWC is mapped can be synchronized.
Basically, timing event used in the BSW module is triggered by Alarm setting in SchM_Init (before the run of Rte_Start). The timing event configured in the Application SWC (SWC hereinafter) runs the same way in Rte_Start. At this time, since the offset information for the Timing Event set in BSW and SWC is not synchronized with each other, it is recommended to use it if users want efficient distribution of resources that run periodic timing events based on the synchronized offset. Also, the order of OsAlarm run for individual timing events can be determined by the parameter AutoEver provides (i.e. RteExpectedActivationPosition). (Yet synchronization of BSW and SWC Timing Events can be done only when the same OsCounter is used.)

※ Please make sure users fully understand 8.6 Synchronized Offset (in Chapter 8. Appendix) before using this feature.

Parameter Name	Value	Categor
RteSynchronizedActivateOffset[1)] (Vendor Specific)		C
RteSynchronizedOsCounterRef[3)] (Vendor Specific)		C

1) RteSynchronizedActivateOffset
Whether to use synchronization feature of Offset at the start time of OsAlarm linked to a specific OsCounter

2) RteSynchronizedOsCounterRef
The Base OsCounter that will run Offset synchronization.

5.5.1 RteUsedOsActivation

Designate Offset to run the OsTask mapped to a timing event in 5.3.1 or 5.4.1 at a specific time point. Add this container to every OsTask for which users want to designate Offset.

Parameter Name	Value	Categor
RteActivationOsAlarmRef[1)]		C
RteActivationOsTaskRef[2)]		C
RteExpectedActivationOffset[3)]		C
RteExpectedActivationPosition[4)] (Vendor Specific)		C
AlarmSetMethodIsRelative[5)] (Vendor Specific)		C
RteExpectedTickDuration		N
RteActivationOsSchTblRef	-	N

1) RteActivationOsAlarmRef
Designate the OsAlarm mapped to a timing event

2) RteActivationOsTaskRef
Designate the mapped Os Task for Timing Event

3) RteExpectedActivationOffset
Enter the desired Offset in seconds.

4) RteExpectedActivationPosition
Define the order of OsAlarm called in SchM_Init or Rte_Start.
※ The value should be designated when RteSynchronizedActivateOffset setting is True.

5) AlarmSetMethodIsRelative
The start method of OsAlarm is set to Relative Type. (If there is no setting or if it is false, the OsAlarm will be started as Absolute Type.
※ If the RteSynchronizedActivateOffset setting is true, it always operates as Relative Type.

	RteSynchronizedActivateOffset	RteSynchronizedActivateOffset == True
AlarmSetMethodIsRelative == False	Absolute Type	Relative Type
AlarmSetMethodIsRelative == True	Relative Type	Relative Type

5.5.2 RteModeToScheduleTableMapping

Parameter Name	Value	Categor
RteModeSchtblMapModeDeclarationRef	-	N
RteModeScheduleTableRef	-	N

5.5.2.1 RteModeSchtblMapBsw

Parameter Name	Value	Categor
RteModeSchtblMapBswProvidedModeGroupRef	-	N
RteModeSchtblMapBswInstanceRef	-	N

5.5.2.2 RteModeSchtblMapSwc

Parameter Name	Value	Categor
RteModeSchtblMapSwcPortRef	-	N
RteModeSchtblMapSwcInstanceRef	-	N

5.6 RteBswGeneral

Parameter Name	Value	Categor
RteSchMVersionInfoApi	false	F
RteUseComShadowSignalApi	true	F
SchMUserDefinedfileName	-	N

5.7 CommonPublishedInformation

Parameter Name	Value	Categor
ArReleaseVersion	4.0.3	F
ModuleId	2	F
SwVersion	4.10.7	F
VendorApiInfix	-	N
VendorId	-	N

5.8 RteImplicitCommunication

Parameter Name	Value	Categor
RteCoherentAccess	-	N
RteImmediateBufferUpdate	-	N
RteVariableReadAccessRef	-	N
RteVariableWriteAccessRef	-	N
RteSoftwareComponentInstanceRef	-	N

5.9 RtePostBuildVariantConfiguration

Parameter Name	Value	Categor
RtePostBuildUsedPredefinedVariant	-	N

5.10 RteInitializationBehavior

Parameter Name	Value	Categor
RteSectionInitializationPolicy	INIT	N
RteInitializationStrategy	-	N

6. Application Programming Interface (API)

6.1 Type Definitions

6.1.1 Predefined Error Codes

The following list shows all Predefined error codes that are part of Rte.h and used by the RTE Module

- Std_ReturnType

Type:	uint8		
Range	RTE_E_OK	0	No error occurred.
	RTE_E_INVALID	1	Generic application error indicated by signal invalidation in sender receiver communication with isQueued = false on the receiver side.
	RTE_E_COM_STOPPED	128	An IPDU group was disabled while the application was waiting for the transmission acknowledgment. No value is available. This is not considered a fault, since the IPDU group is switched off on purpose. The semantics are as follows: • The OUT buffers of a client or of explicit read APIs are not modified • No runnable with StartOnEvent on a DataReceivedEvent for this DataElement-Prototype is triggered. • The buffers for implicit read access will keep the previous value.
	RTE_E_TIMEOUT	129	A blocking API call returned due to expiry of a local timeout rather than the intended result. OUT buffers are not modified. The interpretation of this being an error depends on the application.
	RTE_E_LIMIT	130	An internal RTE limit has been exceeded. Request could not be handled. OUT buffers are not modified.
	RTE_E_NO_DATA	131	An explicit read API call returned no data. (This is no error.)
	RTE_E_TRANSMIT_ACK	132	Transmission acknowledgement received.
	RTE_E_LOST_DATA	64	An API call for reading received data of isQueued = true indicates that some incoming data has been lost due to an overflow of the receive queue or due to an error of the underlying communication stack.
	RTE_E_MAX_AGE_EXCEEDED	64	An API call for reading received data of isQueued = false indicates that the available data has exceeded the aliveTimeout limit. A COM signal outdated callback will result in this error.
	RTE_E_IN_EXCLUSIVE_AREA	135	The error is returned by a blocking API and indicates that the runnable could not enter a wait state, because one Executable Entity of the current task's call stack has entered or is running in an ExclusiveArea.

	RTE_E_SEG_FAULT	136	The error can be returned by an RTE API, if the parameters contain a direct or indirect reference to memory that is not accessible from the caller's partition.
	RTE_E_NEVER_RECEIVED	133	No data received for the corresponding unqueued data element since system start or partition restart
	RTE_E_UNCONNECTED	134	The port used for communication is not connected.
	RTE_E_OUT_OF_RANGE	137	This can be returned by Rte_Read API, if the received value is out of bounds.
	RTE_E_HARD_TRANSFORMER_ERROR	138	An error during transformation occurred.
	RTE_E_SOFT_TRANSFORMER_ERROR	140	An error during transformation occurred which shall be notified to the SWC but still produces valid data as output (comparable to a warning).
	SCHM_E_OK	0	No error occurred.
	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. (This is no error.)
	SCHM_E_TRANSMIT_ACK	132	Transmission acknowledgement received.
	SCHM_E_IN_EXCLUSIVE_AREA	135	The error is returned by a blocking API and indicates that the schedulable entity could not enter a wait state, because one ExecutableEntity of the current task's call stack has entered or is running in an ExclusiveArea.
	SCHM_E_TIMEOUT	129	The configured timeout exceeds before the intended result was ready.

6.2 Macro Constants

None

6.3 Functions

In this chapter, in the parameters for each function, <Instance> is a parameter used when a system design based on Multiple Instances is required. As the current RTE Product does not support a feature for this (Support Multiple Instance), this parameter is not used when using each function.

※ refer to 4.4.2.1

6.3.1 RTE APIs

※ When using an API with Std_ReturnType as a return value among RTE APIs, users should add logic to check the return value. If the Return Value is not RTE_E_OK, RTE API request may not be processed. For more details, see the following description of individual API and AUTOSAR documents.

6.3.1.1 Rte_Start

Function Name	Rte_Start	
Syntax:	Std_ReturnType Rte_Start (void)	
Service ID	0x70	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	**RTE_E_OK :** The API call is successfully completed. **RTE_E_LIMIT :** 1. Exceeded the limit of resource usage. 2. Resource allocation failed.
Description	Rte_Start allocates and initializes system resources and communication resources used by the RTE. It is called only once by the ECU state Manager. This service is not invoked from AUTOSAR software components.	
Preconditions	AUTOSAR Com, OS and Memory Services should be initialized.	
Configuration Dependency	The RTE Start API is always created.	

6.3.1.2 Rte_Stop

Function Name	Rte_Stop
Syntax:	Std_ReturnType Rte_Stop (void)
Service ID	0x71
Sync/Async	NA
Reentrancy	NA
Parameters (In)	None
Parameters (Inout)	None

Parameters (Out)	None	
Return Value	Std_ReturnType	**RTE_E_OK :** The API call is successfully completed. **RTE_E_LIMIT :** Resource release failed.
Description	RTE Stop is used to finalize the RTE itself. This service releases all system and communication resources allocated by the Rte. It is called by the ECU state Manager before the basic software modules required by RTE are shut down.	
Preconditions	AUTOSAR Com, OS and Memory Services should be finalized.	
Configuration Dependency	The RTE Stop API is always created.	

6.3.1.3 Rte_Read

Function Name	Rte_Read	
Syntax:	Std_ReturnType Rte_Read_<p>_<o>([IN Rte_Instance <instance>], OUT <data>, [OUT Rte_TransformerError transformerError]) Where <p> is the port name and <o> the VariableDataPrototype within the sender-receiver interface categorizing the port.	
Service ID	0x19	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	None	
Parameters (Out)	<data>	data element to read
Return Value	Std_ReturnType	**RTE_E_OK :** The API call is successfully completed. **RTE_E_INVALID :** An invalid value is received as DataElement. **RTE_E_MAX_AGE_EXCEEDED :** DataElement expired due to timeout (it is an overlayed error and can come together with other error code).

		RTE_E_NEVER_RECEIVED : Never received any DataElement since the system start or restart of a partition **RTE_E_UNCONNECTED :** Receiver port is not connected to Sender Port. **RTE_E_HARD_TRANSFORMER_ERROR:** In the transformer chain, return value of transformer is hard transformer error. **RTE_E_SOFT_TRANSFORMER_ERROR:** In the transformer chain, return value of at least one transformer is soft error and not hard error.
Description	Performs an “explicit” read on a sender-receiver communication data element with “data” semantics (swImplPolicy! = queued). The Rte_Read API is used for explicit read by argument.	
Preconditions	Rte_Start API needs to be called before Rte_Read is called.	
Configuration Dependency	If a VariableAccess in the dataReceivePointByArgument role references a required VariableDataPrototype with ‘data’ semantics.	
Optional Parameter Configuration Dependency	The optional OUT parameter transformerError of the API shall be generated if PortPrototype of port <p> is referenced by a PortAPIOption which has the attribute errorHandling set to transformerErrorHandling.	

6.3.1.4 Rte_DRead

Function Name	Rte_DRead	
Syntax:	<return> Rte_DRead_<p>_<o>([IN Rte_Instance <instance>]) Where <p> is the port name and <o> the VariableDataPrototype within the sender-receiver interface categorizing the port.	
Service ID	0x1A	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	None	
Parameters (Out)	None	

Return Value	Implementation data type	NA
Description	Performs an “explicit” read on a sender-receiver communication data element with “data” semantics (swImplPolicy! = queued). By compatibility, the port may also have a ParameterInterface or an Nv- DataInterface. The Rte_DRead API is used for explicit read by value.	
Preconditions	Rte_Start API needs to be called before Rte_DRead is called.	
Configuration Dependency	A non-blocking Rte_DRead API will be generated if a VariableAccess in the dataReceivePointByValue role references a required VariableDataPrototype with ‘data’ semantics. This requirement is applicable only for primitive data types.	

6.3.1.5 Rte_Write

Function Name	Rte_Write	
Syntax:	Std_ReturnType Rte_Write_<p>_<o>([IN Rte_Instance <instance>], IN <data>, [OUT Rte_TransformerError transformerError]) Where <p> is the port name and <o> the VariableDataPrototype within the sender-receiver interface categorizing the port.	
Service ID	0x14	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
	<data>	data element to write
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	**RTE_E_OK :** The API call is successfully completed. **RTE_E_COM_STOPPED :** Due to unavailability of the COM Service, it is not possible to run Operation. **RTE_E_HARD_TRANSFORMER_ERROR:**

		In the transformer chain, return value of transformer is hard transformer error. **RTE_E_SOFT_TRANSFORMER_ERROR:** In the transformer chain, return value of at least one transformer is soft error and not hard error.
Description	Initiate an “explicit” sender-receiver transmission of data elements with “data” semantic (swImplPolicy different from ’queued’).	
Preconditions	Rte_Start API needs to be called before Rte_Write is called.	
Configuration Dependency	The presence of a VariableAccess in the dataSendPoint role for a provided VariableDataPrototype with data semantics will result in the generation of an Rte_Write API for the provided VariableDataPrototype.	
Optional Parameter Configuration Dependency	The optional OUT parameter transformerError of the API shall be generated if PortPrototype of port <p> is referenced by a PortAPIOption which has the attribute errorHandling set to transformerErrorHandling.	

6.3.1.6 Rte_IsUpdated

Function Name	Rte_IsUpdated	
Syntax	boolean Rte_IsUpdated_<p>_<o> ([IN RTE_Instance <instance>]) Where <p> is the port name and <o> the VariableDataPrototype within the sender-receiver interface categorizing the port.	
Service ID	0x30	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	boolean	**TRUE :** DataElement has been updated since the last reading. **FALSE :** DataElement has not been updated

		since the last reading.
Description	Indicates if the VariableDataPrototype has been updated or not.	
Preconditions	Rte_Start API needs to be called before Rte_IsUpdated is called.	
Configuration Dependency	The presence of a VariableAccess in the dataReceivePointByArgument or dataReceivePointByValue role referring to the VariableDataPrototype and the enableUpdate attribute is enabled in the NonqueuedReceiverComSpec of the VariableDataPrototype.	

6.3.1.7 Rte_IWrite

Function Name	Rte_IWrite	
Syntax:	void Rte_IWrite_<re>_<p>_<o>([IN RTE_Instance <instance>], IN <data>) Where <re> is the runnable entity name, <p> the port name and <o> the VariableDataPrototype name.	
Service ID	0x22	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
	<data>	data to write
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	

Description	Provides write access to the VariableDataPrototypes referenced by VariableAccesses in the dataWriteAccess role.
Preconditions	Rte_Start API needs to be called before Rte_IWrite is called.
Configuration Dependency	An Rte_IWrite API will be created for a provided VariableDataPrototype, if the RunnableEntity has a VariableAccess in the dataWriteAccess role referring to this VariableDataPrototype.

6.3.1.8 Rte_WriteRef

Function Name	Rte_WriteRef	
Syntax:	<return reference> Rte_IWriteRef_<re>_<p>_<o>([IN RTE_Instance <instance>]) Where <re> is the runnable entity name, <p> the port name and <o> the VariableDataPrototype name.	
Service ID	0x23	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Pointer to implementation data type	NA
Description	Provides a reference to the VariableDataPrototype referenced by a VariableAccess in the dataWriteAccess role.	
Preconditions	Rte_Start API needs to be called before Rte_IWriteRef is called.	
Configuration Dependency	An Rte_IWriteRef API will be created for a provided VariableDataPrototype, if the RunnableEntity has a VariableAccess in the dataWriteAccess role referring to this VariableDataPrototype.	

6.3.1.9 Rte_IRead

Function Name	Rte_IRead	
Syntax:	<return> Rte_IRead_<re>_<p>_<o>([IN Rte_Instance <instance>]) Where <re> is the runnable entity name, <p> the port name and <o> the VariableDataPrototype name.	
Service ID	0x21	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Implementation data type	NA
Description	Provides read access to the VariableDataPrototype referenced by VariableAccess in the dataReadAccess role.	
Preconditions	Rte_Start API needs to be called before Rte_IRead is called.	
Configuration Dependency	An Rte_IRead API will be created for a required VariableDataPrototype, if the RunnableEntity has a VariableAccess in the dataReadAccess role referring to this VariableDataPrototype.	

6.3.1.10 Rte_IStatus

Function Name	Rte_IStatus
Syntax:	Std_ReturnType Rte_IStatus_<re>_<p>_<o>([IN Rte_Instance <instance>], [OUT Rte_TransformerError transformerError]) Where <re> is the runnable entity name, <p> the port name and <o> the VariableDataPrototype name.
Service ID	0x25
Sync/Async	NA

Reentrancy	NA	
Parameters (In)	〈instance〉	SW-C instance
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	**RTE_E_OK :** The API call is successfully completed. **RTE_E_INVALID :** An invalid value is received as DataElement. **RTE_E_MAX_AGE_EXCEEDED :** DataElement expired due to Timeout.(Overlayed Error. It can coincide with other Error Code) **RTE_E_NEVER_RECEIVED :** Never received any DataElement since the system start or restart of a partition **RTE_E_UNCONNECTED :** Receiver port is not connected to Sender Port. **RTE_E_HARD_TRANSFORMER_ERROR:** In the transformer chain, return value of transformer is hard transformer error. **RTE_E_SOFT_TRANSFORMER_ERROR:** In the transformer chain, return value of at least one transformer is soft error and not hard error.
Description	Provides the error status of a VariableDataPrototype referenced by a VariableAccess in the dataReadAccess role.	
Preconditions	Rte_Start API needs to be called before Rte_IStatus is called.	
Configuration Dependency	An Rte_IStatus API will be created for a required VariableDataPrototype, if a RunnableEntity has a VariableAccess in the dataReadAccess role referring to this VariableDataPrototype, and if at the RPortPrototype a NonqueuedReceiverComSpec with either1. the attribute AliveTimeout set to a value greater than zero and/or 2. the attribute handleNeverReceived set to TRUE and/or 3. if at the SenderReceiverInterface classifying the RPort-	

	Prototype an InvalidationPolicy set to keep is specified for this VariableDataPrototype.
Optional Parameter Configuration Dependency	The optional OUT parameter transformerError of the API shall be generated if PortPrototype of port <p> is referenced by a PortAPIOption which has the attribute errorHandling set to transformerErrorHandling.

6.3.1.11 Rte_Mode

Function Name	Rte_Mode	
Syntax:	<return> Rte_Mode_<p>_<o>([IN Rte_Instance <instance>]) Where <m> is the ModeDeclarationGroup name, <p> is the port name, and <o> the ModeDeclarationGroupPrototype name within the ModeSwitchInterface categorizing the port.	
Service ID	0x2C	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Rte_ModeType_<m>	NA
Description	Provides the currently active mode of a mode switch port. ※ Design Decision : If the mode user partition is in stopped or restarting, Rte_Mode (in Mode Manager) shall return RTE_TRANSITION_<ModeDeclarationGroup> (in rte_sws_2731) / If the mode manager partition is in stopped or restarting, Rte_Mode (in Mode User) shall return the mode status which has been updated just before entering in stopped or restarting from Mode manager. (AUTOEVER design decision)	
Preconditions	Rte_Start API needs to be called before Rte_Mode is called.	
Configuration Dependency	The existence of a ModeAccessPoint will result in the generation of an Rte_Mode API.	

6.3.1.12 Rte_Switch

Function Name	Rte_Switch	
Syntax:	Std_ReturnType Rte_Switch_<p>_<o>([IN Rte_Instance <instance>], IN Rte_ModeType_<M> <mode>) Where <p> is the port name and <o> the ModeDeclarationGroup-Prototype within the ModeSwitchInterface categorizing the port.	
Service ID	0x15	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
	<mode>	Mode to be switched
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	**RTE_E_OK:** The mode switch request is successfully delivered. **RTE_E_LIMIT:** The mode switch request is dropped as the queue is full.
Description	Initiate a mode switch. The Rte_Switch API call is used for 'explicit' sending of a mode switch notification.	
Preconditions	Rte_Start API needs to be called before Rte_Switch is called.	
Configuration Dependency	The existence of a ModeSwitchPoint will result in the generation of Rte_Switch API.	

6.3.1.13 Rte_Call

Function Name	Rte_Call
Syntax:	Std_ReturnType Rte_Call_<p>_<o>([IN Rte_Instance <instance>], [IN|IN/OUT|OUT] <data_1>... [IN|IN/OUT|OUT] <data_n>)

	〈p〉: Name of R-Port 〈o〉: Name of ClientServerInterface operation		
Service ID	0x1C		
Sync/Async	NA		
Reentrancy	NA		
Parameters (In)	〈instance〉	SW-C instance	
	〈data_1〉 to 〈data_n〉	Data that will be sent from client to server at the request of server (argument whose direction is defined as IN in the operation of ClientServerInterfae)	
Parameters (Inout)	〈data_1〉 to 〈data_n〉	Data that will be sent from the client to the server at the request of the server and simultaneously execute delivery from the server to the client after the service is run (argument whose direction is defined as INOUT in the operation of ClientServerInterfae)	
Parameters (Out)	〈data_1〉 to 〈data_n〉	Data that will be sent from server to client after service in provided in server at the request of server (argument whose direction is defined as OUT in the operation of ClientServerInterfae) (not used in the asynchronous method).	
Return Value	Std_ReturnType[1)]	**RTE_E_OK:** The API call is successfully completed	
		RTE_E_LIMIT: 1. The previous API call is not completed 2. Server queue is full if the connection is N:1	
		RTE_E_TIMEOUT: Failed to receive response from the server within set timeout (return only in synchronous method)	
		RTE_E_UNCONNECTED: Client port is not connected to the server	
		RTE_E_SEG_FAULT: The memory address received as a parameter cannot be used in a partition that called API (return only in the communication between partitions only)	

		〈Application Error〉[2]: An error that took place in the server during service If the server is executed without any errors except for RTE_E_OK above, the server's return value is sent (return only in the synchronous method).
Description	It is a client function that starts the client-server communication. It is used for both synchronous and asynchronous methods. 1. In synchronous mode, it requests the server to execute the service (deliver requested data), waits for the service execution to be completed, and when the service execution is complete, the response data is delivered to the application, terminating the function. 2. If asynchronous, service request is sent to server (delivery of the request data).	
Preconditions	Rte_Start call should be completed before Rte_Call is called	
Configuration Dependency	Rte_Call API will be created when SynchronousServerCallPoint/AsynchrnousServerCallPoint references ClientServerOperation through RPortPrototype.	

1) In the Direct Function Call method, it returns RTE_E_OK, RTE_E_UNCONNECTED and 〈Application Error〉 only.
2) 〈Application Error〉 can be checked through PossibleErrorRef of ClientServerOperation corresponding to the API. For more information on the BSW service error, see the manual for each module.

6.3.1.14 Rte_Result

Function Name	Rte_Result	
Syntax:	Std_ReturnType Rte_Result_〈p〉_〈o〉 ([IN Rte_Instance 〈instance〉], [IN/OUT|OUT 〈param 1〉]... [IN/OUT|OUT 〈param n〉]) 〈p〉: Name of R-Port 〈o〉: Name of ClientServerInterface operation	
Service ID	0x1D	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	〈instance〉	SW-C instance

Parameters (Inout)	<data_1> to <data_n>	Data that will be sent from the client to the server at the request of the server and simultaneously execute delivery from the server to the client after the service is run (Argument whose direction is defined as INOUT in the operation of ClientServerInterfae)
Parameters (Out)	<data_1> to <data_n>	Data that will be sent from server to client after service in provided in server at the request of server (argument whose direction is defined as OUT in the operation of ClientServerInterfae) (not used in the asynchronous method).
Return Value	Std_ReturnType	**RTE_E_OK:** The API call is successfully completed
		RTE_E_NO_DATA: Rte_Call is not successful or the server run is not completed
		RTE_E_TIMEOUT: No response from server within the set timeout
		RTE_E_UNCONNECTED: Client port is not connected to the server
		RTE_E_SEG_FAULT: The memory address received as a parameter cannot be used in a partition that called API (return only in the communication between partitions only)
		<Application Error>: An error that took place in the server during service Deliver return value in the server if it ran without any error other than RTE_E_OK.
Description	In asynchronous client-server communication, get the server's service run result.	
Preconditions	Call of Rte_Start should be completed before Rte_Result is called.	
Configuration Dependency	If AsynchronousServerCallResultPoint references AsynchrnousServerCallPoint, and AsynchrnousServerCallReturnsEvent does not have WaitPoint set up,	

	Non-blocking Rte_Result API will be generated. If AsynchronousServerCallResultPoint references AsynchrnousServerCallPoint, and AsynchrnousServerCallReturnsEvent has WaitPoint set up, Blocking Rte_Result API will be generated. If AsynchronousServerCallReturnsEvent references RunnableEntity and ClientServerOperation RunnableEntity is activated when delivery of service is completed in the server. AsynchronousServerCallReturnsEvent references RunnableEntity and it cannot be referenced by WaitPoint at the same time.

6.3.1.15 Rte_Invalidate

Function Name	Rte_Invalidate	
Syntax:	Std_ReturnType Rte_Invalidate_<p>_<o>([IN Rte_Instance <instance>], [OUT Rte_TransformerError transformerError]) Where <p> is the port name and <o> the VariableDataPrototype within the sender-receiver interface categorizing the port.	
Service ID	0x16	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Std_ReturnType	**RTE_E_OK :** The API call is successfully completed. **RTE_E_COM_STOPPED :** Due to unavailability of the COM Service, it is not possible to run Operation. **RTE_E_HARD_TRANSFORMER_ERROR:** In the transformer chain, return value of transformer is hard transformer

		error. **RTE_E_SOFT_TRANSFORMER_ERROR:** In the transformer chain, return value of at least one transformer is soft error and not hard error.
Description	Invalidate a data element for an “explicit” sender-receiver transmission.	
Preconditions	Rte_Start API needs to be called before Rte_Invalidate is called.	
Configuration Dependency	An Rte_Invalidate API will be created for any VariableAccess in the dataSendPoint role that references a provided VariableDataPrototype which associated InvalidationPolicy is set to keep or replace.	
Optional Parameter Configuration Dependency	The optional OUT parameter transformerError of the API shall be generated if PortPrototype of port <p> is referenced by a PortAPIOption which has the attribute errorHandling set to transformerErrorHandling.	

6.3.1.16 Rte_IInvalidate

Function Name	Rte_IInvalidate	
Syntax:	void Rte_IInvalidate_<re>_<p>_<o>([IN Rte_Instance <instance>]) Where <re> is the runnable entity name, <p> the port name and <o> the VariableDataPrototype name.	
Service ID	0x24	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	None	

Description	Invalidate a VariableDataPrototype referenced by a VariableAccess in the dataWriteAccess role.
Preconditions	Rte_Start API needs to be called before Rte_IInvalidate is called.
Configuration Dependency	An Rte_IInvalidate API will be created for a provided VariableDataPrototype, if the RunnableEntity has VariableAccesses in the dataWriteAccess role referring to this VariableDataPrototype and the associated Invalidation-Policy of the VariableDataPrototype is set to keep or replace.

6.3.1.17 Rte_Feedback

Function Name	Rte_Feedback	
Syntax:	Std_ReturnType Rte_Feedback_<p>_<o>([IN Rte_Instance <instance>]) Where <p> is the port name and <o> the VariableDataPrototype within the sender-receiver interface categorizing the port.	
Service ID	0x17	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Std_ReturnType	**RTE_E_NO_DATA :** The result of data transfer request is not received yet. **RTE_E_COM_STOPPED :** Due to unavailability of the COM Service, it is not possible to run Operation. **RTE_E_TIMEOUT :** (Inter-ECU and Inter-Partition only) Timeout took place. **RTE_E_TRANSMIT_ACK :** Data transfer is requested (including the case of queue overflow). **RTE_E_UNCONNECTED :** Sender Port is not connected to

		Receiver Port. **RTE_E_HARD_TRANSFORMER_ERROR:** In the transformer chain, return value of transformer is hard transformer error. **RTE_E_SOFT_TRANSFORMER_ERROR:** In the transformer chain, return value of at least one transformer is soft error and not hard error.
Description	Provide access to acknowledgement notifications for explicit sender receiver communication and to pass error notification to senders.	
Preconditions	Rte_Start API needs to be called before Rte_Feedback is called.	
Configuration Dependency	A blocking Rte_Feedback API will be generated for a provided VariableDataPrototype, if acknowledgement is enabled and a WaitPoint references a DataSendCompletedEvent that in turn references the VariableDataPrototype. A non-blocking Rte_Feedback API will be generated for a provided VariableDataPrototype, if acknowledgement is enabled and a VariableAccess in the dataSendPoint role references the VariableDataPrototype but no WaitPoint references the DataSendCompletedEvent that references the VariableDataPrototype.	

6.3.1.18 Rte_IFeedback

Function Name	Rte_IFeedback	
Syntax:	Std_ReturnType Rte_IFeedback_<re>_<p>_<o> ([IN RTE_Instance <instance>]) Where <re> is the runnable entity name, <p> the port name and <o> the VariableDataPrototype within the sender-receiver interface categorizing the port.	
Service ID	0x2F	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	

Return Value	Std_ReturnType	**RTE_E_NO_DATA :** The result of data transfer request is not received yet. **RTE_E_COM_STOPPED :** Due to unavailability of the COM Service, it is not possible to run Operation. **RTE_E_TIMEOUT :** (Inter-ECU and Inter-Partition only) Timeout took place. **RTE_E_TRANSMIT_ACK :** Data transfer is requested **RTE_E_UNCONNECTED :** Sender Port is not connected to Receiver Port. **RTE_E_HARD_TRANSFORMER_ERROR:** In the transformer chain, return value of transformer is hard transformer error. **RTE_E_SOFT_TRANSFORMER_ERROR:** In the transformer chain, return value of at least one transformer is soft error and not hard error.
Description	Provide access to acknowledgement notifications for implicit sender receiver communication and to pass error notification to senders.	
Preconditions	Rte_Start API needs to be called before Rte_IFeedback is called.	
Configuration Dependency	An Rte_IFeedback API will be created for a provided VariableDataPrototype, if acknowledgment is enabled and the RunnableEntity has a VariableAccess in the dataWriteAccess role referring to this VariableDataPrototype. An Rte_IFeedback API will be created for a provided VariableDataPrototype, if acknowledgment is enabled and a DataWriteCompletedEvent references the RunnableEntity as well as the VariableDataPrototype.	

6.3.1.19 Rte_Enter

Function Name	Rte_Enter
Syntax:	void Rte_Enter_<name>([IN Rte_Instance <instance>])

	Where <name> is the exclusive area name.	
Service ID	0x2A	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	None	
Description	Rte_Enter API call is invoked by an AUTOSAR software component to define the start of an exclusive area.	
Preconditions	Rte_Start API needs to be called before Rte_Enter is called.	
Configuration Dependency	An Rte_Enter API will be created for each ExclusiveArea that is declared and which has a canEnterExclusiveArea association.	

6.3.1.20 Rte_Exit

Function Name	Rte_Exit	
Syntax:	void Rte_Exit_<name>([IN Rte_Instance <instance>]) Where <name> is the exclusive area name.	
Service ID	0x2B	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	

Return Value	None
Description	Rte_Exit API call is invoked by an AUTOSAR software component to define the end of an exclusive area.
Preconditions	Rte_Start API needs to be called before Rte_Exit is called.
Configuration Dependency	An Rte_Exit API will be created for each ExclusiveArea that is declared and which has a canEnterExclusiveArea association.

6.3.1.21 Rte_IrvRead

Function Name	Rte_IrvRead	
Syntax:	Primitive Type: <return> Rte_IrvRead_<re>_<o>([IN RTE_Instance <instance>]) Complex type signature: void Rte_IrvRead_<re>_<o>([IN RTE_Instance <instance>], OUT <data>) Where <re> is the name of the runnable entity the API might be used in, <o> is the name of the InterRunnableVariables. The complex type signature is used, if the Implementation-Data Type of the InterRunnableVariable resolves to Array Implementation Data Type or Structure Implementation Data Type, otherwise the primitive type signature is used.	
Service ID	0x28	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	Complex Type: <data>	data to be read
Return Value	Primitive Type: Implementation Data type Complex Type: NA	NA

Description	Provides read access to the defined InterRunnableVariables with explicit behavior within a component Description.
Preconditions	Rte_Start API needs to be called before Rte_IrvRead is called.
Configuration Dependency	An Rte_IrvRead API shall be created for each read InterRunnableVariable using explicit access.

6.3.1.22 Rte_IrvWrite

Function Name	Rte_IrvWrite	
Syntax:	void Rte_IrvWrite_<re>_<o>([IN RTE_Instance <instance>], IN <data>) Where <re> is the name of the runnable entity the API might be used in, <o> is the name of the InterRunnableVariable to access and <data> is the placeholder for the data the InterRunnableVariable shall be set to.	
Service ID	0x29	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
	<data>	data to be written
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	None	
Description	Provides write access to the InterRunnableVariables with explicit behavior within a component description.	
Preconditions	Rte_Start API needs to be called before Rte_IrvWrite is called.	

Configuration Dependency	An Rte_IrvWrite API shall be created for each written InterRunnableVariable using explicit access.

6.3.1.23 Rte_IrvIRead

Function Name	Rte_IrvIRead	
Syntax:	<return> Rte_IrvIRead_<re>_<o>([IN RTE_Instance <instance>]) Where <re> is the name of the runnable entity the API might be used in, <o> is the name of the VariableDataPrototype in role implicitInterRunnableVariable.	
Service ID	0x26	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Implementation data type	NA
Description	Provide read access to the InterRunnableVariables with implicit behavior of an AUTOSAR SW-C.	
Preconditions	Rte_Start API needs to be called before Rte_IrvIRead is called.	
Configuration Dependency	An Rte_IrvIRead API shall be created for each VariableAccess in role readLocalVariable to an implicit-InterRunnableVariable.	

6.3.1.24 Rte_IrvIWrite

Function Name	Rte_IrvIWrite

Syntax:	void Rte_IrvIWrite_<re>_<o>([IN RTE_Instance <instance>],IN <data>) Where <re> the name of the RunnableEntity is API might be used in, <o> is the name of the VariableDataPrototype in the role implicitInterRunnableVariable to access and <data> is the placeholder for the data the InterRunnableVariable shall be set to.	
Service ID	0x27	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
	<data>	data to be written
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	None	
Description	Provide write access to the InterRunnableVariables with implicit behavior of an AUTOSAR SW-C.	
Preconditions	Rte_Start API needs to be called before Rte_IrvIWrite is called.	
Configuration Dependency	An Rte_IrvIWrite API shall be created for each VariableAccess in role writtenLocalVariable to an implicitInterRunnableVariable.	

6.3.1.25 Rte_Trigger

Function Name	Rte_Trigger
Syntax:	without queuing support: void Rte_Trigger_<p>_<o>([IN Rte_Instance <instance>]) with queuing support: Std_ReturnType Rte_Trigger_<p>_<o>([IN Rte_Instance <instance>])

	<p>: Name of P-Port <o>: Name of Trigger in Trigger Interface	
Service ID	0x2D	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Std_ReturnType	RTE_E_OK : The API call is successfully completed
		RTE_E_LIMIT : Queue is full in the Queued method
Description	A function that calls the Runnable(s) of external SW-C.	
Preconditions	Call of Rte_Start should be completed before Rte_Trigger is called.	
Configuration Dependency	ExternalTriggeringPoint references Trigger through the PPortPrototype. WhenExternalTriggerOccurredEvent references the same trigger, Rte_Trigger API will be created.	

6.3.1.26 Rte_IrTrigger

Function Name	Rte_IrTrigger
Syntax:	signatwithout queuing support: void Rte_IrTrigger_<re>_<o>([IN Rte_Instance <instance>]) signature with queuing support: Std_ReturnType Rte_IrTrigger_<re>_<o>([IN Rte_Instance <instance>]) <re>: Name of the RunnableEntity that uses API <o>: Name of the InternalTriggeringPoint set in <re>
Service ID	0x2E
Sync/Async	NA
Reentrancy	NA

Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Std_ReturnType	RTE_E_OK : The API call is successfully completed
		RTE_E_LIMIT : Queue is full in the Queued method
Description	A function that calls other RunnableEntity(s) inside of the SW-C.	
Preconditions	Call of Rte_Start should be completed before Rte_Trigger is called.	
Configuration Dependency	Rte_IrTrigger API is created when the InternalTriggeringPoint set in the calling RunnableEntity references the InternalTrigger-OccurredEvent of the Runnable to be called.	

6.3.1.27 Rte_Receive

Function Name	Rte_Receive	
Syntax:	Std_ReturnType Rte_Receive_<p>_<o>([IN Rte_Instance <instance>], OUT <data>, [OUT uint16 <length>], [OUT Rte_TransformerError transformerError]) Where <p> is the port name and <o> the data element within the sender-receiver interface categorizing the port	
Service ID	0x1B	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	data	data element to read
	length	number of elements in the data element
Return Value	Std_ReturnType	**RTE_E_OK :** The API call is successfully completed. **RTE_E_NO_DATA :** The result of data transfer request is

		not received yet. **RTE_E_LOST_DATA:** Some of the data is lost due to queue overflow or communication error. (it is an overlayed error and can come together with other error code). **RTE_E_UNCONNECTED :** Receiver port is not connected to Sender Port. **RTE_E_HARD_TRANSFORMER_ERROR:** In the transformer chain, return value of transformer is hard transformer error. **RTE_E_SOFT_TRANSFORMER_ERROR:** In the transformer chain, return value of at least one transformer is soft error and not hard error.
Description	Initiate an "explicit" sender-receiver transmission of data elements with "data" semantic (swImplPolicy equal to 'queued').	
Preconditions	Rte_Start API needs to be called before Rte_Receive is called.	
Configuration Dependency	Rte_Receive API shall be generated if a VariableAccess in the dataReceivePointByArgument role references a required VariableDataPrototype with 'event' semantics	
Optional Parameter Configuration Dependency	The optional OUT parameter transformerError of the API shall be generated if PortPrototype of port <p> is referenced by a PortAPIOption which has the attribute errorHandling set to transformerErrorHandling.	

6.3.1.28 Rte_SwitchAck

Function Name	Rte_SwitchAck	
Syntax:	Std_ReturnType Rte_SwitchAck_<p>_<o>([IN Rte_Instance <instance>]) Where <p> is the port name and <o> the ModeDeclarationGroupPrototype within the ModeSwitchInterface categorizing the port	
Service ID	0x18	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance

Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Std_ReturnType	**RTE_E_NO_DATA:** The mode change is ongoing. **RTE_E_TRANSMIT_ACK:** The mode change is completed. **RTE_E_UNCONNECTED:** Manager Port is not connected to User Port.
Description	The Rte_SwitchAck API takes no parameters other than the instance handle – the return value is used to indicate the acknowledgement status to the caller	
Preconditions	Rte_Start API needs to be called before Rte_SwitchAck is called.	
Configuration Dependency	A blocking Rte_SwitchAck API shall be generated for a provided ModeDeclarationGroupPrototype if acknowledgementis enabled and a WaitPoint references a ModeSwitchedAckEvent that in turn references the ModeDeclarationGroupPrototype	

6.3.1.29 Rte_Prm

Function Name	Rte_Prm	
Syntax:	〈return〉 Rte_Prm_〈p〉_〈o〉([IN Rte_Instance 〈instance〉]) Where 〈p〉 is the port name and 〈o〉 is the name of the ParameterDataPrototype within the ParameterInterface categorizing the port	
Service ID	0x20	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	〈instance〉	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	

Return Value	<return>	NA
Description	The Rte_Prm API provides access to the defined parameter within a ParameterSwComponentType	
Preconditions	Rte_Start API needs to be called before Rte_Prm is called.	
Configuration Dependency	A Rte_Prm API shall be generated if a ParameterAccess references a ParameterDataPrototype in a require PortPrototype	

6.3.1.30 Rte_CData

Function Name	Rte_CData	
Syntax:	<return> Rte_CData_<name>([IN Rte_Instance <instance>]) Where <name> is the calibration parameter name	
Service ID	0x1F	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	<return>	NA
Description	The Rte_CData API provides access to the defined calibration parameter Within a software-component. The actual data values for a software-component instance may be set after component compilation.	
Preconditions	Rte_Start API needs to be called before Rte_CData is called.	
Configuration Dependency	An Rte_CData API shall be created for each defined ParameterDataPrototype in the role perInstanceParameter or sharedParameter within an AUTOSAR softwarecomponent.	

6.3.1.31 Rte_Send

Function Name	Rte_Send	
Syntax:	Std_ReturnType Rte_Send_<p>_<o>([IN Rte_Instance <instance>], IN <data>, [IN uint16 <length>], [OUT Rte_TransformerError transformerError]) Where <p> is the port name and <o> the VariableDataPrototype within the sender-receiver interface categorizing the port	
Service ID	0x13	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
	<data>	data element to sent
	<length>	number of elements in the data element
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Std_ReturnType	**RTE_E_OK :** The API call is successfully completed. **RTE_E_COM_STOPPED :** Due to unavailability of the COM Service, it is not possible to run Operation. **RTE_E_LIMIT :** (Intra Ecu Communication Only) Due to a full queue, the DataElement cannot be transferred. **RTE_E_HARD_TRANSFORMER_ERROR:** In the transformer chain, return value of transformer is hard transformer error. **RTE_E_SOFT_TRANSFORMER_ERROR:** In the transformer chain, return value of at least one transformer is soft error and not hard error.

Description	Initiates a sender-receiver communication where the transmission occurs at the point the API call is made transmission
Preconditions	Rte_Start API needs to be called before Rte_Send is called.
Configuration Dependency	The Rte_Send APIs may only be used by the runnable that contains the corresponding VariableAccess in the dataSendPoint role
Optional Parameter Configuration Dependency	The optional OUT parameter transformerError of the API shall be generated if PortPrototype of port <p> is referenced by a PortAPIOption which has the attribute errorHandling set to transformerErrorHandling.

6.3.1.32 Enhanced Rte_Mode

Function Name	Enhanced Rte_Mode	
Syntax:	<return> Rte_Mode_<p>_<o>([IN Rte_Instance <instance>,] OUT <previousmode>, OUT <nextmode>) Where <p> is the port name, and <o> the ModeDeclarationGroup- Prototype name within the ModeSwitchInterface categorizing the Port.	
Service ID	0x2C	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	<previousmode>	
	<nextmode>	
Return Value	RTE_TRANSITION_ <ModeDeclarationGroup>	NA

Description	Provides the currently active mode of a mode switch port and also the mode currently being left and the mode being entered.
Preconditions	Rte_Start API needs to be called before Enhanced Rte_Mode is called.
Configuration Dependency	The existence of a ModeAccessPoint given that the attribute enhancedModeApi of the ModeSwitchReceiverComSpec is set to true shall result in the generation of Enhanced Rte_Mode API.

6.3.2 RTE Call-backs

6.3.2.1 Rte_COMCbkTAck_<Signal>

Function Name	Rte_COMCbk_<Signal>
Syntax:	void Rte_COMCbk_<sn>(void) Where <sn> is a COM signal name.
Service ID	0x95
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that the signal of the primitive data item/event is ready for reception.
Preconditions	NA
Configuration Dependency	Configured in COM: ComNotification as part of ComSignal.

6.3.2.2 Rte_COMCbkTAck_<Signal>

Function Name	Rte_COMCbkTAck_<Signal>
Syntax:	void Rte_COMCbkTAck_<sn>(void) Where <sn> is a COM signal name and "TAck" is literal text indicating transmission acknowledgment.
Service ID	0x90
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that the signal of the primitive data item/event is already handed over by COM to the PDU router.
Preconditions	NA
Configuration Dependency	Configured in COM: ComNotification as part of ComSignal

6.3.2.3 Rte_COMCbkTErr_<Signal>

Function Name	Rte_COMCbkTErr_<Signal>
Syntax:	void Rte_COMCbkTErr_<sn>(void) Where <sn> is a COM signal name and "TErr" is literal text indicating transmission error.
Service ID	0x91
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA

Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that an error occurred when the signal of the primitive data item/event was handed over by COM to the PDU router.
Preconditions	NA
Configuration Dependency	Configured in COM: Com-ErrorNotification as part of ComSignal.

6.3.2.4 Rte_COMCbkInv_<Signal>

Function Name	Rte_COMCbkTErr_<Signal>
Syntax:	void Rte_COMCbkInv_<sn>(void) Where <sn> is a COM signal name and "Inv" is literal text indicating signal invalidation.
Service ID	0x92
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that COM has received a signal and passed it as "invalid".
Preconditions	NA

Configuration Dependency	Configured in Com: Com-InvalidNotification as part of ComSignal.

6.3.2.5 Rte_COMCbkRxTOut_<Signal>

Function Name	Rte_COMCbkRxTOut_<Signal>
Syntax:	void Rte_COMCbkRxTOut_<sn>(void) Where <sn> is a COM signal name and "RxTOut" is literal text indicating reception signal time out.
Service ID	0x93
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that the aliveTimeout after the last successful reception of the signal of the primitive data item/event has expired (data element outdated).
Preconditions	NA
Configuration Dependency	Configured in Com: Com-TimeoutNotification as part of ComSignal.

6.3.2.6 Rte_COMCbkTxTOut_<Signal>

Function Name	Rte_COMCbkTxTOut_<Signal>
Syntax:	void Rte_COMCbkTxTOut_<sn>(void) Where <sn> is a COM signal name and "TxTOut" is literal text indicating transmission failure and time out.

Service ID	0x94
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that the timeout of TransmissionAcknowledgementRequest for sending the signal of the primitive data item/event has expired.
Preconditions	NA
Configuration Dependency	Configured in Com: Com-TimeoutNotification as part of ComSignal.

6.3.2.7 Rte_COMCbk_<SignalGroup>

Function Name	Rte_COMCbk_<SignalGroup>
Syntax:	void Rte_COMCbk_<sg>(void) Where <sg> is the name of the COM signal group, which contains all the signals of the composite data item/event or an operation.
Service ID	0x94
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA

Return Value	None
Description	This callback function indicates that the signals of the composite data item/event or the arguments of an operation are ready for reception.
Preconditions	NA
Configuration Dependency	Configured in Com: ComNotification as part of ComSignalGroup.

6.3.2.8 Rte_COMCbkTAck_<SignalGroup>

Function Name	Rte_COMCbkTAck_<SignalGroup>
Syntax:	void Rte_COMCbkTAck_<sg>(void) Where <sg> is COM signal group name and “TAck” is literal text indicating transmission acknowledgment.
Service ID	0x96
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that the signal of the composite data item/event is already handed over by COM to the PDU router.
Preconditions	NA

Configuration Dependency	Configured in Com: ComNotification as part of ComSignalGroup.

6.3.2.9 Rte_COMCbkTErr_<SignalGroup>

Function Name	Rte_COMCbkTErr_<SignalGroup>
Syntax:	void Rte_COMCbkTErr_<sg>(void) Where <sg> is COM signal group name and "TErr" is literal text indicating transmission error.
Service ID	0x97
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that an error occurred when the signal of the composite data item/event was handed over by COM to the PDU router.
Preconditions	NA
Configuration Dependency	Configured in Com: Com-ErrorNotification as part of ComSignalGroup.

6.3.2.10 Rte_COMCbkInv_<SignalGroup>

Function Name	Rte_COMCbkInv_<SignalGroup>
Syntax:	void Rte_COMCbkInv_<sg>(void) Where <sg> is COM signal group name and "Inv" is literal text indicating signal group invalidation.

Service ID	0x98
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that COM has received a signal group and parsed it as “invalid”.
Preconditions	NA
Configuration Dependency	Configured in Com: Com-InvalidNotification as part of ComSignalGroup.

6.3.2.11 Rte_COMCbkRxTOut_〈SignalGroup〉

Function Name	Rte_COMCbkRxTOut_〈SignalGroup〉
Syntax:	void Rte_COMCbkRxTOut_〈sg〉(void) Where 〈sg〉 is COM signal group name and “RxTOut” is literal text indicating reception signal time out.
Service ID	0x99
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None

Description	This callback function indicates that the aliveTimeout after the last successful reception of the signal group carrying the composite data item has expired (data element outdated).
Preconditions	NA
Configuration Dependency	Configured in Com: Com-TimeoutNotification as part of ComSignalGroup.

6.3.2.12 Rte_COMCbkTxTOut_<SignalGroup>

Function Name	Rte_COMCbkTxTOut_<SignalGroup>
Syntax:	void Rte_COMCbkTxTOut_<sg>(void) Where <sg> is COM signal group name and "TxTOut" is literal text indicating transmission failure and timeout.
Service ID	0x9A
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	NA
Parameters (Out)	NA
Return Value	None
Description	This callback function indicates that the timeout of TransmissionAcknowledgementRequest for sending the signal group of the composite data item/event has expired.
Preconditions	NA
Configuration Dependency	Configured in Com: Com-TimeoutNotification as part of ComSignalGroup.

6.3.3 BSW Scheduler APIs

6.3.3.1 SchM_Init

Function Name	SchM_Init
Syntax:	void SchM_Init(SchM_ConfigType * ConfigPtr) Where ConfigPtr is the Pointer to selected Post build Configuration.
Service ID	0x00
Sync/Async	NA
Reentrancy	NA
Parameters (In)	<ConfigPtr>
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
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. After initialization the scheduling of BswSchedulableEntitys is enabled.
Preconditions	AUTOSAR Com, OS and Memory Services should be finalized.
Configuration Dependency	Variant Handling: If post build data set configuration is present, then SchM_Init will have the input parameter 'ConfigPtr' else void will be the input.

6.3.3.2 SchM_Deinit

Function Name	SchM_Deinit
Syntax:	void SchM_Deinit(void)
Service ID	0x01
Sync/Async	NA
Reentrancy	NA
Parameters (In)	None

Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
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.
Preconditions	None
Configuration Dependency	API may only be used after the RTE finalized i.e., after termination of the Rte_Stop.

6.3.3.3 SchM_Enter

Function Name	SchM_Enter
Syntax:	void SchM_Enter_<bsnp>[_<vi>_<ai>]_<name>() Where <bsnp> is the BSW Scheduler Name Prefix of the calling BSW module derived from the BswModuleDescriptions short name, <vi> is the vendorId of the calling BSW module, <ai> vendorApiInfix of the calling BSW module and <name> name is the exclusive area name. The sub part in squared brackets [_<vi>_<ai>] is omitted if no vendorApiInfix is defined for the Basic Software Module.
Service ID	0x03
Sync/Async	NA
Reentrancy	NA
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None

Description	The SchM_Enter API call is invoked by an AUTOSAR BSW module to define the start of an exclusive area.
Preconditions	SchM_Init API needs to be called before SchM_Enter is called.
Configuration Dependency	This API is created for each ExclusiveArea that is declared in the BswBehavior and which has a CanEnterExclusiveArea association. Variant Handling: if variation point is configured, API will be generated depending on the variation point condition by formula. Note: For post build configuration API may be generated, which is inactive due to the evaluated condition by formula resulting to FALSE.

6.3.3.4 SchM_Exit

Function Name	SchM_Exit
Syntax:	Void SchM_Exit_<bsnp>[_<vi>_<ai>]_<name>() Where <bsnp> is the BSW Scheduler Name Prefix of the calling BSW module derived from the BswModuleDescriptions short name, <vi> is the vendorId of the calling BSW module, <ai> vendorApiInfix of the calling BSW module and <name> name is the exclusive area name. The sub part in squared brackets [_<vi>_<ai>] is omitted if no vendorApiInfix is defined for the Basic Software Module.
Service ID	0x04
Sync/Async	NA
Reentrancy	NA
Parameters (In)	NA
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None

Description	The SchM_Exit API call is invoked by an AUTOSAR BSW module to define the end of an exclusive area.
Preconditions	Schm_Init and SchM_Enter API needs to be called before SchM_Exit is called.
Configuration Dependency	This API is created for each ExclusiveArea that is declared in the BswBehavior and which has a CanEnterExclusiveArea association. Variant Handling: if variation point is configured, API will be generated depending on the variation point condition by formula. Note: For post build configuration API may be generated, which is inactive due to the evaluated condition by formula resulting to FALSE.

6.3.3.5 SchM_Mode

Function Name	SchM_Mode	
Syntax:	SchM_Mode_<bsnp>[_<vi>_<ai>]_<name>() Where <bsnp> is the BSW Scheduler Name Prefix of the calling BSW module derived from the BswModuleDescriptions short name, <vi> is the vendorId of the calling BSW module, <ai> vendorApiInfix of the calling BSW module and <name> name is the mode group.	
Service ID	0x07	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	NA	
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Rte_ModeType_<M>	NA

Description	Provides the currently active mode of a required (requiredModeGroup) ModeDeclarationGroupPrototype.
Preconditions	SchM_Init API needs to be called before SchM_Mode is called.
Configuration Dependency	The existence of an accessedModeGroup association to a providedModeGroup or requiredModeGroup ModeDeclarationGroupPrototype will result in the generation of a SchM_Mode API.

6.3.3.6 SchM_Switch

Function Name	SchM_Switch	
Syntax:	Std_ReturnType SchM_Switch_<bsnp>[_<vi>_<ai>]_<name>(IN Rte_ModeType_<M> <mode>) Where <bsnp> is the BSW Scheduler Name Prefix of the calling BSW module derived from the BswModuleDescriptions short name, <vi> is the vendorId of the calling BSW module, <ai> vendorApiInfix of the calling BSW module and <name> name is the mode group.	
Service ID	0x06	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	<mode>	
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	**SCHM_E_OK:** The mode switch request is successfully delivered. **SCHM_E_LIMIT:** The mode switch request is dropped as the queue is full.
Description	Initiate a mode switch. The SchM_Switch API call is used for sending a mode switch notification by a Basic Software Module.	
Preconditions	SchM_Init API needs to be called before SchM_Switch is called.	

Configuration Dependency	The existence of a managedModeGroup association to a providedModeGroup ModeDeclarationGroupPrototype will result in the generation of a SchM_Switch API.

6.3.3.7 SchM_Trigger

Function Name	SchM_Trigger	
Syntax:	without queuing support: void SchM_Trigger_<bsnp>[_<vi>_<ai>]_<name>() with queuing support: Std_ReturnType SchM_Trigger_<bsnp>[_<vi>_<ai>]_<name>() <bsnp> Name of the BSW Scheduler Name Prefix. The name of BswModuleDescription if not configured. <vi> VendorId value of the calling BSW module <ai> vendorApiInfix value of the calling BSW module <Name> Name of the ReleasedTrigger	
Service ID	0x09	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	NA	
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	with queuing support: Std_ReturnType	with queuing support: RTE_E_OK : The API call is successfully completed
		with queuing support: RTE_E_LIMIT : Queue is full in the Queued method
Description	This function calls BswSchedulableEntity(s) in the external BswModuleDescription.	
Preconditions	SchM_Init API call should be completed before SchM_Trigger is called.	
Configuration Dependency	IssuedTrigger at the Source references ReleasedTrigger and IssuedTrigger on the Sink end references RequiredTrigger. SchM_Trigger API is created when BswExternalTriggerConfig references ReleasedTrigger, and	

	BswRequiredTriggerConnection maps the two triggers.

6.3.3.8 Enhanced SchM_Mode

Function Name	Enhanced SchM_Mode	
Syntax:	<return> SchM_Mode_<bsnp>[_<vi>_<ai>]_<name>(OUT<previousmode>,OUT <nextmode>) <bsnp> is the BSW Scheduler Name Prefix <vi> is the vendorId of the calling BSW module, <ai> vendorApiInfix of the calling BSW module and <name> is the required (requiredModeGroup) ModeDeclarationGroupPrototype name.	
Service ID	0x07	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	NA	
Parameters (Inout)	None	
Parameters (Out)	<previousmode>	
	<nextmode>	
Return Value	Rte_ModeType_<M>	NA
Description	Provides the currently active mode of a mode switch port and also the mode currently being left and the mode being entered.	
Preconditions	SchM_Init API needs to be called before Enhanced SchM_Mode is called.	
Configuration Dependency	accessedModeGroup association to a providedModeGroup or requiredModeGroup ModeDeclarationGroupPrototype given that the attribute enhancedModeApi of the BswModeSenderPolicy set to true shall result in the generation of an Enhanced SchM_Mode API	

6.3.3.9 SchM_ActMainFunction

Function Name	SchM_ActMainFunction

<table>
<tr><td>Syntax:</td><td colspan="2">without queuing support:
void
SchM_ActMainFunction_<bsnp>[_<vi>_<ai>]_<name>()

with queuing support:
Std_ReturnType
SchM_ActMainFunction_<bsnp>[_<vi>_<ai>]_<name>()

Where here
<bsnp> is the BSW Scheduler Name Prefix, <vi> is the vendorId of the calling BSW module, <ai> vendorApiInfix of the calling BSW module and <name> is the associated BswInternalTriggeringPoint short name.

<bsnp> Name of the BSW Scheduler Name Prefix. The name of BswModuleDescription if not configured.
<vi> VendorId value of the calling BSW module
<ai> vendorApiInfix value of the calling BSW module
<Name> Name of the BswInternalTriggeringPoint</td></tr>
<tr><td>Service ID</td><td colspan="2">0x05</td></tr>
<tr><td>Sync/Async</td><td colspan="2">NA</td></tr>
<tr><td>Reentrancy</td><td colspan="2">NA</td></tr>
<tr><td>Parameters (In)</td><td colspan="2">NA</td></tr>
<tr><td>Parameters (Inout)</td><td colspan="2">None</td></tr>
<tr><td>Parameters (Out)</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">Return Value</td><td rowspan="2">with queuing support:
Std_ReturnType</td><td>with queuing support:
RTE_E_OK
: The API call is successfully completed</td></tr>
<tr><td>with queuing support:
RTE_E_LIMIT
: Queue is full in the Queued method</td></tr>
<tr><td>Description</td><td colspan="2">This function calls other BswSchedulableEntity (s) inside the BswModuleDescription.</td></tr>
<tr><td>Preconditions</td><td colspan="2">Before SchM_ActMainFunction is called, SchM_Init API call should be completed.</td></tr>
<tr><td>Configuration Dependency</td><td colspan="2">SchM_ActMainFunction API is created when ActivationPoint references InternalTriggeringPoint, BswInternalTriggerOccurredEvent references InternalTriggeringPoint, and BswInternalTriggerConfig references InternalTriggeringPoint.</td></tr>
</table>

6.3.3.10 SchM_SwitchAck

Function Name	SchM_SwitchAck	
Syntax:	Std_ReturnType SchM_SwitchAck_<bsnp>[_<vi>_<ai>]_<name>() <bsnp> is the BSW Scheduler Name Prefix, <vi> is the vendorId of the calling BSW module, <ai> vendorApiInfix of the calling BSW module and <name> is the required (requiredModeGroup) ModeDeclarationGroupPrototype name.	
Service ID	0x08	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	NA	
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	**SCHM_E_NO_DATA:** No error took place during the API run. **SCHM_E_TRANSMIT_ACK:** During the mode switch, BswSchedulableEntitys was run and the disabling mode was switched to a new mode.
Description	The SchM_SwitchAck API takes no parameters – the return value is used to indicate the acknowledgement status to the caller.	
Preconditions	SchM_Init API needs to be called before SchM_SwitchAck is called.	
Configuration Dependency	Acknowledgement is enabled for a provided (providedModeGroup) ModeDeclarationGroupPrototype by the presence of an ackRequest attribute of the BswModeSender-Policy	

6.3.3.11 Schm_CData

Function Name	Schm_CData
Syntax:	void Schm_CData_<bsnp>[_<vi>_<ai>]_<name>() Where here

	<bsnp> is the BSW Scheduler Name Prefix, <vi> is the vendorId of the calling BSW module, <ai> vendorApiInfix of the calling BSW module and <Name> is the shortName of the ParameterDataPrototype.	
Service ID	NA	
Sync/Async	NA	
Reentrancy	NA	
Parameters (In)	NA	
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Implementation Data Type	NA
Description	The Schm_CData API provides access to the defined calibration parameter Within a Basic Software Module. The actual data values for A Basic Software Module instance may be set after component compilation.	
Preconditions	SchM_Init API needs to be called before Schm_CData is called.	
Configuration Dependency	An Schm_CData API shall be created for each defined ParameterDataPrototype in the role perInstanceParameter	

6.3.4 E2E APIs

6.3.4.1 Single channel wrapper routines

6.3.4.1.1 E2EPW_Write

Function Name	E2EPW_Write
Syntax:	uint32 E2EPW_Write_<p>_<o>(Rte_Instance <instance>, <data>) Where <p> is the port name and <o> the VariableDataPrototype name.
Service ID	NA
Sync/Async	Synchronous
Reentrancy	Non Reentrant

Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	<data>	Data to be protected and sent
Parameters (Out)	NA	
Return Value	uint32	The byte 0(lowest byte) is the status of Rte_Write function: RTE_E_COM_STOPPED RTE_E_SEG_FAULT The byte 1 is the status of runtime checks done within E2E Protection Wrapper function: E2E_E_INPUTERR_NULL E2E_E_INTERR E2E_E_OK E2E_E_INVALID The byte 2 is the return value of E2E_PXXProtect function: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_INVALID RTE_E_OK The byte 3 is a placeholder for future use and takes the following values: E2E_E_OK E2E_E_INVALID
Description	Initiates a safe explicit sender-receiver transmission of a safety-related data Element with data semantic. It protects data with E2E Library function E2E_PXXProtect and then it calls the corresponding Rte_Write function.	
Preconditions	E2EPW_WriteInit needs to be called before E2EPW_Write.	
Configuration Dependency	This Api will be generated if USES-END-TO-END-PROTECTION is set to true in the PROVIDED-COM-SPEC of corresponding P-PORT-PROTOTYPE and END-TO-END profile should be configured for this P-PORT-PROTOTYPE and VARIABLE-DATA-PROTOTYPE.	

6.3.4.1.2 E2EPW_Read

Function Name	E2EPW_Read

Syntax:	uint32 E2EPW_Read_<p>_<o>(Rte_Instance <instance>, <data>) Where <p> is the port name and <o> the VariableDataPrototype name.		
Service ID	NA		
Sync/Async	Synchronous		
Reentrancy	Non Reentrant		
Parameters (In)	<instance>	SW-C instance	
Parameters (Inout)	NA		
Parameters (Out)	<data>	Data to received	
Return Value	uint32		The byte 0(lowest byte) is the status of Rte_Read function: RTE_E_INVALID RTE_E_MAX_AGE_EXCEEDED RTE_E_NEVER_RECEIVED RTE_E_UNCONNECTED RTE_E_OK The byte 1 is the status of runtime checks done within E2E Protection Wrapper function, plus including bit extension checks: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2EPW_E_DESERIALIZATION E2E_E_INTERR E2E_E_OK E2E_E_INVALID The byte 2 is the return value of E2E_PXXProtect function: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_INVALID RTE_E_OK The byte 3 is the value of E2E_PXXReceiverStatusType Enumeration, representing the result of the verification of the Data in E2E Profile XX, determines by the Check

		function. E2EPXXSTATUS_NONEWDATA E2EPXXSTATUS_WRONGCRC E2EPXXSTATUS_INITIAL E2EPXXSTATUS_REPEATED E2EPXXSTATUS_OKSOMELOST E2EPXXSTATUS_OK E2EPXXSTATUS_WRONGSEQUENCE E2E_E_INVALID
Description	Performs safe explicit sender-receiver transmission of a safety-related communication data element with data semantic. The function calls optionally the corresponding function RTE_IsUpdated, Then it calls corresponding function Rte_Read and then checks received data with E2E_PXXCheck.	
Preconditions	E2EPW_ReadInit needs to be called before E2EPW_Read.	
Configuration Dependency	This Api will be generated if USES-END-TO-END-PROTECTION is set to true in the REQUESTED-COM-SPEC of corresponding R-PORT-PROTOTYPE and END-TO-END profile should be configured for this R-PORT-PROTOTYPE and VARIABLE-DATA-PROTOTYPE.	

6.3.4.1.3 E2EPW_WriteInit

Function Name	E2EPW_WriteInit	
Syntax:	uint8 E2EPW_WriteInit_<p>_<o>(Rte_Instance <instance>) Where <p> is the port name and <o> the VariableDataPrototype name.	
Service ID	0x15	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	uint8	E2E_E_INTERR E2E_E_OK

Description	The function reinitializes the corresponding data structure after a detected error or at start up.
Preconditions	Rte_Start API needs to be called before E2E_WriteInit is called.
Configuration Dependency	This Api will be generated if USES-END-TO-END-PROTECTION is set to true in the PROVIDED-COM-SPEC of corresponding P-PORT-PROTOTYPE and END-TO-END profile should be configured for this P-PORT-PROTOTYPE and VARIABLE-DATA-PROTOTYPE.

6.3.4.1.4 E2EPW_ReadInit

Function Name	E2EPW_WriteInit	
Syntax:	uint8 E2EPW_ReadInit_<p>_<o>(Rte_Instance <instance>) Where <p> is the port name and <o> the VariableDataPrototype name.	
Service ID	0x16	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	uint8	E2E_E_INTERR E2E_E_OK
Description	The function reinitializes the corresponding data structure after a detected error or at start up.	
Preconditions	Rte_Start API needs to be called before E2E_ReadInit is called.	
Configuration Dependency	This Api will be generated if USES-END-TO-END-PROTECTION is set to true in the REQUESTED-COM-SPEC of corresponding R-PORT-PROTOTYPE and END-TO-END profile should be configured for this R-PORT-PROTOTYPE and VARIABLE-DATA-PROTOTYPE.	

6.3.4.2 Redundant wrapper routines

6.3.4.2.1 E2EPW_Write1

Function Name	E2EPW_WriteInit		
Syntax:	uint32 E2EPW_Write1_<p>_<o>(Rte_Instance <instance>, <data>) Where <p> is the port name and <o> the VariableDataPrototype name.		
Service ID	NA		
Sync/Async	Synchronous		
Reentrancy	Non Reentrant		
Parameters (In)	<instance>	SW-C instance	
Parameters (Inout)	<data>	Data to be protected and sent	
Parameters (Out)	NA		
Return Value	uint32		The byte 0(lowest byte) is equal to E2E_E_OK (because Rte_Write is not invoked) The byte 1 is the status of runtime checks done within E2E Protection Wrapper function: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_INVALID The byte 2 is the return value of E2E_PXXProtect function: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_INVALID RTE_E_OK The byte 3 is a placeholder for future use and takes the following values: E2E_E_OK E2E_E_INVALID

Description	It protects data with E2E Library function E2E_PXXProtect. It does not call the corresponding Rte_Write function.
Preconditions	E2EPW_WriteInit needs to be called before E2EPW_Write1.
Configuration Dependency	This Api will be generated if USES-END-TO-END-PROTECTION is set to true in the PROVIDED-COM-SPEC of corresponding P-PORT-PROTOTYPE and END-TO-END profile should be configured for this P-PORT-PROTOTYPE and VARIABLE-DATA-PROTOTYPE.

6.3.4.2.2 E2EPW_Write2

Function Name	E2EPW_Write2	
Syntax:	uint32 E2EPW_Write2_<p>_<o>(Rte_Instance <instance>, <data>) Where <p> is the port name and <o> the VariableDataPrototype name.	
Service ID	NA	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	<instance>	SW-C instance
Parameters (Inout)	<data>	Data to be protected and sent
Parameters (Out)	NA	
Return Value	uint32	The byte 0(lowest byte) is the status of Rte_Write function: RTE_E_COM_STOPPED RTE_E_SEG_FAULT RTE_E_OK The byte 1 is the status of runtime Protects done within E2E Protection Wrapper function: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_REDUNDANCY The byte 2 is the return value of

		E2E_PXXProtect function: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_INVALID The byte 3 is a placeholder for future use and takes the following values: E2E_E_OK E2E_E_INVALID
Description	Initiates a safe explicit sender-receiver transmission of a safety-related data element with data semantic. It protects data with E2E Library function E2E_PXXProtect, compares the computed control fields with ones computed by Write1, and then it calls the corresponding Rte_Write function.	
Preconditions	E2EPW_WriteInit needs to be called before E2EPW_Write2.	
Configuration Dependency	This Api will be generated if USES-END-TO-END-PROTECTION is set to true in the PROVIDED-COM-SPEC of corresponding P-PORT-PROTOTYPE and END-TO-END profile should be configured for this P-PORT-PROTOTYPE and VARIABLE-DATA-PROTOTYPE.	

6.3.4.2.3 E2EPW_Read1

Function Name	E2EPW_Read1		
Syntax:	uint32 E2EPW_Read1_<p>_<o>(Rte_Instance <instance>, <data>) Where <p> is the port name and <o> the VariableDataPrototype name.		
Service ID	NA		
Sync/Async	Synchronous		
Reentrancy	Non Reentrant		
Parameters (In)	<instance>	SW-C instance	
Parameters (Inout)	NA		
Parameters (Out)	<data>	Data to received	
Return Value	uint32		The byte 0 (lowest byte) is the status of Rte_Read function: RTE_E_INVALID RTE_E_MAX_AGE_EXCEEDED

		RTE_E_NEVER_RECEIVED RTE_E_UNCONNECTED RTE_E_OK The byte 1 is the status of runtime checks done within E2E Protection Wrapper function: E2E_E_INPUTERR_NULL E2E_E_INTERR E2EPW_E_DESERIALIZATION E2E_E_OK E2E_E_INVALID The byte 2 is the return value of E2E_PXXCheck function: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_INVALID The byte 3 is value of E2E_PXXReceiverStatusType Enumeration, representing the result of the verification of the Data in E2E Profile XX, determined by the check function. E2EPXXSTATUS_NONEWDATA E2EPXXSTATUS_WRONGCRC E2EPXXSTATUS_INITIAL E2EPXXSTATUS_REPEATED E2EPXX_STATUS_OK E2EPXXSTATUS_OKSOMELOST E2EPXXSTATUS_WRONGSEQUENCE E2E_E_INVALID
Description	Performs safe explicit sender-receiver transmission of a safety-related communication data element with data semantic. The function calls optionally the corresponding function RTE_IsUpdated, Then it calls corresponding function Rte_Read and then checks received data with E2E_PXXCheck.	
Preconditions	E2EPW_ReadInit needs to be called before E2EPW_Read1.	
Configuration Dependency	This Api will be generated if USES-END-TO-END-PROTECTION is set to true in the REQUESTED-COM-SPEC of corresponding R-PORT-PROTOTYPE and END-TO-END profile should be configured for this R-PORT-PROTOTYPE and VARIABLE-DATA-PROTOTYPE.	

6.3.4.2.4 E2EPW_Read2

Function Name	E2EPW_Read2		
Syntax:	uint32 E2EPW_Read2_<p>_<o>(Rte_Instance<instance>, <data>) Where <p> is the port name and <o> the VariableDataPrototype name.		
Service ID	NA		
Sync/Async	Synchronous		
Reentrancy	Non Reentrant		
Parameters (In)	<instance>	SW-C instance	
Parameters (Inout)	NA		
Parameters (Out)	<data>	Data to received	
Return Value	uint32	The byte 0 (lowest byte) equal to RTE_E_OK (because Rte_Read is not invoked) The byte 1 is the status of runtime checks done within E2E Protection Wrapper function: E2E_E_INPUTERR_NULL E2E_E_INTERR E2EPW_E_DESERIALIZATION E2E_E_OK E2E_E_INVALID The byte 2 is the return value of E2E_PXXCheck function: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_INVALID The byte 3 is value of E2E_PXXReceiverStatusType Enumeration, representing the result of the verification of the Data in E2E Profile XX, determined by the check function. E2EPXXSTATUS_NONEWDATA	

		E2EPXXSTATUS_WRONGCRC E2EPXXSTATUS_INITIAL E2EPXXSTATUS_REPEATED E2EPXXSTATUS_OK E2EPXXSTATUS_OKSOMELOST E2EPXXSTATUS_WRONGSEQUENCE E2E_E_INVALID
Description	The function re-checks the data received with corresponding function Read1 by means of execution of E2E_PXXCheck.	
Preconditions	E2EPW_ReadInit needs to be called before E2EPW_Read2.	
Configuration Dependency	This Api will be generated if USES-END-TO-END-PROTECTION is set to true in the REQUESTED-COM-SPEC of corresponding R-PORT-PROTOTYPE and END-TO-END profile should be configured for this R-PORT-PROTOTYPE and VARIABLE-DATA-PROTOTYPE.	

6.3.4.3 E2E COM CALLOUTS

6.3.4.3.1 IPDU_e2EProtect_<IPDU ID>

Function Name	IPDU_e2EProtect_<IPDU ID>	
Syntax:	FUNC(Boolean, COM_APPL_CODE) IPDU_e2EProtect_<IPDU ID>(PduIdType id, P2VAR (uint8, AUTOMATIC, COM_VAR_NOINIT) ipduData)	
Service ID	NA	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	<instance>	SW-C instance
	<IPDU ID>	Pdu Id
	<ipduData>	Pdu Data
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Boolean	TRUE FALSE

Description	Initiates a safe explicit sender-receiver transmission of a safety-related data Element with data semantic. It protects data with E2E Library function E2E_PXXProtect and then it calls the corresponding Rte_Write function.
Preconditions	NA
Configuration Dependency	This Api will be generated if END-TO-END-PROTECTION-I-SIGNAL-I-PDU is configured in PROVIDED-COM-SPEC.

6.3.4.3.2 IPDU_E2Echeck_<IPDU ID>

Function Name	IPDU_E2Echeck_<IPDU ID>	
Syntax:	FUNC(Boolean, COM_APPL_CODE) IPDU_e2ECheck_<IPDU ID>(PduIdType id, P2CONST (uint8, AUTOMATIC, COM_VAR_NOINIT) ipduData)	
Service ID	NA	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	<instance>	SW-C instance
	<id>	Pdu Id
	<ipduData>	Pdu Data
Parameters (Inout)	NA	
Parameters (Out)	NA	
Return Value	Boolean	TRUE FALSE
Description	Initiates a safe explicit sender-receiver transmission of a safety-related data Element with data semantic. It protects data with E2E Library function E2E_PXXCheck and then it calls the corresponding Rte_Read function.	
Preconditions	NA	
Configuration Dependency	This Api will be generated if END-TO-END-PROTECTION-I-SIGNAL-I-PDU is configured in PROVIDED-COM-SPEC.	

7. Generator

7.1 Generator Option

Options	*Description*
-O/-Output	Set up the output directory.
-I/-Input	Set up Input Directory. (Treat all Arxml in the Input Directory including child folders as Input.)
-L/-Log	Print the log of generation process as a file (Rte.log)
-H/-Help	Print user guides on the screen.
-V/-Version	Print CopyRight information and Tool Version on the screen
-GenMemMap	Create MemMap Header of the Application ([AppISWC]_MemMap.h)
-GenMemMapOption::Tasking	Create Rte_MemMap.h Header File for Tasking Compiler
-GenMemMapOption::Ghs	Create Rte_MemMap.h Header File for GreenHill Compiler
-GenMemMapOption::GhsFreescale	Create Rte_MemMap.h Header File for GreenHill Compiler (use Os_memmap.h Header File provided by Freescale OS).
-StrictConfigChk	Skip all validation checks
-StrictUnconnRPortChk	Skip validation check for unconnected RPort
-StrictInitValChk	Skip validation check of the InitialValue
-INT	Use Integer only during DataConversion
-IOC	Create Ioc Arxml for Inter Partition Communication
-IOC::GenEcudOs	Create Gen_Ecud_Os.arxml by adding Osloc Configuration for Inter Partition Communication to the existing Os Ecud Arxml.
-Validate	Run reinforced validation check
-OsSecPerTick=[OsSecondsPerTickValue]	Apply the OsSecondsPerTick value to all Counters
-OsSecPerHwTick=[OsSecondsPerTickValue]	Apply the OsSecondsPerTick value to all HwCounters

-CustomPrefix	Skip the default prefix of AutoEver for Os Object (OsConf_[OsObjectType])
-TaskPrefix=[OsTaskPrefixString]	Designate the prefix of a task. This option should be used together with the CustomPrefix option.
-ErrorMsgOnly	Print Error Message only (skip Warning)
-ImmediateBufferUpdate=[true/false]	If True, data is actually transferred before and after Runnable. If False, if it belongs to the same preemption area, it is actually sent from the front and the back of the runnable, and if it belongs to the other preemption area, it is actually sent from the front and the back of the task.
-ImplicitSRMacro	GlueCode of Implicit Sender Receiver is printed not in the form of a function but in the form of Macro.
-NoRteReceiverPullCB	In a specific use case, reduce the use of Rte_ReceiverPullCB to generate optimized code for Sender Receiver Inter-partition communication.
-ParameterDirection	Activate the use of IN, OUT, and INOUT macro. Use IN, OUT, and INOUT macro to create Rte API and Runnable Prototype.

7.2 Generator Error Message

Number	*Message*
1	Name of output directory is not given along with -O option. Name of input directory is not given along. with -Input option. This error occurs, when the output directory is not mentioned along with -O/-OUTPUT option.
	The input file <file name> is repeated more than once. Tool provides the above warning, if the input ECU Configuration Description File having same name is repeated twice. * This Error can be reported whenever the generator faces both error conditions.
2	Option is Invalid. This error occurs, when the options given on command line are other than (-O/-OUTPUT, -V/-VERSION, -H/-HELP, -L/-LOG, -GenMemMap, -C /-CONTRACT, - BSC / -BASIC SOFTWARE SCHEDULER CONTRACT PHASE, - PCC / - PREBUILD DATA SET CONTRACT PHASE, -BSG / - BASIC SOFTWARE SCHEDULER GENERATION PHASE, -G / - RTE GENERATION PHASE, - PCG / - PREBUILD DATA SET GENERATION PHASE, - PBG / - POST BUILD DATA SET GENERATION PHASE, -EXCLFUNC, -E2E_SINGLE, -E2E_REDUNDANT, -E2E_BYPASS, -CSMACRO, StrictConfigChk, StrictUnconnRPortChk, StrictInitValChk,-Error, -IOC).
3	Invalid output directory <output directory name> as the file with same name exists. This error occurs, if the name of the output directory mentioned is same as that of filename in the working directory.
4	The input directory path <input_dir> provided along with <-i/-input> option does not exist.

	This error occurs, if the input directory path <$input_dir> provided along with <-i/-input> option does not exist.
5	Invalid output directory. This error occurs, if the mentioned output directory is Invalid.
6	File <file name> does not exist. This error occurs, if input file name mentioned in the command line is not present in the working directory.
7	The ECU Configuration Description File / Atomic SW-Component Description File are not provided as input to the Tool. This error occurs, if output directory is mentioned along with –O option but ECU Configuration Description File / Atomic SW-Component Description File is not provided as input to the Generation Tool.
8	Cannot open <file name> file. This error occurs, when Log file (Rte.log) could not be opened.
9	Both -CONTRACT and -GENERATION options are specified. Rte Generation Tool can execute in one phase at a time. This error occurs when both -CONTRACT and -GENERATION options are specified. Rte Generation Tool can execute in one phase at a time.
10	The LOWER-LIMIT value <VALUE> and UPPER-LIMIT value <VALUE> in COMPU-SCALE for ENUM Datatype <DATA TYPE> should be identical as the category is TEXTTABLE. Tool will provide an error message if the value configured for lower limit and upper limit of a compu scale are different when the category is TEXTTABLE.
11	The VT value configured within the COMPU-CONST class has to be unique in the whole AUTOSAR system. Tool will provide an error message if the value configured for VT element of each compu const class is not unique.
12	ClientId (NUMERICAL-VALUE-SPECIFICATION) is not configured for the Request, under the container ISIGNAL for corresponding SYSTEM-SIGNAL. This error occurs if, ClientId (NUMERICAL-VALUE-SPECIFICATION) is not configured for the Request, under the container ISIGNAL for corresponding SYSTEM-SIGNAL.
13	More than one MODE-DECLARATION-GROUPS configured with same SHORT-NAME <ShortName> and different MODE-DECLARATIONS. Tool will provide an error message if multiple ModeDeclarationGroups with same short name but different ModeDeclarations are configured.
14	BASE-TYPE-REF or IMPL-TYPE-REF parameter is not configured for IMPLEMENTATION-DATA-TYPE-ELEMENT<Path> within the SW-POINTER-TARGET-PROPS. This error occurs, if the parameter Sw-Base-Type or Impl-type-ref is not configured within the container Sw-Target-Props of Sw-Definition-Condition.
15	The SYS-REF Parameter value <VALUE> in SW-SYSCOND for this <'PATH'> is not valid. This error occurs, if the Reference value in SW-SYSCOND of (RUNNABLE ENTITY, BSW-SCHEDULABLE-ENTITY, and EXCLUSIVE AREA) specified is not valid.

	The VENDOR-ID parameter in BSW-IMPLEMENTATIONis not configured for BSW-MODULE-DESCRIPTION * This Error can be reported whenever the generator faces both error conditions.
16	Both SchmEnableMacroname and SchMDisableMacroName should be configured, when BswExAreaImpl is configured as USER_DEFINED_MACRO 〈Exclusive Area Path〉. This error occurs, if either SchmEnableMacroname or SchMDisableMacroName is not configured when BswExAreaImpl is configured as USER-DEFINED-MACRO.
17	The SHORT-NAME 〈Shortname〉 for all POST-BUILD-VARIANT-CRITERION must be unique. Tool will provide an error message if duplicate name exists for post build variant criterions.
18	The SHORT-NAME 〈Shortname〉 across SW-SYSTEMCONST must be unique. Tool will provide an error message if duplicate name exists for Sw-System-Const.
19	HANDLE-OUT-OF-RANGE-STATUS should not be set to INDICATE for the Queued Receiver Com Spec. Tool will provide an error message if HANDLE-OUT-OF-RANGE-STATUS is set to INDICATE for the Queued Receiver Com Spec.
20	The POST-BUILD-VARIANT-CRITERION-VALUE-SET-REF parameter is not configured in PREDEFINED-VARIANT 〈pre_path〉. Tool will provide an error message if Post Build variant criterion value set reference is not set.
21	More than one same SHORT-NAME 〈Short name〉 in SW-SYSTEMCONSTANT-VALUE-SET value exists. Tool will provide an error message if duplicate pre-compile criterion value is present.
22	The SW-SYSTEMCONSTANT-VALUE-SET-REF parameter is not configured in SW-SYSTEMCONSTANT-VALUE-SET 〈Shortname〉. Tool will provide an error message if sw-systemconstant value set path is not referred in Sw-System-Constant-Value-Set.
23	More than one Background task is configured. This error occurs, if more than one background task exists.
24	Background Task is not having the lowest priority of the core. This error occurs, if Background Task is not assigned lowest priority.
25	More than one OsAlarm is referred to activate the same Basic Task. This error occurs, if more than one OsAlarm reference is configured to activate the basic task basic.
26	Different OsAlarms or OsScheduleTableExpiryPoints should be referred to set the OsEvents in all the RteEventtoTaskMappings which are mapped to same extended task. If All the RteEventtoTaskMappings are referring different OsEvents Different Period or Offset values are configured to set the same OsEvent. This error occurs, if in an extended task RteEventtoTaskMapping or RteBswEventtoTaskMapp--Ings should refer to different OsAlarms or OsScheduleTableExpiryPoints.
27	Event Type 〈EventType〉 is referred in more than one RteBswEventToTaskMapping.

	This error occurs, if in an Event is referred in more than one RteBswEventtoTaskMappings.
28	The 'RteEventRef' configured in 'RteEventtoTaskMapping' or "RteBswEventto- -TaskMappings" is not a valid. Tool will provide an error message if the event in RteEventRef configured is not a valid.
29	The parameters RteUsedOsAlarmRef and RteUsedOsScheduleTableExpiryPoint -Ref or RteBswUsedOsAlarmRef and RteBswUsedOsScheduleTableExpiryPointRef should not be configured when the RTE Event or BSW Event referred using the parameter 'Rteevent' is not TIMING-EVENT or BACKGROUND-EVENT. Tool will provide an error message if the Event mapped in RteEventRef/RteBswEventRef of RteEventtoTaskMapping/RteBswEventtoTaskMapping is neither a TIMINIG-EVENT Nor BACKGROUND-EVENT in Which OsAlarm or ScheduleTableExpiryPoint is referring.
30	Duplicate RtePositionInTask configured for OsTask〈PATH〉. Tool will provide an error message if same RtePositionInTask is configured for an OsTask.
31	Both the parameters RteUsedOsAlarmRef/RteBswUsedOsAlarmRef and RteUsedOsScheduleTableExpiryPointRef/ RteBswUsedOsScheduleTableExpiryPointRef are configured, while the RteEventRef/ RteBswEventRef are referring to TIMING-EVENT or BACKGROUND-EVENT in RteEventtoTaskMapping/ RteBswEventtoTaskMapping. Tool will provide an error message if both OsAlarm and OsScheduleTableExpiryPoint are referred in RteEventtoTaskMapping or if both OsAlarm and OsScheduleTableExpiryPoint are referred in RteBswEventtoTaskMapping.
32	More than one RteEventtoTaskMappings which are referring to the same Extended OsTask but UsedOsEventRef is not configured in all the RteEventtoTaskMappings in OsTask This error occurs, if in Extended Task same OsTask is referred in RteEventtoTaskMappings but UsedOsEventRef is not configured in all the RteEventtoTaskMappings in OsTask.
33	The OsAlarm and OsScheduleTableExpiryPoint mapped to OsTask 〈ostask_path 〉which is not having lowest priority in RteEventtoTaskMapping. Tool will provide an error message if the OsAlarm and OsScheduleTableExpiryPoint are mapped to OsTask which is not having lowest priority in RteEventtoTaskMapping.
34	ActivationOsAlarmRef/ BswActivationOsAlarm Ref should be configured in RteEventtoTaskMapping if minimum start interval is configured 〉0. Tool will provide an error message, if ActivationOsAlarmRef/ BswActivationOsAlarm Ref is not configured in RteEventtoTaskMapping even though minimum start interval is configured 〉0.
35	Runnable Entity is mapped to more than one OsTask, Concurrent activation is forbidden for this i.e. CanBeInvokedConcurrently Attribute set to false i.e. both the tasks can preempt each other. Tool will provide an error message if Runnable Entity is mapped to more than one OsTask when CanBeInvokedConcurrently Attribute Set to false and the mapped tasks have different task schedule settings.
36	START-ON-EVENT-REF referred in 〈EventType〉 is not a valid RunnableEntity. Tool will provide an error message if invalid runnable entity consisting of event referring to START-ON-EVENT-REF.

37	Schedulable Entity is mapped to more than one OsTask i.e. CanBeInvokedConcurrently attribute set to false. Tool will provide an error message if BswSchedulable Entity is mapped to more than one OsTask when CanBeInvokedConcurrently Attribute Set to false.
38	START-ON-EVENT-REF referred is not a valid Schedulable Entity Path. Tool will provide an error message if invalid Schedulable Entity path is referred in START-ON-EVENT-REF.
39	OsCounter referred in OsAlarm is not valid Os Counter reference. This error occurs, if there is invalid OsCounter reference in OsAlarm.
40	OsCounter referred in OsScheduleTable is not a valid Os Counter reference. This error occurs, if there is invalid OsCounter reference in OsScheduleTable.
41	Same VARIANT-CRITERION-REF parameter<variant-criterion-ref> is referred more than once in EVALUATED-VARIANT-REFS <Evaluated-variant-ref>. Whenever VARIANT-CRITERION path is referenced in VARIANT-CRITERION-REF parameter of EVALUATED-VARIANT-REFS more than once this error will be thrown.
42	The PREDEFINED-VARIANT-REFS<References> is not configured in ECUC-RESOLVER in ECU file. This error occurs, if post build criterion value set path is referred instead of prebuild criterion value set path in ECUC variant Resolver.
43	Mandatory parameter in container is not configured. This error occurs, when the Mandatory parameter in container is not configured.
44	OS Component configuration is not present in the input file(s). This error occurs, if value for RTEEventRef is configured inside RunnableEntityMapping container and OS Component is not present in the input ECU Configuration Description File(s).
45	Mandatory parameter 'parameter name' is not configured in container 'container name'. Tool provides the above error, if the following mandatory parameters are not configured in respective containers. ※ Refer to end of this table
46	Runnable Symbol <Runnable Entity Symbol> provided for the parameter 'RUNNABLE-SYMBOL' should be unique across all the Runnable Entities. This error occurs, if runnable symbol name is not unique across the entire software component description file
47	Enable update value is set to true in NonQueuedRecieverComSpec of Data read access variable type<Com Spec Path>, <Variable Access Path>. This error occurs, if Data Read access variable type is set with enable update true value in NonQueuedRecieverComSpec of R-Port-type.
48	Port <PortPath> is having an M:N connection RTE does not support M:N connections. This error occurs, if port is having multiple sender and multiple receiver connection.
49	Both Synchronous Server Call point and Asynchronous Call point is referring to same client server operation.

	This error occurs, if Both Synchronous Server Call point and Asynchronous Call point are referring to same client server operation.
50	Interface name is not configured for the Port<Port Path>. This error occurs, if Sender receiver/client server interface path is not configured for the mentioned port.
51	PortPath should be configured for ModeSwitchPoints within the modes. This error occurs, if port is not referred in ModeSwitchPoints configured within modes.
52	The Init value within the Non Queue Receiver Com Spec <port> doesn't match with the variable data prototype init value. This error occurs, when different Init Values are configured within Non Queue Receiver Com Spec and Variable Data Prototype.
53	Vendor Id not configured for the BSW Implementation Data Type<impl_path> This error occurs, when Vendor Id not configured for the BSW Implementation Data Type<impl_path>.
54	The queue type configured for variable Data prototype <path >is of type 'MEASUREMENT-POINT'. This error occurs, when queue type parameter is set to MEASUREMENT-POINT in Variable data prototype.
55	DatatypeMappingset is not configured for <Mode Declaration Group>. This error occurs, when Data Type Mapping Set path is not referred in <Mode Declaration Group>.
56	ServerArgumentImplPolicy should be set to UseArrayBaseType only for the data type of category 'ARRAY'. This error occurs, when ServerArgumentImplPolicy is set to UseArrayBaseType when data type category is not of type 'ARRAY'.
57	WAIT-POINT should not be configured for the RUNNABLE-ENTITY <run_path> which is using Exclusive area in the role RUNS-INSIDE-EXCLUSIVE-AREA. This error occurs, when WAIT-POINT is configured for the RUNNABLE-ENTITY <run_path> which is using Exclusive area in the role RUNS-INSIDE-EXCLUSIVE-AREA
58	ServerArgumentImplPolicy should be set to UseVoid only for the< DATA -TYPE> whose data type category is either of primitive or pointer data type. This error occurs, when ServerArgumentImplPolicy is not set to UseVoid for the PRIMITIVE or POINTER Data Type.
59	Different Queue Length is configured for the P-ports sharing the same Runnable <Run-Symbol>. This error occurs, when a client is request from more than one Server of Port Path.
60	A client <component> cannot request from more than one Server of Port Path. This error occurs, when client port is connected to more than one server port for the same operation element.
61	ClientServerOperation has an ArgumentDataPrototype whose ImplementationDataType is of category DataReference and direction is OUT or INOUT This error occurs, when direction is not configured as OUT/INOUT for Argument data prototype

	within ClientServerOperation for the Implementation data type of category 'DATA_REFERENCE'.
62	Queue Length in ServerComSpec of P-PORT should be greater than zero<Path>. This error occurs, when Queue Length in ServerComSpec of P-PORT is configured less than zero.
63	The NV Ram Block is not configured in NvBlockDescriptors <Path>. This error occurs, when the Nv Ram Block is not configured in NvBlockDescriptors <Path>.
64	The Data Type referred in RamBlock and RomBlock of NvBlockDescriptors are of different type. This error occurs, when the data types referred in RamBlock and RomBlock of NvBlockDescriptors are of different compatible types.
65	Data type referred in read nv data written nv data and ram block are not of compatible types. This error occurs, when Variable Data Prototype is referred in read NV data and written NV data are not of compatible types.
66	The TransmissionAcknowledgementRequest attribute for transmission acknowledgment for 1: n communication is configured and is invalid. This error occurs, when TransmissionAcknowledgementRequest is configured for P-Port, in case of 1: n communication.
67	Either SYMBOL or SHORT-LABLE or valid c identifier for VT-ELEMENT should be configured for the COMPU-METHOD if the CATEGORY attribute is configured as TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE or SCALE_RATIONAL_AND_TEXTTABLE. This error occurs, when Either SYMBOL or SHORT-LABLE or valid c identifier for VT-ELEMENT is not configured for the COMPU-METHOD.
68	The timeout value for transmission acknowledgement and wait point is not same which is invalid. This error occurs, when the timeout value and the wait point value are not the same.
69	The Event is referencing a RunnableEntity and is referenced by a WaitPoint which is invalid. This error occurs, when the Event <event> is referencing a RunnableEntity and is referenced by a WaitPoint which is invalid.
70	NATIVE-DECLARATION <Path> configured is not a valid-'C' data type. This error occurs, when a Native declaration path provided, is not pointing to a valid ‘C’ data type.
71	The RunnableEntity that has a WAIT-POINT must not be referenced by an event_actual_name. This error occurs, when RunnableEntity that has a WAIT-POINT is referenced by an event_actual_name.
73	The HANDLE-TERMIATION-AND-RESTART parameter should be set to CAN-BE-TERMINATED-AND-RESTARTED for the SOFTWARE-COMPONENT-PROTOTYPE which mapped to an ECU-PARTITION with PARTITION-CAN-BE-RESTARTED parameter set to true This error occurs, when the HANDLE-TERMIATION-AND-RESTART parameter is not set to CAN-BE-TERMINATED-AND-RESTARTED for the SOFTWARE-COMPONENT-PROTOTYPE which mapped to an ECU-PARTITION with PARTITION-CAN-BE-RESTARTED parameter set to true

74	A single task cannot be mapped to more than one Partition. Task <Task Path> of <Software Component Name> is mapped to the following Partition <Partitions Path>. This error occurs, when a single task is mapped to more than one partition.
75	ExclusiveAreaMechanism parameter is not configured in ExAreaImpl container of ExclusiveAreaRef of <Path>. This error occurs, when ExclusiveAreaImplMechanism parameter is not configured in ExAreaimpl container of ExclusiveAreaRef.
76	The SWC <Path> should have an internal behavior. This error occurs, when mention SWC path does not have an internal behavior.
77	Data Type is not referred in TypeTRef parameter of Per Instance Parameter of given <path>. This error occurs, when Data Type is not referred in TypeTRef parameter of Per Instance Parameter of given <path>.
78	The nativeDeclaration in SwBaseType referred by ImplementatinDataTypes needs to be configured. This error occurs, when native declaration is not configured for the BASE-TYPE.
79	Type parameter is not configured in Per-Instance_Memory of <Path>. This error occurs, when type parameter is not configured in Per-Instance_Memory.
80	TypeDefinition parameter is not configured in Per-Instance_Memory of< Path >. This error occurs, when type definition parameter is not configured in Per-Instance_Memory.
81	Data Type is not referred in TypeTRef parameter of ArTyped-Per-Instance_Memory of< path >. This error occurs, when data type is not referred in TypeTRef parameter of ArTyped-Per-Instance_Memory.
85	OsScheduleTableExpiryPointOffset is not configured in <path>. This error occurs, when OsScheduleTableExpiryPointOffset is not configured in given path.
86	REQUIRE-PORTS <port_path> Should not be configured for PARAMETER-SW-COMPONENT-TYPE <SWC> This error occurs, when REQUIRE-PORTS is configured for PARAMETER-SW-COMPONENT-TYPE <SWC>.
87	OnTransitionValue is not configured for the MODE-DECLARTION_GROUP of Category Type 'EXPLICIT-ORDER' of given <path>. This error occurs, when OnTransitionValue is not configured for the MODE-DECLARTION_GROUP of Category Type 'EXPLICIT-ORDER' of given <path>.
89	Mapping is not possible for the given vdps as DataProtoMapping direction should be PPort to RPort. This error occurs, when the DataProtoMapping direction is not PPort to RPort and Try to map for given vdps.
91	The value configured for the parameter CATEGORY should be either PROFILE_01 or PROFILE_02 or NONE.

	This error occurs, when the value configured for the parameter CATEGORY will not be either PROFILE_01 or PROFILE_02 or NONE.
92	Counter offset should be configured for the EndToEndProfile. This error occurs, when Counter offset is not configured for the EndToEndProfile.
93	DataIDMode should be configured for the EndToEndProfile. This error occurs, when DataIDMode is not configured for the EndToEndProfile.
94	CrcOffset should be configured for the EndToEndProfile. This error occurs, when CrcOffset is not configured for the EndToEndProfile.
95	DataLength should be configured for the EndToEndProfile. This error occurs, when DataLength is not configured for the EndToEndProfile.
96	Minimum start interval should be a positive number value in runnable path. This error occurs, when Minimum start interval value in runnable entity <Path> is not a positive number.
98	The OsTask path is not referred in RteTaskComMapping container. This error occurs, when The OsTask path is not referred in RteTaskComMapping container.
99	Redundant CRC element found This error occurs, when redundant CRC element found.
100	Redundant COUNTER element found This error occurs, when redundant COUNTER element found.
101	All the R-Ports connected to the specific P-Port should be mapped to same partition. This error occurs, when R-Ports connected to the specific P-Port are not mapped to same partition.
102	All the ModeSwitchEvents of the R-Ports connected to the specific P-Port should be mapped to the same task. This error occurs, when R-Ports connected to the specific P-Port are not mapped to same task.
103	Variable Data Prototype is not referred in Invalidation Policy. This error occurs, when Variable Data Prototype is not referred in Invalidation Policy.
104	Handle Invalid Value is not configured in Invalidation Policy. This error occurs, when Handle Invalid Value is not configured in Invalidation Policy.
105	ModeDisablingDependency cannot be configured for OperationInvokedEvent of <Event Path>. This error occurs, when ModeDisablingDependency is configured for OperationInvokedEvent of mentioned Event Path.
106	Value of a Mode cannot be more than the range of its ModeDeclarationGroup the Mode has the value which is more than the range specified by its ModeDeclarationGroup. This error occurs, when Mode is having the value which is more than the range specified by its ModeDeclarationGroup.
107	The port path configured in ModeScheduleTableMap for Software Component is not a P-Port Path < port_path >. This error occurs, when the port path configured in ModeScheduleTableMap for Software Component is not a P-Port Path.

108	Different Timeout values are configured for the AsynchronousServerCallpoint and for the WaitPoint associated with the AsynchronousServerCallReturnsEvent for this AsynchronousServerCallPoint. Same values should be configured. This error occurs, when the WaitPoint for AsynchronousServerResultCallPoint and timeout for AsynchrnousServerCallPoint is not same.
109	MINIMUM-START-INTERVAL should not be configured and should not set to greater than 0 when CAN-BE-INVOKED-CONCURRENTLY is true. This error occurs, when the MINIMUM-START-INTERVAL is configured when CAN-BE-INVOKED-CONCURRENTLY is true.
110	In DELEGATION-SW-CONNECTOR, INNER PORT path should be SW-COMPONENT-PROTOTYPE and OUTER Port path should be COMPOSITION-SW-COMPONENT-TYPE. But for this DELEGATION-SW-CONNECTOR Inner and Outer port are same
112	The Handle Out Of Range for Queue Receive Com Spec <dataElementRef> should not be INVALIDATE or DEFAULT or EXTERNAL-REPLACEMENT. This error occurs, when the Handle Out Of Range for Queue Receive Com Spec is configured as INVALID or DEFAULT or EXTERNAL-REPLACEMENT.
113	INIT-VALUE should be configured for the VARIABLE-DATA-PROTOTYPE <vdp> when the strictInitialValuesCheck is enabled and SwAddrMethod has a sectionInitializationPolicy set to init. This error occurs, when INIT-VALUE is not configured for the VARIABLE-DATA-PROTOTYPE <vdp> when the strictInitialValuesCheck is enabled and SwAddrMethod has a sectionInitializationPolicy set to init.
114	The INIT-VALUES should be configured for the port in NONQUEUED-RECEIVER-COM-SPEC when filter is configured. This error occurs, when the filter is configured in NONQUEUED-RECEIVER-COM-SPEC and INIT-VALUE is not given in port.
115	RunInsideExclusiveArea is configured for the BswCalledEntity of given path This error occurs, when the RunInsideExclusiveArea is configured for the BswCalledEntity of given path.
116	Runnable Entity or Schedulable Entity mapped to different tasks should have same schedule point settings. This error occurs, when Runnable entity or Schedulable entity <run_ref >mapped to different tasks are not having same schedule point settings.
117	Rte Event <eve_path> should not be mapped to only virtual task. This error occurs, when the Rte Event <eve_path> is mapped to only virtual task.
118	IMPLEMENTATION-DATA-TYPE <impl_path> and APPLICATION-DATA-TYPE <appl_data> configured in SWC <swc_name> are referring to compu methods which are having duplicate UPPER-LIMIT or LOWER-LIMIT for their Compu Scales This error occurs, when IMPLEMENTATION-DATA-TYPE and APPLICATION-DATA-TYPE configured in SWC are referring to compu methods which are having duplicate UPPER-LIMIT or LOWER-LIMIT for their Compu Scales.
119	One or more Runnable Entities of same component is mapped to tasks of different partition. The Events in below sw component has been mapped to different partitions

	This error occurs, when Runnable Entity is mapped to tasks of different partitions.
120	Timing Events which are implemented based on the Schedule table, can not have offset value bigger than zero. This error occurs, when the Runnables of different offset mapped to the same Schedule Table< sched_path > and all offset value must be set as “Zero”.
121	BswEvent is mapped only to virtual task. This error occurs, when the Event <eve_path> is not mapped to virtual task <eve_path>.
122	Priority of the OsTask to which the RunnableEntity is mapped should be greater than the priority of the OsTask to which the RunnableEntity is virtually mapped. This error occurs, when the Priority of the OsTask to which the RunnableEntity is mapped not greater than the priority of the OsTask to which the RunnableEntity is virtually mapped.
123	Runnable reference should not be configured for DREvent which is referred in a waitpoint. This error occurs, when the Runnable reference is configured for DREvent which is referred in a waitpoint.
125	DRead should not be configured with composite data types in Runnable Path <run_ref>. This error occurs, when the DRead is configured with composite data types in Runnable Path <run_ref>.
126	Mandatory parameter SHORTLABEL needs to be configured for variation point of RUNNABLE ENTITY of given Path. This error occurs, when the Mandatory parameter SHORTLABEL is not configured for variation point of RUNNABLE ENTITY of given Path.
127	COMPU-DENOMINATOR for compumethod is configured as 0. It should be a non-zero positive number. This error occurs, when the COMPU-DENOMINATOR for comp method <path1> is configured as 0.
128	Data element with SwImplPolicy as queued cannot be accessed in a variable access in role of DataReceivePointByValues. This error occurs, when the Data element with SwImplPolicy as queued has accessed in a variable access in role of DataReceivePointByValues.
130	The P-PORT-PROTOTYPE-REF <comp_port> configured in the PROVIDER-IREF inside ASSEMBLY-CONNECTOR-PROTOTYPE does not belong to the COMPONENT-PROTOTYPE-REF. This error occurs, when the P-PORT-PROTOTYPE-REF <comp_port> configured in the PROVIDER-IREF inside ASSEMBLY-CONNECTOR-PROTOTYPE belong to the COMPONENT-PROTOTYPE-REF.
131	The R-PORT-PROTOTYPE-REF <comp_port> configured in the REQUESTER-IREF inside ASSEMBLY-CONNECTOR-PROTOTYPE does not belong to the COMPONENT-PROTOTYPE-REF. This error occurs, when the R-PORT-PROTOTYPE-REF <comp_port> configured in the REQUESTER-IREF inside ASSEMBLY-CONNECTOR-PROTOTYPE is belong to the COMPONENT-PROTOTYPE-REF.
132	COMPONENT-PROTOTYPE-REF parameter should be configured in INNER-PORT-IREF container inside DELEGATION-CONNECTOR-PROTOTYPE.

	This error occurs, when the COMPONENT-PROTOTYPE-REF parameter is not configured in INNER-PORT-IREF container inside DELEGATION-CONNECTOR-PROTOTYPE.
133	The PORT-PROTOTYPE-REF <$comp_port> configured in the INNER-PORT-IREF inside DELEGATION-CONNECTOR-PROTOTYPE does not belong to the COMPONENT-PROTOTYPE-REF. This error occurs, when the PORT-PROTOTYPE-REF <$comp_port> configured in the INNER-PORT-IREF inside DELEGATION-CONNECTOR-PROTOTYPE belong to the COMPONENT-PROTOTYPE-REF.
134	The OUTER-PORT-REF <comp_port> configured in the DELEGATION-CONNECTOR-PROTOTYPE does not belong to the composition <composition_path>. This error occurs, when the OUTER-PORT-REF <comp_port> configured in the DELEGATION-CONNECTOR-PROTOTYPE belong to the composition <composition_path>.
135	SwImplPolicy is set to MEASUREMENT-POINT for the variable data prototype of Variable Access Path. This error occurs, when the SwImplPolicy is not set to MEASUREMENT-POINT for the variable data prototype of Variable Access Path.
136	COMPU-NUMERATOR V2 for compumethod is configured as 0. It should be a non-zero positive number. This error occurs, when the COMPU-NUMERATOR V2 for comp method <path1> is configured as 0.
137	Invalidation policy is not configured for the variable data prototype of path <vdp_path>. This error occurs, when the Invalidation policy is not configured for the variable data prototype of path <vdp_path>.
139	Symbol name should be unique. This error occurs, when the Symbol name is not unique
140	Symbol Name should be unique if the SHORT-NAME of the SW-COMPONENT is duplicated. This error occurs, when the Symbol Name is not unique if the SHORT-NAME of the SW-COMPONENT is duplicated.
141	INIT-VALUE should not be configured for the PARAMETER-DATA-PROTOTYPE <vdp> when the strictInitialValuesCheck is enabled. This error occurs, when INIT-VALUE is configured for the PARAMETER-DATA-PROTOTYPE when the strictInitialValuesCheck is enabled.
142	Data conversion not possible between a linear data representation and a texttable data representation. This error occurs, when the conversion take place between a linear data representation and a texttable data representation.
144	The ModeSwitchEvent Runnable/SchedulableEntity <runnable_path > should not be used by another Event for which ModeDisablingDependency is configured in case of EXTENDED Task This error occurs, when ModeSwitchEvent Runnable/SchedulableEntity < runnable_path > is used by another Event for which ModeDisablingDependency is configured in case of EXTENDED Task

145	Within a ModeSwitch Communication multiple P-PORTS should not be connected to the same R-PORT. This error occurs, when in ModeSwitch Communication multiple P-PORTS are connected to the same R-PORT.
146	In Assembly connector both the provider port <pport_path> and requester port <rport_path> belongs to RPort. This error occurs, when In Assembly connector both the provider port and requester port belongs to RPort.
147	In Delegation connector both the inner port <inner_port_path> and outer port<outer_port_path> should belongs to Provider port or Receiver port. This error occurs, when In Delegation connector both the inner port and outer port are not belongs to Provider port or Receiver port.
148	Queue Length in ModeSwitchSender ComSpec of P-PORT <port_path > should be greater than zero. This error occurs, when Queue Length in ModeSwitchSender ComSpec of P-PORT is less than zero.
149	More than one MODE-DECLARATION-GROUP is configured with same SHORT-NAME <mode_grp_sn> but have different INITIAL-MODES. This error occurs, when More than one MODE-DECLARATION-GROUP is configured with same SHORT-NAME but have different INITIAL-MODES.
150	Runnables/ BswSchedulabelEntities which are called Directly within Rte API's should not be mapped to any Ostask. This error occurs, when Runnables which are called Directly within Rte API's are mapped to any Ostask.
151	The Priority of the Task of the event (which has DISABLED-MODE-IREF) should be higher than the priority of the task f ModeSwitchEvent <ms_event> This error occurs, when The Priority of the Task of the event (which has DISABLED-MODE-IREF) has lower priority than the priority of the task of ModeSwitchEvent.
152	ModeDeclarationGroup is mapped to more than one implementation datatype via DATA-TYPE-MAPPING-SET. This error occurs, when ModeDeclarationGroup is mapped to more than one implementation datatype via DATA-TYPE-MAPPING-SET.
153	A communication path from an AUTOSAR Software Component to an ECU Abstraction located on a remote ECU should not be configured. This error occurs, when a communication path from an AUTOSAR Software Component to an ECU Abstraction located on a remote ECU has configured.
154	The MODE-DECLARATION-GROUP should be mapped to PRIMITIVE Datatype within the container MODE-REQUEST-TYPE-MAP in DATA-TYPE-MAPPING-SET. This error occurs, when the MODE-DECLARATION-GROUP is not mapped to PRIMITIVE Datatype within the container MODE-REQUEST-TYPE-MAP in DATA-TYPE-MAPPING-SET.
155	The Event referenced by a WaitPoint should not have DisabledMode Configured.

	This error occurs, when the Event referenced by a WaitPoint have DisabledMode Configured.
156	TIME-OUT in MODE-SWITCHED-ACK and the TIME-OUT in WAIT-POINT should be same. This error occurs, when TIME-OUT in MODE-SWITCHED-ACK and the TIME-OUT in WAIT-POINT are not same.
157	Initvalues for PARAMETER-REQUIRE-COM-SPEC are not configured for unconnected RPort. This error occurs, when Initvalues for PARAMETER-REQUIRE-COM-SPEC are not configured for unconnected RPort.
158	Require ports<port path> are attached to ParameterSwComponentTypes. However only provider ports must be attached to ParameterSwComponentTypes This error occurs, when require ports are attached to ParameterSwComponentTypes.
159	The port referenced in PORT-PROTOTYPE-REF <port_path> for the VARIABLE-ACCESS <access_path> is not a P-PORT. However it should be a P-PORT. This error occurs, when the port referenced in PORT-PROTOTYPE-REF for the VARIABLE-ACCESS is not a P-PORT.
160	Qlength parameter for sender receiver queued communication should be configured as a non-zero positive value. This error occurs when Qlength parameter for sender receiver queued communication is not configured as a non-zero positive value.
161	External-Replacement reference should be configured for the HANDLE-OUT-OF-RANGE EXTERNAL-REPLACEMENT. This error occurs, when External-Replacement reference is not configured for the HANDLE-OUT-OF-RANGE EXTERNAL-REPLACEMENT.
162	INIT-VALUE should be configured for the Variable Data Prototype of given Path<path>. This error occurs, when INIT-VALUE is not configured for the Variable Data Prototype of given Path.
163	Sw-System-Const value reference is not referred in SYSC-REF in component <comp> of port <port_name>. This error occurs, when Sw-System-Const value reference is not referred in SYSC-REF in component of port.
165	More than one Asynchronous Server Call Result Point should not point to same Asynchronous Server Call Point. This error occurs, when More than one Asynchronous Server Call Result Point pointing to same Asynchronous Server Call Point.
166	Application datatype <app_data> should be mapped to any implementation datatype in data mapping set. This error occurs, when Application datatype <app_data> is not mapped to any implementation datatype in data mapping set.
167	Same implementation datatype <used_data_type [-1]}> is configured twice with different implementation symbol. This error occurs, when same implementation datatype is configured twice with different

	implementation symbol.
168	Two or more Implementation Datatype <impl_name> having same symbol and different type declaration. This error occurs, when Two or more Implementation Datatype <impl_name> having same symbol and different type declaration.
169	Either of the MODE-SWITCH-POINT or the MANAGED-MODE-GROUP should be configured when a MODE-DECLARATION-GROUP-PROTOTYPE is synchronized. This error occurs, when either of the MODE-SWITCH-POINT or the MANAGED-MODE-GROUP is not configured when a MODE-DECLARATION-GROUP-PROTOTYPE is synchronized.
170	Duplicate Component for is Configured. This error occurs, when duplicate Component is configured.
171	IMPLEMENTATION-DATA-TYPE with category DATA_REFERENCE should not be referred in APPLICATION-SW-COMPONENT-TYPE. This error occurs, when IMPLEMENTATION-DATA-TYPE with category DATA_REFERENCE is referred in APPLICATION-SW-COMPONENT-TYPE.
172	DATA-PROTOTYPE which is referring to IMPLEMENTATION-DATA-TYPE with ARRAY-SIZE-SEMANTICS as VARIABLE should not be Unqueued and should be of type uint8. This error occurs, when DATA-PROTOTYPE which is referring to IMPLEMENTATION-DATA-TYPE with ARRAY-SIZE-SEMANTICS as VARIABLE is Unqueued and should be of type uint8.
173	RTE does not support receiving with wait points for VARIABLE-DATA-PROTOTYPE <vdp> when SW-IMPL-POLICY is unqueued. This error occurs, when wait points for VARIABLE-DATA-PROTOTYPE are configured when SW-IMPL-POLICY is unqueued.
174	The ApplicationErrors with same ShortName are configured with different value. This error will occur when ApplicationErrors with same name do have different error Codes ApplicationErrors are conflicting.
175	DataReceivedEvent <event> is referenced by a WaitPoint should not reference a VariableDataPrototype <vdp> referenced by an NvDataInterface. This error occurs, when a DataReceivedEvent is referenced by a WaitPoint and references a VariableDataPrototype Referenced by an NvDataInterface.
176	R-Port with an NvDataInterface <$nvm_interface [-1]> is not connected and no NvRequireComSpec with a initValue. This error occurs, when R-Port with an NvDataInterface is not connected and no NvRequireComSpec with an initValue.
177	ON-EXIT Runnables should not have higher value for the Parameter RtePositionInTask than ON-TRANSITION/ON-ENTRY Runnables. ON-TRANSITION Runnables ($runnable_path) should not have higher value for the Parameter RtePositionInTask than ON-ENTRY Runnables. This error occurs, when ON-TRANSITION or ON-EXIT Runnables have higher value for the Parameter RtePositionInTask than the ON-ENTRY or ON-TRANSITION Runnables.

178	Two or more DataReceivedEvents having same VariableDataPrototype trigger different runnable entities mapped to different tasks. This error occurs, when two or more DataReceivedEvents having same VariableDataPrototype trigger different runnable entities mapped to different tasks.
179	Two different runnables which are having wait points connected to the same data received event <dr_eve> should not be mapped to different Os Tasks. This error occurs, when two different runnables which are having wait points connected to the same data received event are mapped to different Os Tasks.
180	More than one DATA-RECEIVED-EVENT under the same INTERNAL-BEHAVIOR which is sharing the same VARIABLE-DATA-PROTOTYPE should not be mapped to different tasks. This error occurs, when More than one DATA-RECEIVED-EVENT under the same INTERNAL-BEHAVIOR which is sharing the same VARIABLE-DATA-PROTOTYPE is mapped to different tasks.
181	Either RteUsedOsAlarmRef or RteUsedOsScheduleTableExpiryPointRef should be configured when the RteEventRef is referring to TIMING-EVENT or BACKGROUND-EVENT in RteEventtoTaskMapping. This error occurs, when Either RteUsedOsAlarmRef or RteUsedOsScheduleTableExpiryPointRef is not configured when the RteEventRef is referring to TIMING-EVENT or BACKGROUND-EVENT in RteEventtoTaskMapping.
184	NativeDeclaration is not configured for the Implementation datatype <value>. This error occurs, when NativeDeclaration is not configured for the Implementation datatype <value>.
185	The same MODE-DECLARATION cannot be referred by both SWC-MODE-SWITCH-EVENT and DISABLED-MODE-IREF. This error occurs, when the same MODE-DECLARATION is referred by both SWC-MODE-SWITCH-EVENT and DISABLED-MODE-IREF.
186	For ON-TRANSITION Events both the MODE-DECLARATIONS should belong to the same MODE-GROUP. This error occurs, when for ON-TRANSITION Events both the MODE-DECLARATIONS are not belonging to the same MODE-GROUP.
187	The interfaces configured for PORT-PROTOTYPE and configured for PORT-PROTOTYPE are not compatible as the number of ModeDeclarations in PORTS is not same. This error occurs, when the interfaces configured for PORT-PROTOTYPE and configured for PORT-PROTOTYPE are not compatible.
188	The interfaces configured for PORT-PROTOTYPE <p_port_path> and configured for PORT-PROTOTYPE <r_port_path>are not compatible as the Shortnames for ModeDeclarations in PORTS are not same. This error occurs, when the interfaces configured for PORT-PROTOTYPE both P port and R port is not compatible as the Shortnames for ModeDeclarations in PORTS are not same.
189	The interfaces configured for PORT-PROTOTYPE <p_port_path> and configured for PORT-PROTOTYPE <r_port_path>are not compatible as the INITIAL-MODE-REF for ModeDeclarations in PORTS are not same.

	This error occurs, when the interfaces configured for PORT-PROTOTYPE both P port and R port is not compatible as the INITIAL-MODE-REF for ModeDeclarations are not same.
190	More than one IMPLEMENTATION-DATA-TYPE configured with same SHORT-NAME and different NATIVE-DECLARATION. This error occurs, when the More than one IMPLEMENTATION-DATA-TYPE configured with same SHORT-NAME and different NATIVE-DECLARATION.
191	More than one ImplementationDataType whose category is STRUCTURE or UNION have same ShortName but number of ImplementationDataTypeElement are different. This error occurs, when the More than one STRUCTURE or UNION IMPLEMENTATION-DATA-TYPE having same SHORT-NAME but number of elements is different.
192	Invalid value is not specified for the Implementation of given Path. This error occurs, when the Invalid value is not specified for the Implementation of given Path.
193	SW-IMPL-POLICY should be configured as QUEUED for the DATA-ELEMENT which is mapped to PDU with ComIpduType equals to TP. This error occurs, when the SW-IMPL-POLICY is not configured as QUEUED for the DATA-ELEMENT which is mapped to PDU with ComIpduType equals to TP.
194	SUPPORTS-MULTIPLE-INSTANTIATION attribute should be set to true for the SWC-INTERNAL-BEHAVIOR This error occurs, when the SUPPORTS-MULTIPLE-INSTANTIATION attribute is not set to true for the SWC-INTERNAL-BEHAVIOR.
195	ENABLE-TAKE-ADDRESS attribute should be set to false for the PORT-REF since multiple instances are configured for the APPLICATION-SW-COMPONENT. This error occurs, when the ENABLE-TAKE-ADDRESS attribute is set to true.
196	Software Component related information should not be configured in Basic Software Scheduler Generation Phase. This error occurs, when the Software Component related information is configured in Basic Software Scheduler Generation Phase.
197	Duplicate Component is configured in same AR-PACKAGE. This error occurs, when the Duplicate Component is configured in same AR-PACKAGE.
198	Symbol Name should be configured if the SHORT-NAME of the SW-COMPONENT is duplicated. This error occurs, when the Symbol Name is not configured but SHORT-NAME of the SW-COMPONENT is duplicated.
199	Execution instances of Runnable entity are mapped to different preemption areas which is invalid. This error occurs, when the Execution instances of Runnable entity are mapped to different pre-emption.
201	The TARGET-DATA-PROTOTYPE-REF/ Sender-Receiver interface referred in EVENT does not match with the DATA-ELEMENT-REF/ Sender-Receiver interface given in the Port. The Sender-Receiver interface referred in EVENT does not match with the Sender-Receiver interface given in the Port .

	This error occurs, when The TARGET-DATA-PROTOTYPE-REF/ Sender-Receiver interface referred in EVENT does not match with the DATA-ELEMENT-REF/ Sender-Receiver interface given in the Port.
202	The parameter 'BSW-ENTITY-REF' in the Container 'SWC-BSW-RUNNABLE-MAPPING' within 'SWC-BSW-MAPPING' cannot refer 'CAT-1' BSW-INTERRUPT-ENTITY SWC-BSW-MAPPING. This error occurs, when the parameter 'BSW-ENTITY-REF' in the Container 'SWC-BSW-RUNNABLE-MAPPING' within 'SWC-BSW-MAPPING' not refer 'CAT-1' BSW-INTERRUPT-ENTITY SWC-BSW-MAPPING.
203	The TARGET-DATA-PROTOTYPE-REF/ Sender-Receiver interface referred in VARIABLE-ACCESS does not match with the DATA-ELEMENT-REF/ Sender-Receiver interface given in the Port. This error occurs, when The TARGET-DATA-PROTOTYPE-REF/ Sender-Receiver interface referred in VARIABLE-ACCESS does not match with the DATA-ELEMENT-REF/ Sender-Receiver interface given in the Port
204	The ClientServerInterface configured for PPortPrototype and the ClientServerInterface configured for RPortPrototype are not compatible as the ShortName of Operation configured for both the ClientServerInterfaces are not same. This error occurs, when the interfaces is not configured for 'P-PORT-PROTOTYPE' <p_port_path> and <r_iface> is not configured for 'R-PORT-PROTOTYPE'.
205	The Variable data prototype of Sender Receiver interfaces configured for P-PORT-PROTOTYPE is not compatible with the Variable data prototype of Sender Receiver interfaces configured for R-PORT-PROTOTYPE. This error occurs, when the Variable data prototype of Sender Receiver interfaces configured for P-PORT-PROTOTYPE is not compatible with the Variable data prototype of Sender Receiver interfaces configured for R-PORT-PROTOTYPE
206	The OperationIref configured in OprationInvokedEvent and the OperationIref in ServerComSpec do not match. This error occurs, when Either the OPERATION-IREF configured in OPERATION-INVOKED-EVENT and the OPERATION-IREF in SERVER-COM-SPEC do not match or OPERATION-IREF is not configured in SERVER-COM-SPEC
208	Same SystemSignal cannot be used by more than one Clients System Signal. This error occurs, when Same SystemSignal is used by more than one Clients System Signal.
209	Different PredefinedVariants should not assign different values to the same PostBuildVariantCriterion for the same RtePostBuildVariantConfiguration PredefinedVariants. This error occurs when Different PredefinedVariants are assigning different values to the same PostBuildVariantCriterion for the same RtePostBuildVariantConfiguration PredefinedVariants.
210	Some of OperationInvokedEvents which invoke same Runnable Entity are mapped to different OsTask or same .OsTask with different Position in Task, or are not mapped to OsTask This error occurs, when OperationInvokedEvents mapped to the same runnable are mapped to different task or to the same task with different position in task.
211	The number of elements in the datatype and the number of elements initialized do not match. This error occurs, when the numbers of elements in the datatype and the number of elements initialized are not same.

212	SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING is configured for the PRIMITIVE datatype. This error occurs, when SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING is configured for PRIMITIVE datatype.
213	SENDER-RECEIVER-TO-SIGNAL-MAPPING is configured for the COMPLEX datatype. This error occurs, when SENDER-RECEIVER-TO-SIGNAL-MAPPING is configured for COMPLEX datatype.
214	The datatype provided in ComSignalType does not match with the datatype provided in ComSignal. This error occurs, when datatype provided in ComSignalType does not match with datatype provided in ComSignal.
215	Number of elements in array does not match with number of signals in signalgroup. This error occurs, when number of elements in array does not match with the number of signals configured in SignalGroup.
216	Number of elements in structure does not match with number of signals in signalgroup. This error occurs, when number of elements in structure does not match with the number of signals configured in SignalGroup.
217	The length of Array does not match with the length provided in ComSignalLength. This error occurs, when length of array is not same as the length configured in ComSignalLength.
218	Sw-System-Const value reference referred in SYSC-REF in SW-SYSCOND of component <component>of port <port> is not valid. This error occurs, when Sw-System-Const value referenced in SYSC-REF in SW-SYSCOND Of component <component name> of port <port name> is not valid.
219	The task referred in EveToTskMap/BswEveToTskMap is invalid as Os container is not present. This error occurs, when the task path specified in EveToTskMap/BswEveToTskMap is invalid as the OsContainer is not present.
220	The MappingDirection for DATA-PROTOTYPE-MAPPING should not be 'SECOND-TO-FIRST' if second data element is used for R port and first data element is used for P port. This error occurs, when the Data-Prototype is from second-to-first when second data element is used for Rport and the first data element is used by Pport.
221	VT-ELEMENT should not be configured for the COMPU-METHOD if the CATEGORY attribute is not configured. This error occurs, when Display format is not configured for the Compu-Method.
222	The MappingDirection for DATA-PROTOTYPE-MAPPING should not be 'FIRST-TO-SECOND' if first data element is used for R port and second data element is used for Pport. This error occurs, when the Data-Prototype is from first-to-second when first data element is used for Rport and the second data element is used by Pport.
223	The START-ON-EVENT-REF Configured in RteEvent is not a Valid RUNNABLE-ENTITY.

	This error occurs, The Start-On-Event-Ref (Runnable Entity) referred in RteEvent Container is not a valid runnable path.
224	The RteEvent is referred more than once in RteEventToTaskMapping container. This error occurs, when the RteEvent is referred more than once in RteEventToTaskMapping Container
225	Direction given for the IPduPort is invalid. This error occurs, when the Direction is given for the IPduPort is IN and mapping is done for Rport or direction is given for the IPduPort is OUT and mapping is done for Pport.
226	Base type of the configured record element <type_path> is not allowed in E2E configuration. This error occurs, when base type of the configured record element is present in E2E configuration.
227	The VALUE of a MODE-DECLARATION and ON-TRANSITION-VALUE should be unique within a MODE-DECLARATION-GROUP. This error occurs, when the VALUE of a MODE-DECLARATION and ON-TRANSITION-VALUE are not unique within a MODE-DECLARATION-GROUP.
231	The Sw-Impl-Policy configured in variable data prototype of NV Data Interface is set to QUEUED. This error occurs, when the Sw-Impl-Policy configured in variable data prototype of NV Data Interface is set to QUEUED.
232	The ImplementationDataType whose category is UNION should have at least two ImplementationDataTypeElement as the sub element. This error occurs, when the Union Element is having less than two sub elements.
233	The ImplementationDataType whose category is STRUCTURE should have at least one ImplementationDataTypeElement as the sub element. This error occurs, when the structure Element is having less than one sub elements.
234	The reference path used for the mention parameter is invalid parameter. This error occurs, when the path of reference parameter is invalid.
235	Type attribute in PerInstanceMemory of 'C' Type of respective path and the Implementation data type shortname should not be same. This error occurs, when the Type attribute in PerInstanceMemory of 'C' Type of respective path and the Implementation data type shortname is same.
236	The Handle Out Of Range for Queue Send Com Spec should not be INVALIDE or DEFAULT or EXTERNAL-REPLACEMENT. This error occurs, when the Handle Out Of Range for Queue Send Com Spec is INVALID or DEFAULT or EXTERNAL-REPLACEMENT.
237	Identical TYPE attribute in Per Instance Memory of 'C' type is configured for the same TYPE DEFINTION of component. This error occurs, when Identical TYPE attribute in Per Instance Memory of 'C' type is configured for the same TYPE DEFINTION of component.
238	The attribute swImplPolicy of a dataElement referenced by a QueuedSenderComSpec must be set to the value queued.

	This error occurs, when the attribute swImplPolicy of a dataElement referenced by a QueuedSenderComSpec is not set to the value queued.
239	An initial value cannot be specified when the implementation policy is set to queued. This error occurs, when an initial value is specified when the implementation policy is set to queued.
240	RteReceiverUsedOsEventRef parameter should be configured when the RtelocInteractionReturnValue is set to 'RTE_COM' for the Task Path. This error occurs, when RteReceiverUsedOsEventRef parameter is not configured when the RtelocInteractionReturnValue is set to 'RTE_COM' for the Task Path.
241	The TimingEvent and Background Event can not be mapped into OsTask which other RteEvent has been allocated. This error occurs, when the RTE/BSW event is mapped to the ostask path which is same as the task mapped to Timing/Background event or when the Timing event and Background event are mapped to same ostask path.
242	RTE does not support multiple instances for the component and partition. This error occurs, when the multiple instances for the component along with partition is configured.
243	MAX-DELTA-COUNTER-INIT should be configured for the EndToEndProfile <Path> This error occurs, when the MAX-DELTA-COUNTER-INIT is not configured for EndToEndProfile
244	The value for DATA-ID should be in the range of 0-65535/0-255 when category is PROFILE_01/PROFILE_02 for the EndToEndProfile <Path> This error occurs, when the value for DATA-ID is not in the range of 0-65535/0-255 when category is PROFILE_01/ PROFILE_02 respectively for the EndToEndProfile
245	The value for DATA-LENGTH should be in the range of 0-65535/0-240 when Category is PROFILE_02/PROFILE_01 respectively for the EndToEndProfile <Path>. This error occurs, when the value for DATA-LENGTH is not in the range of 0-65535/0-240 when category is PROFILE_01/ PROFILE_02 respectively for the EndToEndProfile
246	The value for CRC-OFFSET should be in the range of 0-65535 when category is PROFILE_01 for the EndToEndProfile <Path>. This error occurs, when the value for CRC-OFFSET is not in the range of 0-65535 when category is PROFILE_01 for the EndToEndProfile.
247	The value for DATA-ID-MODE should be in the range of 0-2 when category is PROFILE_01 for the EndToEndProfile <Path>. This error occurs, when the value for DATA-ID-MODE is not in the range of 0-2 when category is PROFILE_01 for the EndToEndProfile.
248	The value for COUNTER-OFFSET should be in the range of 0-65535 when category is PROFILE_01 for the EndToEndProfile <Path>. This error occurs, when the value for COUNTER-OFFSET is not in the range of 0-65535 when category is PROFILE_01 for the EndToEndProfile.
249	The value for MAX-DELTA-COUNTER-INIT should be in the range of 0-14/ 0-15, when category is PROFILE_01/PROFILE_02 respectively for the EndToEndProfile <Path>.

	This error occurs, when the value for MAX-DELTA-COUNTER-INIT is not in the range of 0-14/ 0-15 when category is PROFILE_01/ PROFILE_02 respectively for the EndToEndProfile
250	There should be exact 1/16 DATA-IDs when the category is PROFILE_01/ PROFILE_02 respectively for the EndToEndProfile <Path>. This error occurs, when the DATA-IDs are not exactly 1 for PROFILE_01 or 16 for PROFILE_02 for the EndToEndProfile
251	The Mandatory Parameter RtelocInteractionReturnValue is not configured for <ECU Generation Path> This error occurs, when the RtelocInteractionReturnValue is not configured for given Generation path.
252	BSW-MODE-SWITCHED-ACK-EVENT should be configured when TIMEOUT value is greater than zero For the PROVIDED-MODE-GROUP <Path>. This error occurs, when the BSW-MODE-SWITCHED-ACK-EVENT is not configured when TIMEOUT value is greater than zero.
253	BSW-MODE-SWITCHED-ACK-EVENT should be mapped to an Extended Task when TIMEOUT value is greater than zero. This error occurs, when the BSW-MODE-SWITCHED-ACK-EVENT is not mapped to an Extended Task when TIMEOUT value is greater than zero.
254	All the BSW-MODE-SWITCH-EVENT<Event Paths> of the REQUIRED-MODE-GROUP connected to the specific PROVIDED-MODE-GROUP should be mapped to the same task <Task Path>. This error occurs, when All the BSW-MODE-SWITCH-EVENT of the REQUIRED-MODE-GROUP connected to a specific PROVIDED-MODE-GROUP are not mapped to the same task.
255	Task mapped to BSW-MODE-SWITCHED-ACK-EVENT <Event Path> should have higher priority than the Task priority of BSW-MODE-SWITCH-EVENT <Event Path> when TIMEOUT is greater than zero. This error occurs, when the task mapped to BSW-MODE-SWITCHED-ACK-EVENT has less priority than task priority of BSW- MODE-SWITCH-EVENT when TIMEOUT is greater than zero.
258	Position in Task is not configured for Event <Event Path> and Task<Task Path>. This error occurs, when PositionInTask is not configured for the mentioned event and task path.
259	Com notification is not configured for <Isignaltoipdu mapping path> since com notification is mandatory for configuration that has one system signal shared by many isignals for receiving System signal <system signal path>. This error occurs, when ComNotifictation is not configured when the configuration has one system signal shared by many isignals for the receiving System signal.
261	Timeout value should be zero for the AsynchronousServerCallPoint as WaitPoint is not configured.. This error occurs, when waitpoint is not configured if timeout is configured for client server asynchronous communication.
262	RteWaitOsEventRef in RteEventToTaskMapping should be configured for the AsynchronousServerCallReturnsEvent as WaitPoint is configured.

	This error occurs, when WaitOsEveRef is not configured if timeout configured is greater than zero for client server asynchronous communication,
263	The interfaces <p_iface> configured for P-PORT-PROTOTYPE <p_port_path> and <r_iface> configured for R-PORT-PROTOTYPE<r_port_path>are not compatible as the SHORT-NAMES for 'VARIABLE-DATA-PROTOTYPE' configured for both the interfaces are not same. This error occurs, when interfaces <p_iface> configured for P-PORT-PROTOTYPE <p_port_path> and <r_iface> configured for R-PORT-PROTOTYPE<r_port_path>are not compatible as the SHORT-NAMES for 'VARIABLE-DATA-PROTOTYPE' configured for both the interfaces are not same.
264	AsynchronousServerCallResultPoint not configured for the AsynchronousServerCallPoint. This error occurs, when the Result point is not configured for the Call api for the mentioned call point.
265	Result Point is not triggerred by any event.
266	Result Point and Call point are configured in the same runnable entity. However the the call point and corresponding result point should always be configured in different runnable entities.
300	In synchronous mode switching, Mode Switched Event's runnable which is called in OsTask should be mapped to one OsTask at least. This error occurs, when the runnable for Mode Switch Event in synchronous mode switching, is not mapped into any OsTask.
301	The task which Mode Manager with RteEvent has been mapped into should .have lower task priority than the task for Mode User with ModeSwitchEvent. This error occurs, when OsTask which the mode manager is allocated in, has the higher priority than OsTask for Mode User.
302	The Software Component which has the R-Ports connected to the specific P-Port, is not allocated in any partition. The Software Component name regarding R-Ports <Port Name> must be located in one of partition be configured. This error occurs, when Software Component with one of port (between PPort and RPort) is not allocated in any partition.
304	Eventhough ModeSwitchAck Event is configured in <SWC>, there is no configuration for Mode Switched Ack Request in the Com Spec regarding this Port <Port Path> This error occurs, when the ModeSwitchAck Event is configured in SWC without configuration regarding Mode Switched Ack Request in Com Spec in related Port.
305	The Configuration for Exclusive Area Implementation for <Exclusive Area>, should be done in Rte configuration This error occurs, when the exclusive area implementation is not done during Rte configuration, even though this is referenced by runnable to permit this runnable be running in Exclusive area.
306	In AUTOSAR 4.0.3 standard, the BSW modules distribution on multi partitions is not supported except for the EcuM module. Therefore the BswModuleExecution parameter in EcuC, must be enabled for only one partition. This error occurs, when the BswModuleExecution for each partition, is enabled in over than one partition.

307	The disabledMode Configuration for “Event Name” will not be working properly. Because mode switch point for this mode instance, is not configured in any runnable. This error occurs, when the DisabledMode for specific Rte Event has been configured without any mode switch point configuration for this mode instance.
308	When the synchronized activate offset among OsAlarms based on one specific OsCounter is enabed, the MaxAllowedCounterValue for this OsCounter, should be set with proper value This error occurs, when the MaxAllowedCounterValue is not configured or is not set with proper value.
309	In Enhanced Mode, ImplementationDataType which mapped to ModeDeclarationGroup should have uint8 or uint16 type. This error occurs, when ImplementationDataType which mapped ModeDeclarationGroup does not have uint8 or uint16 type while enhanced mode set to be true.
310	OsAlarm in ActivaionOsAlarmRef must not be duplicated, 〈ActivaionOsAlarmRef〉 is referenced more than twice in UsedOsActivaion. This error occurs, when referenced OsAlarm in ActivationOsAlarmRef in RteOsInteraction is duplicated.
311	When the Synchronized Activate Offset among OsAlarms based on one specific OsCounter is enabled the activation order (start alarm) should be defined amongs each alarm to give the value in ExpectedActivationPosition. 〈RteUsedOsActivaion〉 This error occurs, when OsAlarm with Synchronized Activate Offset, is configured and the order for setting the alarm among OsAlarms is not defined via ExpectedActivationPosition.
312	When the Synchronized Activate Offset among OsAlarms based on one specific OsCounter is enabled (Setting True) via SynchronizedActivateOffset configuration, the referenced OsCounter which OsAlarm will be synchronized, must be defined vai SynchronizedOsCounterRef 〈OsInteract Path〉 〈File Name〉 This error occurs, when SynchronizedActivateOffset is set as true without any Counter which should be referenced via SynchronizedOsCounterRef in OsInteract.
313	The Background Event can not be mapped to the OsTask which is same as the task mapped to TimingEvent. This error occurs, when both background event and timing event are mapped into same OsTask at the sametime
314	Either RteUsedOsAlarmRef or RteUsedOsScheduleTableExpiry-PointRef should be configured in all the RteEventToTaskMappings which are referring to the same OsTask. This error occurs, when to trigger the TimingEvent, either OsAlarm or SchedulePointExpiryPoint has been configured and if trigger source for whole of Timing Events are not same.
315	The ProvidedModeGroup path configured in ModeScheduleTableMap for BSW module is not a ProvidedModeGroup. This error occurs, when the port type in PortRef in ModeScheduleTableMap is not Provided
316	PortPath should be configured for ModeAccessPoints within MODE-ACCESS-POINT. This error occurs, when Port has not been referenced in ModeAccessPoints.
401	RteMappedToTaskRef of EventToTaskMapping for OperationInvokedEvent should be configured if CanBeInvokedConcurrently is false.

	This error occurs, when OsTask is not mapped to OperationInvokedEvent which starts a server runnable with false value on CanBeInvokedConcurrently parameter.
403	When timeout, inter-partition client-server communication, minimum start interval, or etc is used, RteWaitOsEventRef in RteEventToTaskMapping for the client runnable should be configured for the SynchronousServerCallPoint. Configure RteWaitOsEventRef in RteEventToTaskMapping for the client runnable.
404	If timeout is configured for SynchronousServerCallPoint, RteWaitOsAlarmRef in RteEventToTaskMapping for the client runnable should be configured for the SynchronousServerCallPoint. Assign a OsAlarm to RteWaitOsAlarmRef in RteEventToTaskMapping which invokes the client runnable.
405	When timeout for SynchronousServerCallPoint is used, OsAlarm referenced by RteWaitOsAlarmRef should have OsAlarmSetEvent of OsAlarmAction. Choose the OsAlarmSetEvent for OsAlarmAction and set the OsEventRef.
406	ValueTypeTRef of PortDefinedArgumentValue does not exist. Configure ValueTypeTRef in PortDefinedArgumentValue.
407	PortDefinedArgumentValue in PortApiOption does not have any value. Fill the Value of PortDefinedArgumentValue.
408	The ValueTypeTRef of PortDefinedArgumentValue is incorrect. Configure the correct value for ValueTypeTRef of PortDefinedArgumentValue. .For example, if the referenced path by ValueTypeTRef is ImplementationDataType, .then check whether the ImplementationDataType exists,.and if ApplicationDataType, then check whether the mapping to ImplementationDataType is correct.
409	ImplementationDataType referred by an ArgumentDataPrototype in Operation of ClientServerInterface does not exist. Check the configuration of ImplementationDataType referred by the ArgumentDataPrototype.
410	If ImplementationDataType is equal to a name of one of the Platform or Standard Types predefined in AUTOSAR code, nativeDeclaration in the SwBaseType shall not be configured or be same as base types of Platform or Standard Types. Change the ShortName of ImplementationDataType different from ShortNames of Platform or Standard Types, or make nativeDeclaration in SwBaseType be same as them of Platform or Standard Types.
411	BaseTypeRef ImplementationDataTypeRef needs to be configured for ImplementationDataTypes. Configure BaseTypeRef or ImplementationDataTypeRef in ImplementationDataType
413	ClientServerOperation(or ClientServerInterface) is duplicated. Remove duplicated ClientServerOperation or ClientServerInterface.
414	ArgumentDataPrototype of ClientServerOperaion is duplicated. Remove duplicated ArgumentDataPrototype or change ShortName.
415	For each asynchronous invocation of an operation prototype only one AsynchronousServerCallReturnsEvent shall be passed to the client component by the RTE. The

	AsynchronousServerCallReturnsEvent shall indicate either that the transmission was successful or that the transmission was not successful. Remove the other AsynchronousServerCallReturnsEvents except for one.
418	RteMappedToTaskRef for AsynchronousServerCallReturnsEvent is not configured If AsynchronousServerCallReturnsEvent is used for activation of a RunnableEntity, RteMappedToTaskRef of RteEventToTaskMapping for the AsynchronousServerCallReturnsEvent should be configured.
419	There is no Implementation Data Type or .Application Data Type which is mapped to ImplementationDataType for an ArgumentDataPrototype of the Operation in ClientServerInterface. Check the configuration of ImplementationDataType referred by an ArgumentDataPrototype of the Operation.
421	If the SynchronousServerCallPoint is configured on a RunnableEntity, .then the OsTaskSchedule of the OsTask mapped to the RunnableEntity should be FULL. Change the value of the OsTaskSchedule of an OsTask to 'FULL'.
422	The OsTaskPriority value of the OsTask mapped to the RunnableEntity where SynchronousServerCallPoint (client) is configured should be greater than the OsTaskPriority value of the OsTask mapped to OperationInvokedEvent (server). Adjust OsTaskPriority values of OsTasks to meet the priority condition of OsTasks (client > server).
423	Category of EndToEndProfile should be configured as either PROFILE_01 or PROFILE_02 Configure Category correctly (PROFILE_01/PROFILE_02)
424	The OsTaskPriority value of the OsTask mapped to the RunnableEntity where AsynchronousServerCallPoint (client) is configured should be greater than the OsTaskPriority value of the OsTask mapped to AsynchronousServerCallResultPoint (result). Adjust OsTaskPriority values of OsTasks to meet the priority condition of OsTasks (client > result).
425	The OsTaskPriority value of the OsTask mapped to the RunnableEntity where AsynchronousServerCallResultPoint (result) is configured should be greater than the OsTaskPriority value of the OsTask mapped to OperationInvokedEvent (server). Adjust OsTaskPriority values of OsTasks to meet the priority condition of OsTasks (result > server).
426	When OsEvent is used, The OsTask should have the OsEvent. Add the OsEvent to the OsEventRef of the OsTask.
427	RteEndToEndProtectionWrapperMode in RteGeneration shall be confiured as either SINGLE or REDUNDANT. Configure RteEndtoEndProectionWrapperMode correctly (SINGLE/REDUNDANT).
428	The ClientServerInterface referenced by RequiredInterfaceTRef of RPortPrototype does not include the ClientServerOperation refereced by OperationIRef of SynchronousServerCallPoint. Select the ClientServerOperation for OperationIRef of SynchronousServerCallPoint, which should be included the ClientServerInterface referenced by RequiredInterfaceTRef of

	RPortPrototype.
429	The RteUsedOsEventRef in RteEventToTaskMapping should be configured if the RunnableEntity has AsynchronousServerCallResultPoint (result) and the corresponding WaitPoint. Set the OsEvent to RteUsedOsEventRef in RteEventToTaskMapping for RunnableEntity which has AsynchronousServerCallResultPoint.
450	Context RPort is not configured for ExternalTriggerOccurredEvent. Context RPort should be configured for ExternalTriggerOccurredEvent
451	Target Trigger is not configured for ExternalTriggerOccurredEvent. Target Trigger should be configured for ExternalTriggerOccurredEvent.
452	TriggerInterface for this P-port is not configured. Valid TriggerInterface shall be configured for this P-port.
453	P-port specified by a TriggerInterface is connected to an R-port with an incompatible interface and no TriggerInterfaceMapping for this pair of interfaces is associated with the connection. TriggerInterfaces of connected P/R-port should be compatible and TriggerInterfaceMapping should be associated with this connection.
454	The same Trigger in a Trigger Sink should not be connected to multiple Trigger Sources. (N:1 communication is not supported.) Do not connect a ReleasedTrigger to multiple BswTriggerConnections.
455	The same Trigger in a Trigger Sink should not be connected to multiple Trigger Sources. (N:1 communication is not supported.) Do not connect a Trigger Sink to multiple Trigger Sources.
456	If RunnableEntity has MinimumStartInterval or QueueLentgh is configured, RunnableEntity should be mapped to at least one OsTask. RunnableEntity should be mapped to at least one OsTask.
457	If RunnableEntity has not MinimumStartInterval and QueueLentgh is not configured, RunnableEntity should not be mapped to any OsTask. RunnableEntity should not be mapped to any OsTask.
458	Rte Generator does not support Queued External Trigger in case of Inter Partition Communication. Do not configure Queued External Trigger in case of Inter Partition Communication.
459	A BswTriggerDirectImplementation is specified and an ExecutableEntity that is activated by an ExternalTriggerOccurredEvent associated to a connected Trigger of the TriggerSource is mapped to an OS task different from the one defined by the task attribute of the BswTriggerDirectImplementation. Map ExternalTriggerOccurredEvent of EventToTaskMapping to Task attribute of the BswTriggerDirectImplementation
460	A SynchronizedTrigger should not be referenced by more than one type of access method, where the type is one of the followings: ExternalTriggeringPoint, IssuedTrigger and BswTriggerDirectImplementation.

	Please select one of the followings. 1) Do not use BswTrigger for the IssuedTrigger which is defined in SynchronizedTrigger. 2) Do not use SwcTrigger for the ExternalTriggeringPoint which is defined in SynchronizedTrigger. 3) Do not connect BswTrigger and SwcTrigger for SynchronizedTrigger.
461	A SynchronizedTrigger should not be referenced by more than one type of access method, where the type is one of the followings: ExternalTriggeringPoint, IssuedTrigger and BswTriggerDirectImplementation. Please select one of the followings. 1) Do not use BswTrigger for the IssuedTrigger which is defined in SynchronizedTrigger. 2) Do not use SwcTrigger for the ExternalTriggeringPoint which is defined in SynchronizedTrigger. 3) Do not connect BswTrigger and SwcTrigger for SynchronizedTrigger.
462	A SynchronizedTrigger should not be referenced by more than one type of access method, where the type is one of the followings: ExternalTriggeringPoint, IssuedTrigger and BswTriggerDirectImplementation. Please select one of the followings. 1) Do not use BswTrigger for the IssuedTrigger which is defined in SynchronizedTrigger. 2) Do not use BswTrigger for the BswTriggerDirectImplementation which is defined in SynchronizedTrigger.
463	A Trigger of ReleasedTrigger shall not be referenced by both a IssuedTrigger and a BswTriggerDirectImplementation. Do not use same Trigger of ReleasedTrigger in both IssuedTrigger and BswTriggerDirectImplementation.
464	'SwImplPolicy' shall be configured to 'QUEUED' in case of 'RteBswTriggerSourceQueueLength' > 0. To use Queued Bsw External Trigger, please configure 'SwImplPolicy' to 'QUEUED'.
465	'SwImplPolicy' shall not be configured to 'QUEUED' in case of 'RteBswTriggerSourceQueueLength' = 0. To use Non Queued Bsw External Trigger, please do not configure 'SwImplPolicy' to 'QUEUED'.
466	'SwImplPolicy' shall be configured to 'QUEUED' in case of 'RteBswTriggerSourceQueueLength' > 0. To use Queued Bsw Internal Trigger, please configure 'SwImplPolicy' to 'QUEUED'.
467	'SwImplPolicy' shall not be configured to 'QUEUED' in case of 'RteBswTriggerSourceQueueLength' = 0. To use Non Queued Bsw Internal Trigger, please do not configure 'SwImplPolicy' to 'QUEUED'.
468	'SwImplPolicy' shall be configured to 'QUEUED' in case of 'RteTriggerSourceQueueLength' > 0. To use Queued External Trigger, please configure 'SwImplPolicy' to 'QUEUED'.
469	'SwImplPolicy' shall not be configured to 'QUEUED' in case of 'RteTriggerSourceQueueLength' = 0. To use Non Queued External Trigger, please do not configure 'SwImplPolicy' to 'QUEUED'.
470	'SwImplPolicy' shall be configured to 'QUEUED' in case of 'RteTriggerSourceQueueLength' > 0.

	To use Queued Internal Trigger, please configure 'SwImplPolicy' to 'QUEUED'.
471	'SwImplPolicy' shall not be configured to 'QUEUED' in case of 'RteTriggerSourceQueueLength' = 0. To use Non Queued Internal Trigger, please do not configure 'SwImplPolicy' to 'QUEUED'.
472	There is no valid BswExternalTriggerEvent. Please configure BswExternalTriggerEvent.
473	There is no RteBswEventToTaskMapping for BswExternalTriggerEvent. Please configure RteBswEventToTaskMapping for BswExternalTriggerEvent.
474	There is no valid BswInternalTriggerEvent. Please configure BswInternalTriggerEvent.
475	There is no RteBswEventToTaskMapping for BswInternalTriggerEvent. Please configure RteBswEventToTaskMapping for BswInternalTriggerEvent.
476	There is no valid RteExternalTriggerEvent. Please configure RteExternalTriggerEvent.
477	There is no RteEventToTaskMapping for RteExternalTriggerEvent. Please configure RteEventToTaskMapping for RteExternalTriggerEvent.
478	There is no valid RteInternalTriggerEvent. Please configure RteInternalTriggerEvent.
479	There is no RteEventToTaskMapping for RteInternalTriggerEvent. Please configure RteEventToTaskMapping for RteInternalTriggerEvent.
501	There is a EcucPartition which has no ShortName
502	EcucPartitionBswModuleExecution shall be set.
503	PartitionCanBeRestarted shall be set.
504	Software component path in the EcucPartitionSoftwareComponentInstanceRef is not valid.
505	ShortName of RteBswModuleInstance is empty.
506	BswImplRef of RteBswModuleInstance is empty.
507	BswModuleConfigRef of RteBswModuleInstance is empty.
508	There is an OsAppAlarmRef which is double referenced.
509	There is an OsAppCounterRef which is double referenced.
510	There is an OsAppIsrRef which is double referenced.

511	There is an OsAppTaskRef which is double referenced.
512	OsAppEcucPartRefOsAppEcucPartRef of OsApplication is invalid.
513	There is a mismatching between partition of SwComponentInstance and task of OsApplication.
514	There is a mismatching between partition of SwComponentInstance and alarm of OsApplication.
515	DataReceivedEvent, DataReceiveErrorEvent, DataSendCompletedEvent, DataWriteCompletedEvent shall be mapped to OsTask.
516	There is a mismatching OsSchedulePoint configuration of events which reference same runnable entity.
517	RteMappedToTaskRef is not set but OsSchedulePoint is set.
518	OsSchedulePoint shall be set as None if OsTask is full preemptive.
519	There is a mismatching OsSchedulePoint configuration of events which reference same runnable entity.
520	RteBswMappedToTaskRef is not set but OsSchedulePoint is set.
521	OsSchedulePoint shall be set as None if OsTask is full preemptive.
522	SwComponent Event shall not be mapped to OsTask which is used by Bsw Module.
523	RteExclusiveAreaImplMechanism is COOPERATIVE_RUNNABLE_PLACEMENT. But RteBswExclusiveAreaOsResourceRef is not set.
524	RteExclusiveAreaImplMechanism is OS_RESOURCE. But RteBswExclusiveAreaOsResourceRef is not set.
525	RteExclusiveAreaImplMechanism is COOPERATIVE_RUNNABLE_PLACEMENT. But RteExclusiveAreaOsResourceRef is not set.
526	RteExclusiveAreaImplMechanism is OS_RESOURCE. But RteExclusiveAreaOsResourceRef is not set.
527	System Signal is not mapped to ISignal.
528	ISignal is not mapped to ISignalToIPduMapping.
529	ISignalToIPduMapping is not mapped to ComSignal.
530	The RteMappedToTaskRef of RteEventToTaskMapping has invalid reference.
531	VariableAccessRef of RteVariableReadAccessRef is invalid.
532	VariableAccessRef of RteVariableWriteAccessRef is invalid.
533	Coherency Group shall be mapped to same ostask.
534	The RteMappedToTaskRef of RteEventToTaskMapping has invalid reference.

535	Format of OsSecPerTick is not valid.
536	Format of OsSecPerHwTick is not valid.
537	Format of TaskPrefix is not valid.
540	StartOnEvent cannot reference runnable of other SwComponent or InternalBehavior.
541	ParameterSwComponentType shall not have InternalBehavior.
542	ParameterSwComponentType shall not have require port.
543	SupportMultipleInstance is not set. SwComponent cannot created more than once.
544	ParameterSwComponentType cannot created more than once.
545	BswBehaviorRef of SwcBswMapping is empty.
546	SwcBehaviorRef of SwcBswMapping is empty.
547	SwcBswMapping is only valid when software component type is ServiceSwComponentType or EcuAbstractionSwComponentType or ComplexDeviceDriverSwComponentType.
548	CompositionSwComponentType shall not be multiple instantiated.
549	There is a infinite loop of hierarchy.
550	ParameterSwComponentType does not support inter-partition communication.
551	There is no SwComponentType for the name.
552	There is no PortPrototype for the path.
553	There is no interface for the path.
554	There is no VariableDataPrototype for the path.
555	There is an error for determining queued or not.
556	There is no SwComponentPrototype for the path.
557	There is an error for getting Rte_Send Function Name.
558	There is an error for getting Rte_Receive Function Name.
559	InitValue is invalid.

560	The NvmWriteRamBlockToNvm and NvmReadRamBlockFromNvm parameters set to the Rte_GetMirror and Rte_SetMirror API of the NvBlockDescriptor when NvmBlockUseSyncMechanism is enabled
561	InvalidValue of ApplicationDataType is invalid.
562	InvalidValue of ImplementationDataType is invalid.
563	PortPrototype path is invalid.
564	TypeTRef of ApplicationArrayElement is invalid.
565	TypeTRef of ApplicationRecordElement is invalid.
566	TypeTRef for InitValue is invalid.
571	RootSwCompositionPrototype of System is empty.
572	SoftwareCompositionTRef of RootSwCompositionPrototype is empty.
575	There is no PortPrototype for the path.
581	There is no RteEvent for the RunnableEntity.
582	There is no RteEventToTaskMapping for the RteEvent which uses implicit sender receiver communication.
583	There is no RteMappedtoTaskRef for the RteEvent which uses implicit sender receiver communication.
584	Different execution instances of a runnable entity, which use implicit data access, shall not be mapped to different Preemption Areas.
585	ImplementationDataType Path is invalid.
586	ApplicationDataType Path is invalid.
591	OsEvent has been used for more than one RTE Event.
592	The format of RTE Generator Version is not correct.
593	There is no ISignalIPdu for the path.
594	ISignal and DataElement of PortPrototype shall have same HandleOutOfRange configuration.
595	OsCounter has no value of OsSecondsPerTick. You can also set OsSecondsPerTick for Counter by using -OsSecPerTick option.
596	OsCounterMaxAllowedValue shall be greater than or equal to OsAlarm's Offset.
597	OsCounterMaxAllowedValue shall be greater than or equal to OsAlarm's Period.

598	OsCounterMaxAllowedValue shall be greater than or equal to OsScheduleTable's Offset.
605	TypeTRef in SwComponentPrototype is empty or invalid. TypeTRef in SwComponentPrototype shall be set with valid SwComponentType Path.
606	TypeTRef in VariableDataPrototype is empty or invalid. TypeTRef in VariableDataPrototype shall be set with valid DataType Path.
607	TypeTRef in ParameterDataPrototype is empty or invalid. TypeTRef in ParameterDataPrototype shall be set with valid DataType Path.
612	TypeTRef in ModeGroup is empty or invalid. TypeTRef in ModeGroup shall be set with valid ModeDeclarationGroup Path.
613	TypeTRef in ModeDeclarationGroupPrototype is empty or invalid. TypeTRef in ModeDeclarationGroupPrototype shall be set with valid ModeDeclarationGroup Path.
614	ProvidedInterfaceTRef/RequiredInterfaceTRef in PPortPrototype/RPortPrototype is empty or invalid. ProvidedInterfaceTRef/RequiredInterfaceTRef in PPortPrototype/RPortPrototype shall be set with valid Interface Path.
615	Category in ApplicationValueSpec/AutosarDataType/CompuMethod/EndToEndProfile/ModeDeclarationGroup is empty or invalid. Category in ApplicationValueSpec/AutosarDataType/CompuMethod/EndToEndProfile/ModeDeclarationGroup shall be set with valid value.
616	The number of SwcInternalBehavior in SwComponentType is not 1. The number of SwcInternalBehavior in SwComponentType shall be 1.
619	RteMappedToTaskRef in RteEventToTaskMapping is empty or invalid. RteMappedToTaskRef in RteEventToTaskMapping shall be set with valid OsTask Path. ※See 8.7 Guide for RteEvent
620	ContextComponentRef in PPortInCompositionInstanceRef/ProviderIRef/RequesterIRef/RPortInCompositionInstanceRef is empty or invalid. ContextComponentRef in PPortInCompositionInstanceRef/ProviderIRef/RequesterIRef/RPortInCompositionInstanceRef shall be set with valid SwComponentPrototype path.
621	LowerLimit in CompuScale/InternalConstrs/PhysConstrs is empty. LowerLimit in CompuScale/InternalConstrs/PhysConstrs shall be set.
622	UpperLimit in CompuScale/InternalConstrs/PhysConstrs is empty. UpperLimit in CompuScale/InternalConstrs/PhysConstrs shall be set.
624	Even though IntervalType is not INFINITE, Value in Limit is empty.

	If IntervalType is not INFINITE, Value in Limit shall be set.
625	InternalConstr in DataConstrRule is empty. InternalConstr in DataConstrRule shall be set.
627	The number of DataConstrRule in DataConstr is not 1. The number of DataConstrRule in DataConstr shall be 1.
628	Even though DataType is used for HandleOutOfRange, DataConstrRef in SwDataDefProps is empty. If DataType is used for HandleOutOfRange, DataConstrRef in SwDataDefProps shall be set.
629	SwDataDefProps in AutosarDataType is empty, even though DataType is used for HandleOutOfRange. SwDataDefProps in AutosarDataType shall be set, if DataType is used for HandleOutOfRange.
630	SwDataDefProps in ArgumentDataPrototype/AutosarDataType/SwPointerTargetProps/VariableDataPrototype is empty. SwDataDefProps in ArgumentDataPrototype/AutosarDataType/SwPointerTargetProps/VariableDataPrototype shall be set.
631	RteGeneration in RTE ECU Configuration is empty. RteGeneration in RTE ECU Configuration shall be set.
632	ComIPduDirection in ComIPdu is empty or invalid. ComIPduDirection in ComIPdu shall be set with valid value.
633	ImplementedEntryRef in BswEntity is empty or invalid. ImplementedEntryRef in BswEntity shall be set with valid BswModuleEntry path.
636	RteInitializationBehavior in RTE ECU Configuration is empty. RteInitializationBehavior in RTE ECU Configuration shall be set.
637	EcucPartition of BswModuleDescription/SwComponentPrototype/OsTask is empty or invalid. EcucPartition of BswModuleDescription/SwComponentPrototype/OsTask shall be set.
638	RteTaskComMapping in RteGeneration is empty or invalid. RteTaskComMapping in RteGeneration shall be set with valid OsTask path.
639	Symbol in CompuScale/RunnableEntity/SchedulerNamePrefix is empty or invalid. Symbol in CompuScale/RunnableEntity/SchedulerNamePrefix shall be set with valid value.
640	OsApplication of EcucPartition/OsAlarm/OsTask/SwComponentPrototype is empty or invalid. OsApplication of EcucPartition/OsAlarm/OsTask/SwComponentPrototype shall be set.
641	BswBehaviorRef in SwcBswMapping is empty or invalid. BswBehaviorRef in SwcBswMapping shall be set with valid BswInternalBehavior path.
642	BswEntityRef in SwcBswRunnableMapping is empty or invalid.

	BswEntityRef in SwcBswRunnableMapping shall be set with valid BswEntity path.
643	StartsOnEventRef in BswEvent is empty or invalid. StartsOnEventRef in BswEvent shall be set with BswSchedulableEntity.
644	StartOnEventRef in RteEvent is empty or invalid. StartOnEventRef in RteEvent shall be set with valid RunnableEntity path.
645	ComSignalType in ComGroupSignal/ComSignal is empty or invalid. ComSignalType in ComGroupSignal/ComSignal shall be set with valid value.
646	ParameterRef in ParameterComSpec is empty or invalid. "ParameterRef in ParameterComSpec shall be set with valid ParameterDataPrototype path.
647	DataElementRef in SenderComSpec/ReceiverComSpec/InvalidationPolicy is empty or invalid. DataElementRef in SenderComSpec/ReceiverComSpec/InvalidationPolicy shall be set with valid VariableDataPrototype path.
648	There is no RteEventToTaskMapping for RteEvent. RteEventToTaskMapping for RteEvent shall be exist.
649	EventSourceRef in DataSendeCompletedEvent / DataWriteCompletedEvent / AsynchronousServerCallReturnsEvent / ModeSwitchedAckEvent / InternalTriggerOccurredEvent is empty or invalid. EventSourceRef DataSendeCompletedEvent / DataWriteCompletedEvent / AsynchronousServerCallReturnsEvent / ModeSwitchedAckEvent / InternalTriggerOccurredEvent shall be set with valid value.
650	Path of AutosarObject is duplicated. Path of AutosarObject shall be uniqueue.
651	Value in Limit / ModeDeclaration / NumValueSpec / TextValueSpec / V is empty or invalid. Value in Limit / ModeDeclaration / NumValueSpec / TextValueSpec / V shall be set with valid value.
652	PhysConstr in DataConstrRule is empty. PhysConstr in DataconstrRule shall be set.
653	ServiceId in BswModuleEntry is empty. ServiceId in BswModuleEntry shall be set.
654	ComIPduSignalProcessing in ComIPdu is empty. ComIPduSignalProcessing in ComIPdu shall be set.
655	Even though there is no EcucPartition or single EcucPartition, RteTaskComMapping in RteGeneration is set. If there is no EcucPartition or single EcucPartition, RteTaskComMapping in RteGeneration shall not be set.
656	SwBaseTypeRef in SwDataDefProps is empty or invalid. SwBaseTypeRef in SwDataDefProps shall be set with valid SwBaseType path.

657	ModuleId in BswModuleDescription and ModuleId in BswModuleDescription is same. ModuleId in BswModuleDescription and ModuleId in BswModuleDescription shall be different.
659	ServiceId in BswModuleEntry and ServiceId in BswModuleEntry is same. ServiceId in BswModuleEntry and ServiceId in BswModuleEntry shall be different.
660	TargetPPortRef in PPortInCompositionInstanceRef / ProviderIRef is empty or invalid. TargetPPortRef in PPortInCompositionInstanceRef / ProviderIRef shall be set with valid PPortPrototype path.
661	"TargetRPortRef in RPortInCompositionInstanceRef / RequesterIRef is empty or invalid. TargetRPortRef in RPortInCompositionInstanceRef / RequesterIRef shall be set with valid RPortPrototype path.
662	PPortPrototype and RPortPrototype are not compatible. PPortPrototype and RPortPrototype shall be compatible.
663	ContextPortRef in ModelRef is empty or invalid. ContextPortRef in ModelRef shall be set with valid PortPrototype path.
664	ProviderIRef in AssemblySwConnector is empty. ProviderIRef in AssemblySwConnector shall be set.
665	RequesterIRef in AssemblySwConnector is empty. RequesterIRef in AssemblySwConnector shall be set.
666	InnerPortIRef in AssemblySwConnector is empty. InnerPortIRef in DelegationSwConnector shall be set.
667	PPortInCompositionInstanceRef in InnerPortIRef is empty. PPortInCompositionInstanceRef in InnerPortIRef shall be set.
668	RPortInCompositionInstanceRef in InnerPortIRef is empty. RPortInCompositionInstanceRef in InnerPortIRef shall be set.
669	There is a M:N connection. RTE does not support M:N connections. M:N connections shall be removed.
670	ShortName in AutosarObject is duplicated. ShortName in AutosarObject shall be unique.
671	ShortName in AutosarObject is empty or invalid. ShortName in AutosarObject shall be set.
672	OperationIref in OperationInvokedEvent / ServerCallPoint is empty. OperationIref in OperationInvokedEvent / ServerCallPoint shall be set.
673	ContextPPortRef in ModeGroupIRef / OperationIref / PTriggerInAtomicSwcTypeInstanceRef is empty or invalid. ContextPPortRef in ModeGroupIRef / OperationIref / PTriggerInAtomicSwcTypeInstanceRef shall be set with valid PPortPrototype path.

674	TargetProvidedOperationRef in OperationIref is empty or invalid. TargetProvidedOperationRef in OperationIref shall be set with valid ClientServerOperation path.
675	ModeIRef in SwcModeSwitchEvent is empty. ModeIRef in SwcModeSwitchEvent shall be set.
676	ContextModeDeclarationGroupPrototypeRef in ModeIRef is empty or invalid. ContextModeDeclarationGroupPrototypeRef in ModeIRef shall be set with valid ModeGroup path.
677	TriggerIRef in ExternalTriggerOccurredEvent is empty. TriggerIRef in ExternalTriggerOccurredEvent shall be set.
678	ContextRPortRef in DataIRef/OperationIref/RTriggerInAtomicSwcInstanceRef is empty or invalid. ContextRPortRef in DataIRef/OperationIref/RTriggerInAtomicSwcInstanceRef shall be set with valid RPortPrototype path.
679	ModeGroup in ModeSwitchInterface is empty. ModeGroup in ModeSwitchInterface shall be set.
680	RteBswImplementationRef in RteBswModuleInstance is empty or invalid. RteBswImplementationRef in RteBswModuleInstance shall be set with valid BswImplementation path.
681	RteBswImplementationRef in RteBswModuleInstance is duplicated. RteBswImplementationRef in RteBswModuleInstance shall be unique.
682	RteBswModuleConfigurationRef in RteBswModuleInstance is empty or invalid. RteBswModuleConfigurationRef in RteBswModuleInstance shall be set with valid BswModuleDescription path.
683	RteSoftwareComponentInstanceRef in RteSwComponentInstance is empty or invalid. RteSoftwareComponentInstanceRef in RteSwComponentInstance shall be set with valid SwComponentPrototype path.
684	RteSoftwareComponentInstanceRef in RteSwComponentInstance andRteSoftwareComponentInstanceRef in RteSwComponentInstance are same. RteSoftwareComponentInstanceRef in RteSwComponentInstance andRteSoftwareComponentInstanceRef in RteSwComponentInstance shall not be same.
685	There is no EcucPartition in which BswModuleExecution is true. There shall be an EcucPartition in which BswModuleExecution is true.
686	The number of EcucValueCollection is not 1. The number of EcucValueCollection shall be 1.
687	The number of OsOs is not 1. The number of OsOs shall be 1.
688	RteBswModuleConfigurationRef in RteBswModuleInstance and RteBswModuleConfigurationRef in RteBswModuleInstance is same.

	RteBswModuleConfigurationRef in RteBswModuleInstance and RteBswModuleConfigurationRef in RteBswModuleInstance shall not be same.
689	The number of RteGeneration is not 1. The number of RteGeneration shall be 1.
690	The number of RteInitializationBehavior is not 1. The number of RteInitializationBehavior shall be 1.
691	ISignalRef in ISignalToIPduMapping and ISignalRef in ISignalToIPduMapping is same. ISignalRef in ISignalToIPduMapping and ISignalRef in ISignalToIPduMapping shall not be same.
692	ISignalGroupRef in ISignalToIPduMapping and ISignalGroupRef in ISignalToIPduMapping is same. ISignalGroupRef in ISignalToIPduMapping and ISignalGroupRef in ISignalToIPduMapping shall not be same.
693	TargetTriggerRef in PTriggerInAtomicSwcTypeInstanceRef/RTriggerInAtomicSwcInstanceRef is empty or invalid. TargetTriggerRef in PTriggerInAtomicSwcTypeInstanceRef/RTriggerInAtomicSwcInstanceRef shall be set with valid Trigger path.
694	SwComponentPrototype for the SwComponentType is not exist. SwComponentPrototype for the SwComponentType shall be exist.
695	BehaviorRef in SwcImplementationandBehaviorRef in SwcImplementationis Same. BehaviorRef in SwcImplementationandBehaviorRef in SwcImplementationshall not be Same.
697	SystemSignal is not mapped to ISignal. SystemSignal shall be mapped to ISignal.
698	ISignal is not mapped to ISignalToIPduMapping. ISignal shall be mapped to ISignalToIPduMapping.
699	SystemSignalGroup is not mapped to ISignalGroup. SystemSignalGroup shall be mapped to ISignalGroup.
700	ISignalGroup is not mapped to ISignalToIPduMapping. ISignalGroup shall be mapped to ISignalToIPduMapping.
701	There are different definitions of the ImplementationDataTypes which have a same short name. There shall be a same definitions of the ImplementationDataTypes which have a same short name.
702	ComIPdu for the ComSignal/ComSignalGroup is not exist. ComIPdu for the ComSignal/ComSignalGroup shall be exist.
703	ImplementationDataTypeElement in ImplementationDataType/ImplementationDataTypeElement is empty. ImplementationDataTypeElement in ImplementationDataType/ImplementationDataTypeElement shall be set.
704	BswModuleEntryRef in BswModuleEntryRefConditional is empty or invalid.

	BswModuleEntryRef in BswModuleEntryRefConditional shall be set with valid BswModuleEntry path.
705	Direction of ComSignal/ComSignalGroup/ISignalIPdu cannot be determined. Please check Direction of ComIPdu for ComSignal/ComSignalGroup/ISignalIPdu.
706	ImplementationDataTypeRef in DataTypeMap/SwDataDefProps is empty or invalid. ImplementationDataTypeRef in DataTypeMap/SwDataDefProps shall be set with valid ImplementationDataType path.
707	QueueLength in QueuedReceiverComSpec/ServerComSpec is empty or invalid. QueueLength in QueuedReceiverComSpec/ServerComSpec shall be set with valid value.
708	BswModuleDescriptionRef in BswModuleDescriptionRefConditional is empty or invalid. BswModuleDescriptionRef in BswModuleDescriptionRefConditional shall be set with valid BswModuleDescription path.
709	Even though ImplementationDataType/ImplementationDataTypeElement is Array, the number of ImplementationDataTypeElement is not 1. If ImplementationDataType/ImplementationDataTypeElement is Array, the number of ImplementationDataTypeElement shall be 1.
710	QueuedReceiverComSpec in RPortPrototype is empty. QueuedReceiverComSpec in RPortPrototype shall be set.
715	There is no ComSignal for ISignalToIPduMapping. There shall be a ComSignal for ISignalToIPduMapping.
716	There is no ComSignalGroup for ISignalToIPduMapping. There shall be a ComSignalGroup for ISignalToIPduMapping.
717	The number of V in SwValuesPhys is not 1. The number of V in SwValuesPhys shall be 1.
718	Ports of AssemblySwConnector is not compatible. Ports of AssemblySwConnector shall be compatible.
719	ImplementationDataType cannot be resolved for ApplicationDataType. ImplementationDataType shall be resolved for ApplicationDataType.
720	ParameterProvideComSpec in PPortPrototype is empty. ParameterProvideComSpec in PPortPrototype shall be set.
721	ParameterRequireComSpec in RPortPrototype is empty. ParameterRequireComSpec in RPortPrototype shall be set.
722	The number of RootVariableDataPrototypeRef in AutosarVariableInImplDataType is over 1. The number of RootVariableDataPrototypeRef in AutosarVariableInImplDataType shall be 0 or 1.
723	The number of TargetDataPrototypeRef in AutosarVariableInImplDataType is over 1. The number of TargetDataPrototypeRef in AutosarVariableInImplDataType shall be 0 or 1.

724	Even though BswModuleEntry is used as BswCalledEntity, CallType in BswModuleEntry is not REGULAR or CALLBACK. If BswModuleEntry is used as BswCalledEntity, CallType in BswModuleEntry shall be REGULAR or CALLBACK.
725	Even though BswModuleEntry is used as BswInterruptEntity, CallType in BswModuleEntry is not INTERRUPT. If BswModuleEntry is used as BswInterruptEntity, CallType in BswModuleEntry shall be INTERRUPT.
726	Even though BswModuleEntry is used as BswSchedulableEntity, CallType in BswModuleEntry is not SCHEDULED. If BswModuleEntry is used as BswSchedulableEntity, CallType in BswModuleEntry shall be SCHEDULED.
727	Even though BswModuleEntry is used as BswInterruptEntity, ExecutionContext in BswModuleEntry and InterruptCategory in BswInterruptEntity is not matched. If BswModuleEntry is used as BswInterruptEntity, ExecutionContext in BswModuleEntry and InterruptCategory in BswInterruptEntity shall be matched.
728	Even though BswModuleEntry is used as BswSchedulableEntity, ExecutionContext in BswModuleEntry is not TASK. If BswModuleEntry is used as BswSchedulableEntity, ExecutionContext in BswModuleEntry shall be TASK.
729	The number of Period in TimingEvent/BswTimingEvent/DataFilter is not 1. The number of Period in TimingEvent/BswTimingEvent/DataFilter shall be 1.
730	ImplInitValue in CalibrationParameterValue is invalid. ImplInitValue in CalibrationParameterValue shall be valid.
731	ApplInitValue in CalibrationParameterValue is invalid. ApplInitValue in CalibrationParameterValue shall be valid.
732	PossibleErrorRef in ClientServerOperation is invalid. PossibleErrorRef in ClientServerOperation shall be valid.
733	ISignalRef in ISignalToIPduMapping is invalid. ISignalRef in ISignalToIPduMapping shall be set with valid ISignal path.
734	ISignalGroupRef in ISignalToIPduMapping is invalid. ISignalGroupRef in ISignalToIPduMapping shall be set with valid ISignalGroup path.
735	Even though ComSignalType in ComSignal/ComGroupSignal is UINT8_DYN.DynamicLength in SystemSignal is not set true. If ComSignalType in ComSignal/ComGroupSignal is UINT8_DYN.DynamicLength in SystemSignal shall be set true.
736	Even though ComSignalType in ComSignal/ComGroupSignal is not UINT8_DYN.DynamicLength in SystemSignal is set true. If ComSignalType in ComSignal/ComGroupSignal is not UINT8_DYN.DynamicLength in

	SystemSignal shall not be set true.
737	Even though ComSignalType in ComSignal/ComGroupSignal is UINT8_DYN.ArraySizeSemantics in ImplementationDataType is not VARIABLE-SIZE. If ComSignalType in ComSignal/ComGroupSignal is UINT8_DYN.ArraySizeSemantics in ImplementationDataType shall be VARIABLE-SIZE.
738	Even though ComSignalType in ComSignal/ComGroupSignal is not UINT8_DYN.ArraySizeSemantics in ImplementationDataType is VARIABLE-SIZE. If ComSignalType in ComSignal/ComGroupSignal is not UINT8_DYN.ArraySizeSemantics in ImplementationDataType shall not be VARIABLE-SIZE.
739	ConstantRef in ConstantReference is empty or invalid. ConstantRef in ConstantReference shall be set with valid Constant path.
740	ComSignal/ComSignalGroup for TX is not mapped to PPortPrototype. ComSignal/ComSignalGroup for TX shall be mapped to PPortPrototype.
741	ComSignal/ComSignalGroup for RX is not mapped to RPortPrototype. ComSignal/ComSignalGroup for RX shall be mapped to RPortPrototype.
742	The number of DataConstrRule in DataConstr is not 1. The number of DataConstrRule in DataConstr shall be 1.
743	The number of AutosarVariableIRef in AccessedVariable/ReadNvData/WrittenNvData is not 1. The number of AutosarVariableIRef in AccessedVariable/ReadNvData/WrittenNvData shall be 1.
744	The number of AutosarParameterIRef in AccessedParameter is not 1. The number of AutosarParameterIRef in AccessedParameter shall be 1.
745	ComSpec in PortPrototype is empty. ComSpec in PortPrototype shall be set.
746	NonQueuedSenderComSpec in PPortPrototype is empty. NonQueuedSenderComSpec in PPortPrototype shall be set.
747	ComTimeout and TransmissionAcknowledgeTimeout is not matched. ComTimeout and TransmissionAcknowledgeTimeout shall be matched.
748	ConstantSpecification/RecordValueSpec is invalid. ConstantSpecification/RecordValueSpec shall be valid.
749	ComTimeout and AliveTimeout is not matched. ComTimeout and AliveTimeout shall be matched.
750	Even though there is a DataSendCompletedEvent, TransmissionAcknowledge is not set. If there is a DataSendCompletedEvent, TransmissionAcknowledge shall be set.
751	Even though there is a DataWriteCompletedEvent, TransmissionAcknowledge is not set. If there is a DataWriteCompletedEvent, TransmissionAcknowledge shall be set.

752	Even though there is a Rte_COMCbkTAck, TransmissionAcknowledge is not set. If there is a Rte_COMCbkTAck, TransmissionAcknowledge shall be set.
753	There is no OsApplication for EcucPartition. There shall be an OsApplication for EcucPartition.
754	There is multiple AccessedVariable/NvRamBlockElements which have same LocalVariableRef. There shall not be multiple AccessedVariable/NvRamBlockElements which have same LocalVariableRef.
755	ImplementationDataType/ImplementationDataTypeElement is invalid. ImplementationDataType/ImplementationDataTypeElement shall be valid.
756	ValueSpec in InitValue is invalid. ValueSpec in InitValue shall be valid.
757	ImplementationDataType/ImplementationDataTypeElement and ImplementationDataType/ImplementationDataTypeElement are not compatible. ImplementationDataType/ImplementationDataTypeElement and ImplementationDataType/ImplementationDataTypeElement shall be compatible.
758	TriggerInterface and TriggerInterface are not compatible. TriggerInterface and TriggerInterface shall be compatible.
759	DataConstr and DataConstr are not compatible. DataConstr and DataConstr shall be compatible.
760	SenderReceiverInterface and SenderReceiverInterface are not compatible. SenderReceiverInterface and SenderReceiverInterface shall be compatible.
761	NvDataInterface and NvDataInterface are not compatible. NvDataInterface and NvDataInterface shall be compatible.
762	Category in ApplicationDataType and Category in ImplementationDataType are not compatible. Category in ApplicationDataType and Category in ImplementationDataType shall be compatible.
763	TypeTRef in RamBlock and TypeTRef in RomBlock are not compatible. TypeTRef in RamBlock and TypeTRef in RomBlock shall be compatible.
764	DataType in VariableDataPrototype and DataType in VariableDataPrototype are not compatible. DataType in VariableDataPrototype and DataType in VariableDataPrototype shall be compatible.
765	PortPrototype has an incompatible ComSpec with Interface. ComSpec shall be compatible with Interface.
766	HandleInvalid in VariableDataPrototype and HandleInvalid in VariableDataPrototype are not compatible. HandleInvalid in VariableDataPrototype and HandleInvalid in VariableDataPrototype shall be compatible.
767	Even though SwComponentType supports multiple instantiation, EnableTakeAddress is used.

	If SwComponentType supports multiple instantiation, EnableTakeAddress shall not be used.
768	RTE currently does not support initial value of CalibrationParameterValue. Do not use initial value of CalibrationParameterValue.
769	There are multiple RoleBasedPortAssignment which have same role. There shall be single RoleBasedPortAssignment for each role.
770	The number of ModeMapping is over Multiplicity. The number of ModeMapping shall be within Multiplicity.
771	The number of NvBlockNeeds is over Multiplicity. The number of NvBlockNeeds shall be within Multiplicity.
772	The number of RamBlock is over Multiplicity. The number of RamBlock shall be within Multiplicity.
773	The number of RomBlock is over Multiplicity. The number of RomBlock shall be within Multiplicity.
774	The number of CalcRamBlockCrc is over Multiplicity. The number of CalcRamBlockCrc shall be within Multiplicity.
775	The number of RestoreAtStart is over Multiplicity. The number of RestoreAtStart shall be within Multiplicity.
776	The number of StoreAtShutDown is over Multiplicity. The number of StoreAtShutDown shall be within Multiplicity.
777	The number of WritingFrequency is over Multiplicity. The number of WritingFrequency shall be within Multiplicity.
778	The number of NvRamBlockElement is over Multiplicity. The number of NvRamBlockElement shall be within Multiplicity.
779	The number of ReadNvData is over Multiplicity. The number of ReadNvData shall be within Multiplicity.
780	The number of WrittenNvData is over Multiplicity. The number of WrittenNvData shall be within Multiplicity.
781	The number of VariableRef is over Multiplicity. The number of VariableRef shall be within Multiplicity.
782	The number of AutosarVariableInImplDataType is over Multiplicity. The number of AutosarVariableInImplDataType shall be within Multiplicity.
783	The number of OsResourceProperty is over Multiplicity. The number of OsResourceProperty shall be within Multiplicity.
784	The number of InitValue in ParameterComSpec is over Multiplicity.

	The number of InitValue in ParameterComSpec shall be within Multiplicity.
785	The number of TypeTRef is over Multiplicity. The number of TypeTRef shall be within Multiplicity.
786	The number of PortPrototypeRef in RoleBasedPortAssignment is over Multiplicity. The number of PortPrototypeRef in RoleBasedPortAssignment shall be within Multiplicity.
787	The number of Role is over Multiplicity. The number of Role shall be within Multiplicity.
788	The number of RteVariableReadAccessRef is over Multiplicity. The number of RteVariableReadAccessRef shall be within Multiplicity.
789	The number of RteVariableWriteAccessRef is over Multiplicity. The number of RteVariableWriteAccessRef shall be within Multiplicity.
790	The number of LocalParameterRef is over Multiplicity. The number of LocalParameterRef shall be within Multiplicity.
791	The number of BehaviorRef in SwcImplementation is over Multiplicity. The number of BehaviorRef in SwcImplementation shall be within Multiplicity.
792	The number of SwcBswMappingRef is over Multiplicity. The number of SwcBswMappingRef shall be within Multiplicity.
793	Parameter Interface does not support N:1 communication. Parameter Interface shall be only used for 1:1, 1:N communication.
794	RamBlock in NvBlockDescriptor is empty. RamBlock in NvBlockDescriptor shall be set.
795	QueueLength for PortPrototype and VariableDataPrototype is empty. QueueLength for PortPrototype and VariableDataPrototype shall be set.
796	RTE_COMCbk Function is not used with RPortPrototype. RTE_COMCbk Function shall be used with RPortPrototype.
797	InitValue in NonQueuedSenderComSpec is empty. InitValue in NonQueuedSenderComSpec shall be set.
798	There is no SenderRecArrayElementMapping in SenderRecArrayTypeMapping for the index . SenderRecArrayElementMapping in SenderRecArrayTypeMapping shall be exist for the index .
799	There is no SenderRecRecordElementMapping in SenderRecRecordTypeMapping for ImplementationDataTypeElement. SenderRecRecordElementMapping in SenderRecRecordTypeMapping shall be exist for ImplementationDataTypeElement.
800	Parameter Interface supports only Intra Partition Communication. Do not use Parameter Interface except Intra Partition Communication.

801	SwComponentPrototypes for the SwComponentType is not assigned to the same partition. SwComponentPrototypes for the SwComponentType shall be assigned to the same partition.
802	There is no SenderRecRecordElementMapping in SenderRecRecordTypeMapping for ApplicationRecordElement. SenderRecRecordElementMapping in SenderRecRecordTypeMapping shall be exist for ApplicationRecordElement.
803	The number of TypeTRef in RomBlock is over Multiplicity. The number of TypeTRef in RomBlock shall be within Multiplicity.
804	Even though multiple instance is used, SupportsMultipleInstantiation in InternalBehavior is not set true. If multiple instance is used, SupportsMultipleInstantiation in InternalBehavior shall be set true.
805	Both ISignalRef and ISignalGroupRef in ISignalToIPduMapping exist. Only one of ISignalRef or ISignalGroupRef in ISignalToIPduMapping shall exist.
806	Neither ISignalRef nor ISignalGroupRef in ISignalToIPduMapping exists. One of ISignalRef or ISignalGroupRef in ISignalToIPduMapping shall exist.
807	The number of NvBlockDataMapping in NvBlockDescriptor is not 1. The number of NvBlockDataMapping in NvBlockDescriptor shall be 1.
808	The number of internal OsResource in OsTask is over 1. The number of internal OsResource in OsTask shall be 1.
809	There is an inconsistency between initial values of DataElement and ComSignal/ComSignalGroup. Initial values of DataElement and ComSignal/ComSignalGroup shall be consistent.
810	Unconnected PortPrototype typed with NvDataInterface does not have NvRequireComSpec with a InitValue. Unconnected PortPrototype typed with NvDataInterface shall have NvRequireComSpec with a InitValue.
811	RTE does not support REPLACE HandleTimeoutType for the intra ecu communication. RTE does not support REPLACE HandleTimeoutType for the intra ecu communication.
812	HandleTimeoutType and ComRxDataTimeoutAction have inconsistency. HandleTimeoutType and ComRxDataTimeoutAction have inconsistency.
813	Even though HandleTimeoutType is REPLACE, ComTimeoutNotification is not set. If HandleTimeoutType is REPLACE, ComTimeoutNotification shall be set.
814	Even though HandleTimeoutType is REPLACE, InitValue is not set. If HandleTimeoutType is REPLACE, InitValue shall be set.
815	Even though HandleTimeoutType is REPLACE, AliveTimeout is not set. If HandleTimeoutType is REPLACE, AliveTimeout shall be set.

816	Both RteUsedOsAlarmRef and RteUsedOsSchTblExpiryPointRef in RteEventToTaskMapping exist. Only one of RteUsedOsAlarmRef or RteUsedOsSchTblExpiryPointRef in RteEventToTaskMapping shall exist.
817	Neither RteUsedOsAlarmRef nor RteUsedOsSchTblExpiryPointRef in RteEventToTaskMapping exists. One of RteUsedOsAlarmRef or RteUsedOsSchTblExpiryPointRef in RteEventToTaskMapping shall exist.
818	Both OsAlarmSetEvent and OsAlarmActivateTask in OsAlarmAction exist. Only one of OsAlarmSetEvent or OsAlarmActivateTask in OsAlarmAction shall exist.
819	Neither OsAlarmSetEvent nor OsAlarmActivateTask in OsAlarmAction exists. One of OsAlarmSetEvent or OsAlarmActivateTask in OsAlarmAction shall exist.
820	TimingEvent, DataReceivedEvent, DataReceiveErrorEvent, DataSendCompletedEvent, DataWriteCompletedEvent shall be mapped to OsTask. TimingEvent, DataReceivedEvent, DataReceiveErrorEvent, DataSendCompletedEvent, DataWriteCompletedEvent shall be mapped to OsTask.
821	RteUsedOsEventRef in RteEventToTaskMapping is not set for ExtendedTask. RteUsedOsEventRef in RteEventToTaskMapping shall be set for ExtendedTask.
822	OsTask is not mapped to OsApplication. OsTask shall be mapped to OsApplication.
823	OsAppEcucPartitionRef in OsApplication is empty or invalid. OsAppEcucPartitionRef in OsApplication shall be set with valid EcucPartition path.
824	For using EcucPartition, SwComponentPrototype is not mapped to EcucPartition. For using EcucPartition, SwComponentPrototype shall be mapped to EcucPartition.
825	There is mismatching partition between SwComponentPrototype and OsTask. There is mismatching partition between SwComponentPrototype and OsTask.
826	There is mismatching partition between SwComponentPrototype and OsAlarm. There is mismatching partition between SwComponentPrototype and OsAlarm.
827	OsAlarm is not mapped to OsApplication. OsAlarm shall be mapped to OsApplication.
828	The number of InitValue in RamBlock is over Multiplicity. The number of InitValue in RamBlock shall be within Multiplicity.
829	The number of InitValue in RomBlock is over Multiplicity. The number of InitValue in RomBlock shall be within Multiplicity.
835	The number of ServerCallPoints which have same ContextRPortRef and TargetProvidedOperationRef is over 1. The number of ServerCallPoints which have same ContextRPortRef and TargetProvidedOperationRef shall be 1.

836	The number of ServerCallPoints which have same ContextRPortRef and TargetProvidedOperationRef is over 1. The number of ServerCallPoints which have same ContextRPortRef and TargetProvidedOperationRef shall be 1.
837	ParameterValue of AutosarObject is empty or invalid. ParameterValue of AutosarObject shall be set.
838	ComNotification in ComSignal/ComSignalGroup is empty or invalid. ComNotification in ComSignal/ComSignalGroup shall be set with valid value.
839	ComNotification in ComSignal/ComSignalGroup is empty or invalid. ComNotification in ComSignal/ComSignalGroup shall be set with valid value.
840	ComTimeoutNotification in ComSignal/ComSignalGroup is empty or invalid. ComTimeoutNotification in ComSignal/ComSignalGroup shall be set with valid value.
841	ComTimeoutNotification in ComSignal/ComSignalGroup is empty or invalid. ComTimeoutNotification in ComSignal/ComSignalGroup shall be set with valid value.
842	ComErrorNotification in ComSignal/ComSignalGroup is empty or invalid. ComErrorNotification in ComSignal/ComSignalGroup shall be set with valid value.
843	ComInvalidNotification in ComSignal/ComSignalGroup is empty or invalid. ComInvalidNotification in ComSignal/ComSignalGroup shall be set with valid value.
844	Tx ComSignal/ComSignalGroup has Rte_COMCbk callback function as ComNotification. Tx ComSignal/ComSignalGroup shall not have Rte_COMCbk callback function as ComNotification.
845	Tx ComSignal/ComSignalGroup has Rte_COMCbkInv callback function as ComInvalidNotification. Tx ComSignal/ComSignalGroup shall not have Rte_COMCbkInv callback function as ComInvalidNotification.
846	Tx ComSignal/ComSignalGroup has Rte_COMCbkRxTOut callback function as ComTimeoutNotification. Tx ComSignal/ComSignalGroup shall not have Rte_COMCbkRxTOut callback function as ComTimeoutNotification.
847	Rx ComSignal/ComSignalGroup has Rte_COMCbkTAck callback function as ComNotification. Rx ComSignal/ComSignalGroup shall not have Rte_COMCbkTAck callback function as ComNotification.
848	Rx ComSignal/ComSignalGroup has Rte_COMCbkTErr callback function as ComErrorNotifcation. Rx ComSignal/ComSignalGroup shall not have Rte_COMCbkTErr callback function as ComErrorNotifcation.
849	Rx ComSignal/ComSignalGroup has Rte_COMCbkTxTOut callback function as ComTimeoutNotification. Rx ComSignal/ComSignalGroup shall not have Rte_COMCbkTxTOut callback function as

	ComTimeoutNotification.
850	QueueLength in ServerComSpec is lesser than or equal to 0. QueueLength in ServerComSpec shall be greater than 0.
851	The number of RteInitializationStrategy in RteInitializationBehavior is over 1. The number of RteInitializationStrategy in RteInitializationBehavior shall be 1.
852	HandleOutOfRange in QueuedSenderComSpec is INVALID or DEFAULT or EXTERNAL-REPLACEMENT. HandleOutOfRange in QueuedSenderComSpec shall not be INVALID or DEFAULT or EXTERNAL-REPLACEMENT.
853	HandleOutOfRangeStatus in ReceiverComSpec is INDICATE. HandleOutOfRangeStatus in ReceiverComSpec shall not be INDICATE.
854	ComSignalType in ComSignal/ComGroupSignal does not match with the ImplementationDataType. ComSignalType in ComSignal/ComGroupSignal shall match with the ImplementationDataType.
855	The number of ApplicationArrayElement in ApplicationArrayDataType is not 1. The number of ApplicationArrayElement in ApplicationArrayDataType shall be 1.
856	CompuMethodRef in SwDataDefProps is set with other categories except VALUE and BOOLEAN. CompuMethodRef in SwDataDefProps shall not be set with other categories except VALUE and BOOLEAN.
857	CompuMethodRef in SwDataDefProps is set with other categories except VALUE and TYPE_REFERENCE. CompuMethodRef in SwDataDefProps shall not be set with other categories except VALUE and TYPE_REFERENCE.
858	DataConstrRef in SwDataDefProps is set with other categories except VALUE and BOOLEAN. DataConstrRef in SwDataDefProps shall not be set with other categories except VALUE and BOOLEAN.
859	DataConstrRef in SwDataDefProps is set with other categories except VALUE. DataConstrRef in SwDataDefProps shall not be set with other categories except VALUE.
860	InvalidValue in SwDataDefProps is set with other categories except VALUE and BOOLEAN. InvalidValue in SwDataDefProps shall not be set with other categories except VALUE and BOOLEAN.
861	InvalidValue in SwDataDefProps is set with other categories except VALUE. InvalidValue in SwDataDefProps shall not be set with other categories except VALUE.
862	CompuMethodRef in SwDataDefProps is set with ApplicationArrayElement or ApplicationRecordElement. CompuMethodRef in SwDataDefProps shall not be set with ApplicationArrayElement or ApplicationRecordElement.

863	DataConstrRef in SwDataDefProps is set with ApplicationArrayElement or ApplicationRecordElement. DataConstrRef in SwDataDefProps shall not be set with ApplicationArrayElement or ApplicationRecordElement.
864	InvalidValue in SwDataDefProps is set with ApplicationArrayElement or ApplicationRecordElement. InvalidValue in SwDataDefProps shall not be set with ApplicationArrayElement or ApplicationRecordElement.
865	HandleOutOfRange in QueuedReceiverComSpec is INVALID or DEFAULT or EXTERNAL-REPLACEMENT. HandleOutOfRange in QueuedReceiverComSpec shall not be INVALID or DEFAULT or EXTERNAL-REPLACEMENT.
868	Even though IntervalType in Limit is INFINITE, Value in Limit is not empty. If IntervalType in Limit is INFINITE, Value in Limit shall be empty.
869	RteExpectedActivationOffset in RteUsedOsActivation is not multiples of OsSecondsPerTick in OsCounter. RteExpectedActivationOffset in RteUsedOsActivation shall be multiples of OsSecondsPerTick in OsCounter.
870	RteExpectedActivationOffset in RteUsedOsActivation is not within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter. RteExpectedActivationOffset in RteUsedOsActivation shall be within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter.
871	There are multiple RteEventToTaskMappings which have same RteEventRef in RteSwComponentInstance. There shall not be multiple RteEventToTaskMappings which have same RteEventRef in RteSwComponentInstance.
872	There are multiple RteSectionInitializationPolicies in RteInitializationBehavior. There shall not be multiple RteSectionInitializationPolicies in RteInitializationBehavior.
873	OsTask of RteMappedToTaskRef in RteEventToTaskMapping does not have OsEvent of RteUsedOsEventRef in RteEventToTaskMapping. OsTask of RteMappedToTaskRef in RteEventToTaskMapping shall have OsEvent of RteUsedOsEventRef in RteEventToTaskMapping.
874	RteEventRef in RteEventToTaskMapping is not member of RteSoftwareComponentInstanceRef in RteSwComponentInstance. RteEventRef in RteEventToTaskMapping shall be member of RteSoftwareComponentInstanceRef in RteSwComponentInstance.
875	Even though RteEventRef in RteEventToTaskMapping does not reference TimingEvent, RteActivationOffset in RteEventToTaskMapping is set. If RteEventRef in RteEventToTaskMapping does not reference TimingEvent, RteActivationOffset in RteEventToTaskMapping shall not be set.

876	RteActivationOffset in RteEventToTaskMapping is not multiples of OsSecondsPerTick in OsCounter. RteActivationOffset in RteEventToTaskMapping shall be multiples of OsSecondsPerTick in OsCounter.
877	RteActivationOffset in RteEventToTaskMapping is not within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter. RteActivationOffset in RteEventToTaskMapping shall be within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter.
878	Period in TimingEvent is not multiples of OsSecondsPerTick in OsCounter. Period in TimingEvent shall be multiples of OsSecondsPerTick in OsCounter.
879	Period in TimingEvent is not within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter. Period in TimingEvent shall be within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter.
880	RteVariableWriteAccessRef in RteImplicitCommunication is invalid. RteVariableWriteAccessRef in RteImplicitCommunication shall be valid.
881	RteVariableReadAccessRef in RteImplicitCommunication is invalid. RteVariableReadAccessRef in RteImplicitCommunication shall be valid.
882	OsTask of RteVirtuallyMappedToTaskRef in RteEventToTaskMapping is not activated by OsAlarm of RteUsedOsAlarmRef in RteEventToTaskMapping. OsTask of RteVirtuallyMappedToTaskRef in RteEventToTaskMapping shall be activated by OsAlarm of RteUsedOsAlarmRef in RteEventToTaskMapping.
883	OsTask of RteMappedToTaskRef in RteEventToTaskMapping is not activated by OsAlarm of RteUsedOsAlarmRef in RteEventToTaskMapping. OsTask of RteMappedToTaskRef in RteEventToTaskMapping shall be activated by OsAlarm of RteUsedOsAlarmRef in RteEventToTaskMapping.
884	OsTask of RteVirtuallyMappedToTaskRef in RteEventToTaskMapping does not have OsEvent of RteUsedOsEventRef in RteEventToTaskMapping. OsTask of RteVirtuallyMappedToTaskRef in RteEventToTaskMapping shall have OsEvent of RteUsedOsEventRef in RteEventToTaskMapping.
885	OsTask of RteMappedToTaskRef in RteEventToTaskMapping does not have OsEvent of RteUsedOsEventRef in RteEventToTaskMapping. OsTask of RteMappedToTaskRef in RteEventToTaskMapping shall have OsEvent of RteUsedOsEventRef in RteEventToTaskMapping.
886	RteBswActivationOffset in RteBswEventToTaskMapping is not multiples of OsSecondsPerTick in OsCounter. RteBswActivationOffset in RteBswEventToTaskMapping shall be multiples of OsSecondsPerTick in OsCounter.
887	RteBswActivationOffset in RteBswEventToTaskMapping is not within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter.

	RteBswActivationOffset in RteBswEventToTaskMapping shall be within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter.
888	Period in BswTimingEvent is not multiples of OsSecondsPerTick in OsCounter. Period in BswTimingEvent shall be multiples of OsSecondsPerTick in OsCounter.
889	Period in BswTimingEvent is not within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter. Period in BswTimingEvent shall be within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter.
890	Even though RteBswEventRef in RteBswEventToTaskMapping does not reference BswTimingEvent, RteBswActivationOffset in RteBswEventToTaskMapping is set. If RteBswEventRef in RteBswEventToTaskMapping does not reference BswTimingEvent, RteBswActivationOffset in RteBswEventToTaskMapping shall not be set.
891	OsTask of RteUsedOsEventRef in RteEventToTaskMapping is not activated by OsAlarm of RteUsedOsAlarmRef in RteEventToTaskMapping. OsTask of RteUsedOsEventRef in RteEventToTaskMapping shall be activated by OsAlarm of RteUsedOsAlarmRef in RteEventToTaskMapping.
892	There are multiple NvRequireComSpecs which have VariableRef in RPortPrototype. There shall not be multiple NvRequireComSpecs which have VariableRef in RPortPrototype.
893	There are multiple ParameterRequireComSpecs of ParameterRef in RPortPrototype. There shall be single ParameterRequireComSpec of ParameterRef in RPortPrototype.
894	There are multiple ReceiverComSpecs which have DataElementRef in RPortPrototype. There shall not be multiple ReceiverComSpecs which have DataElementRef in RPortPrototype.
895	There are multiple ServerComSpecs which have OperationRef in PortPrototype. There shall not be multiple ServerComSpec which have OperationRef in PortPrototype.
896	There are multiple NvProvideComSpecs which have VariableRef in PPortPrototype. There shall not be multiple NvProvideComSpecs which have VariableRef in PPortPrototype.
897	There are multiple ParameterProvideComSpecs which have same ParameterRef in PPortPrototype. There shall not be multiple ParameterProvideComSpecs which have same ParameterRef in PPortPrototype.
898	There are multiple SenderComSpecs which have same DataElementRef in PPortPrototype. There shall not be multiple SenderComSpecs which have same DataElementRef in PPortPrototype.
899	PortPrototype in NvBlockSwComponentType has InterfaceTypeTRef. which references InterfaceIsService in Interface is not FALSE . PortPrototype in NvBlockSwComponentType shall not have InterfaceTypeTRef. which references InterfaceIsService in Interface shall be FALSE .
900	PortPrototype in NvBlockSwComponentType has InterfaceTypeTRef which does not reference NvDataInterface or ClientServerInterface.

	PortPrototype in NvBlockSwComponentType shall have InterfaceTypeTRef which references NvDataInterface or ClientServerInterface.
901	Even though InvalidationPolicy in VariableDataPrototype is REPLACE, InitValue and InvalidValue is same. If InvalidationPolicy in VariableDataPrototype is REPLACE, InitValue and InvalidValue shall be different.
902	Even though implicit Sender-Receiver Communication is used, EnableUpdate in NonQueuedReceiverComSpec is set TRUE. If implicit Sender-Receiver Communication is used, EnableUpdate in NonQueuedReceiverComSpec shall be FALSE.
903	Neither NumericalValueSpecification , ConstantReference, nor ApplicationValueSpecification in InvalidValue exists. One of NumericalValueSpecification , ConstantReference, or ApplicationValueSpecification in InvalidValue shall exist.
904	Both PPortInCompositionInstanceRef and RPortInCompositionInstanceRef in InnerPortIRef exist. Only one of PPortInCompositionInstanceRef or RPortInCompositionInstanceRef in InnerPortIRef shall exist.
905	Neither PPortInCompositionInstanceRef nor RPortInCompositionInstanceRef in InnerPortIRef exists. One of PPortInCompositionInstanceRef or RPortInCompositionInstanceRef in InnerPortIRef shall exist.
906	If ImplementationDataType has Category of ARRAY, RTE does not support ImplementationDataTypeElement which has Category of ARRAY or STRUCTURE. Please use ImplementationDataTypeElement which has Category of TYPE_REFERENCE for multi dimensional array or array of structure.
907	Category in ImplementationDataType is FUNCTION_REFERENCE. Category in ImplementationDataType shall not be FUNCTION_REFERENCE.
908	Category in ApplicationValueSpec is not VALUE or BOOLEAN. Category in ApplicationValueSpec shall be VALUE or BOOLEAN.
909	Category in ApplicationRecordDataType is not STRUCTURE. Category in ApplicationRecordDataType shall be STRUCTURE.
910	Category in ApplicationRecordElement and Category in ApplicationDataType which TypeTRef in ApplicationRecordElement references is not identical. Category in ApplicationRecordElement and Category in ApplicationDataType which TypeTRef in ApplicationRecordElement references shall be identical.
911	Both LocalParameterRef and AutosarParameterRef in AccessedParameter exist. Only one of LocalParameterRef or AutosarParameterRef in AccessedParameter shall exist.
912	Both LocalVariableRef and AutosarVariableIRef in AccessedVariable exist. Only One of LocalVariableRef or AutosarVariableIRef in AccessedVariable shall exist.

913	Neither LocalVariableRef nor AutosarVariableIRef in AccessedVariable exists. One of LocalVariableRef or AutosarVariableIRef in AccessedVariable shall exist.
914	Neither LocalParameterRef nor AutosarParameterRef in AccessedParameter exists. One of LocalParameterRef or AutosarParameterRef in AccessedParameter shall exist.
915	ImplementationDataType of VariableDataPrototype and ImplementationDataType of VariableDataPrototype is inconsistent. ImplementationDataType of VariableDataPrototype and ImplementationDataType of VariableDataPrototype shall be consistent.
916	ContextComponentRef in DataElementIRef is empty or invalid. ContextComponentRef in DataElementIRef shall be set with valid SwComponentPrototype path.
917	ContextPortRef in DataElementIRef is empty or invalid. ContextPortRef in DataElementIRef shall be set with valid PortPrototype path.
918	TargetDataPrototypeRef in DataElementIRef is empty or invalid. TargetDataPrototypeRef in DataElementIRef shall be set with valid DataElement Path.
919	RteSoftwareComponentInstanceRef in RteImplicitCommunication is empty. RteSoftwareComponentInstanceRef in RteImplicitCommunication shall be set.
920	RteSoftwareComponentInstanceRef in RteImplicitCommunication is invalid. RteSoftwareComponentInstanceRef in RteImplicitCommunication shall be set with valid SwComponentPrototype path.
921	RteSoftwareComponentInstanceRef in RteImplicitCommunication is duplicated. RteSoftwareComponentInstanceRef in RteImplicitCommunication shall be unique.
922	PortPrototypeRef in AutosarParameterIRef is empty or invalid. PortPrototypeRef in AutosarParameterIRef shall be set with valid PortPrototype Path.
923	PortPrototypeRef in AutosarVariableIRef is empty or invalid. PortPrototypeRef in AutosarVariableIRef shall be set with valid PortPrototype Path.
924	PortPrototypeRef in RoleBasedPortAssignment is empty or invalid. PortPrototypeRef in RoleBasedPortAssignment shall be set with valid PortPrototype Path.
925	TargetDataPrototypeRef in AutosarParameterIRef is empty or invalid. TargetDataPrototypeRef in AutosarParameterIRef shall be set with valid ParameterDataPrototype Path.
926	TargetDataPrototypeRef in AutosarVariableInImplDataType is empty or invalid. TargetDataPrototypeRef in AutosarVariableInImplDataType shall be set with valid VariableDataPrototype Path.
927	TargetDataPrototypeRef in AutosarVariableIRef is empty or invalid. TargetDataPrototypeRef in AutosarVariableIRef shall be set with valid VariableDataPrototype Path.

928	ComSystemTemplateSystemSignalRef in ComSignal is empty or invalid. ComSystemTemplateSystemSignalRef in ComSignal shall be set with valid ISignalToIPduMapping path.
929	ComSystemTemplateSystemSignalRef in ComGroupSignal is empty or invalid. ComSystemTemplateSystemSignalRef in ComGroupSignal shall be set with valid ISignalToIPduMapping path.
930	ComSystemTemplateSystemSignalGroupRef in ComSignalGroup is empty or invalid. ComSystemTemplateSystemSignalGroupRef in ComSignalGroup shall be set with valid ISignalToIPduMapping path.
931	BehaviorRef in BswImplementation is empty or invalid. BehaviorRef in BswImplementation shall be set with valid BswInternalBehavior path.
932	BehaviorRef in SwcImplementation is empty or invalid. BehaviorRef in SwcImplementation shall be set with valid SwcInternalBehavior path.
933	SystemSignalRef in ISignal is empty or invalid. SystemSignalRef in ISignal shall be set with valid SystemSignal path.
934	SystemSignalRef in SenderReceiverToSignalMapping is empty or invalid. SystemSignalRef in SenderReceiverToSignalMapping shall be set with valid SystemSignal Path.
935	SystemSignalRef in SenderRecArrayElementMapping is empty or invalid. SystemSignalRef in SenderRecArrayElementMapping shall be set with valid SystemSignal path.
936	SystemSignalRef in SenderRecRecordElementMapping is empty or invalid. SystemSignalRef in SenderRecRecordElementMapping shall be set with valid SystemSignal path.
937	SystemSignalGroupRef in ISignalGroup is empty or invalid. SystemSignalGroupRef in ISignalGroup shall be set with valid SystemSignalGroup path.
938	SystemSignalGroupRef in SenderReceiverToSignalGroupMapping is empty or invalid. SystemSignalGroupRef in SenderReceiverToSignalGroupMapping shall be set with valid SystemSignalGroup path.
939	SystemSignalRef in SystemSignalGroup is empty. SystemSignalRef in SystemSignalGroup shall be set with valid SystemSignal path.
940	SystemSignalRef in SystemSignalGroup is invalid. SystemSignalRef in SystemSignalGroup shall be set with valid SystemSignal path.
941	SenderRecArrayTypeMapping in SenderReceiverToSignalGroupMapping is empty. SenderRecArrayTypeMapping in SenderReceiverToSignalGroupMapping shall be set.
942	SenderRecRecordTypeMapping in SenderReceiverToSignalGroupMapping is empty. SenderRecRecordTypeMapping in SenderReceiverToSignalGroupMapping shall be set.
950	Index in IndexedArrayElement is empty. Index in IndexedArrayElement shall be set.

951	ImplementationArrayElementRef in IndexedArrayElement is empty. ImplementationArrayElementRef in IndexedArrayElement shall be set.
952	ImplementationArrayElementRef in IndexedArrayElement is set. ImplementationArrayElementRef in IndexedArrayElement shall not be set.
953	ApplicationArrayElementRef in IndexedArrayElement is empty. ApplicationArrayElementRef in IndexedArrayElement shall be set.
954	ApplicationArrayElementRef in IndexedArrayElement is set. ApplicationArrayElementRef in IndexedArrayElement shall not be set.
955	IndexedArrayElement in SenderRecArrayElementMapping is not set. IndexedArrayElement in SenderRecArrayElementMapping shall be set.
956	SenderRecArrayTypeMapping in SenderRecArrayElementMapping is not set. SenderRecArrayTypeMapping in SenderRecArrayElementMapping shall be set.
957	SenderRecArrayTypeMapping in SenderRecArrayElementMapping is set. SenderRecArrayTypeMapping in SenderRecArrayElementMapping shall not be set.
958	SenderRecRecordTypeMapping in SenderRecArrayElementMapping is not set. SenderRecRecordTypeMapping in SenderRecArrayElementMapping shall be set.
959	SenderRecRecordTypeMapping in SenderRecArrayElementMapping is set. SenderRecRecordTypeMapping in SenderRecArrayElementMapping shall not be set.
960	ImplementationRecordElementRef in SenderRecRecordElementMapping is empty. ImplementationRecordElementRef in SenderRecRecordElementMapping shall be set.
961	ImplementationRecordElementRef in SenderRecRecordElementMapping is set. ImplementationRecordElementRef in SenderRecRecordElementMapping shall not be set.
962	ApplicationRecordElementRef in SenderRecRecordElementMapping is empty. ApplicationRecordElementRef in SenderRecRecordElementMapping shall be set.
963	ApplicationRecordElementRef in SenderRecRecordElementMapping is set. ApplicationRecordElementRef in SenderRecRecordElementMapping shall not be set.
964	SenderRecArrayElementMapping for Index in SenderRecArrayTypeMapping does not exist. SenderRecArrayElementMapping for Index in SenderRecArrayTypeMapping shall exist.
965	The number of SenderRecArrayElementMapping in SenderRecArrayTypeMapping is not correct. The number of SenderRecArrayElementMapping in SenderRecArrayTypeMapping shall be correct.
966	The number of SenderRecRecordElementMapping in SenderRecRecordTypeMapping is not correct. The number of SenderRecRecordElementMapping in SenderRecRecordTypeMapping shall be correct.
967	SenderRecRecordElementMapping for ImplementationDataTypeElement in SenderRecArrayTypeMapping does not exist. SenderRecRecordElementMapping for ImplementationDataTypeElement in

	SenderRecArrayTypeMapping shall exist.
968	SenderRecArrayTypeMapping in SenderReceiverToSignalGroupMapping is set. SenderRecArrayTypeMapping in SenderReceiverToSignalGroupMapping shall not be set.
969	SenderRecRecordTypeMapping in SenderReceiverToSignalGroupMapping is set. SenderRecRecordTypeMapping in SenderReceiverToSignalGroupMapping shall not be set.
970	DataElementIRef in SenderReceiverToSignalMapping is empty. DataElementIRef in SenderReceiverToSignalMapping shall be set.
971	DataElementIRef in SenderReceiverToSignalGroupMapping is empty. DataElementIRef in SenderReceiverToSignalGroupMapping shall be set.
972	Activation in SwcModeSwitchEvent is empty or invalid. Activation in SwcModeSwitchEvent shall be set with valid value.
973	SenderRecRecordElementMapping for ApplicationRecordElement in SenderRecArrayTypeMapping does not exist. SenderRecRecordElementMapping for ApplicationRecordElement in SenderRecArrayTypeMapping shall exist.
974	SenderRecRecordTypeMapping in SenderRecRecordElementMapping is set. SenderRecRecordTypeMapping in SenderRecRecordElementMapping shall not be set.
975	SenderRecRecordTypeMapping in SenderRecRecordElementMapping is not set. SenderRecRecordTypeMapping in SenderRecRecordElementMapping shall be set.
976	SenderRecArrayTypeMapping in SenderRecRecordElementMapping is set. SenderRecArrayTypeMapping in SenderRecRecordElementMapping shall not be set.
977	SenderRecArrayTypeMapping in SenderRecRecordElementMapping is not set. SenderRecArrayTypeMapping in SenderRecRecordElementMapping shall be set.
978	Category in ApplicationArrayElement and Category in ApplicationDataType which TypeTRef in ApplicationArrayElement references is not identical. Category in ApplicationArrayElement and Category in ApplicationDataType which TypeTRef in ApplicationArrayElement references shall be identical.
979	InitValue in NonQueuedReceiverComSpec is empty. InitValue in NonQueuedReceiverComSpec shall be set.
981	InitValue in ParameterDataPrototype is empty. InitValue in ParameterDataPrototype shall be set.
982	InitValue in ParameterProvideComSpec is empty. InitValue in ParameterProvideComSpec shall be set.
985	InitValue in RamBlock is empty. InitValue in RamBlock shall be set.

986	InitValue in RomBlock is empty. InitValue in RomBlock shall be set.
987	InitValue in VariableDataPrototype is empty. InitValue in VariableDataPrototype shall be set.
988	RteExclusiveAreaRef in RteExclusiveAreaImplementation is empty or invalid. RteExclusiveAreaRef in RteExclusiveAreaImplementation shall be set with valid ExclusiveArea path.
989	RteExclusiveAreaImplMechanism in RteExclusiveAreaImplementation is empty or invalid. RteExclusiveAreaImplMechanism in RteExclusiveAreaImplementation shall be set with valid value.
990	RteExclusiveAreaOsResourceRef in RteExclusiveAreaImplementation is empty or invalid. RteExclusiveAreaOsResourceRef in RteExclusiveAreaImplementation shall be set with valid OsResource path.
991	RteBswExclusiveAreaRef in RteBswExclusiveAreaImpl is empty or invalid. RteBswExclusiveAreaRef in RteBswExclusiveAreaImpl shall be set with valid ExclusiveArea path.
992	RteExclusiveAreaImplMechanism in RteBswExclusiveAreaImpl is empty or invalid. RteExclusiveAreaImplMechanism in RteBswExclusiveAreaImpl shall be set with valid value.
993	RteBswExclusiveAreaOsResourceRef in RteBswExclusiveAreaImpl is empty or invalid. RteBswExclusiveAreaOsResourceRef in RteBswExclusiveAreaImpl shall be set with valid OsResource path.
994	There is no RteExclusiveAreaImplementation for the ExclusiveArea. There shall be a RteExclusiveAreaImplementation for the ExclusiveArea.
995	There is no RteBswExclusiveAreaImpl for the ExclusiveArea. There shall be a RteBswExclusiveAreaImpl for the ExclusiveArea.
996	ISignalRef in ISignalGroup is invalid. ISignalRef in ISignalGroup shall be set with valid ISignal Path.
997	HandleOutOfRange values in all ISignals which are referenced by one ISignalGroup are not same. HandleOutOfRange values in all ISignals which are referenced by one ISignalGroup shall be same.
998	HandleOutOfRange in iSignalGroup and HandleOutOfRange in DataElement is not same. HandleOutOfRange in iSignalGroup and HandleOutOfRange in DataElement shall be same.
1000	RteBswReleasedTriggerModInstRef in RteBswRequiredTriggerConnection is empty or invalid. RteBswReleasedTriggerModInstRef in RteBswRequiredTriggerConnection shall be set with RteBswModuleInstance path.
1001	RteBswReleasedTriggerRef in RteBswRequiredTriggerConnection is empty or invalid. RteBswReleasedTriggerRef in RteBswRequiredTriggerConnection shall be set with

	RteBswModuleInstance path.
1002	RteBswRequiredTriggerRef in RteBswRequiredTriggerConnection is empty or invalid. RteBswRequiredTriggerRef in RteBswRequiredTriggerConnection shall be set with RteBswModuleInstance path.
1003	RteComponentTypeRef in RteSwComponentType is empty or invalid. RteComponentTypeRef in RteSwComponentType shall be set with valid SwComponentType path.
1004	SwcBswMappingRef in BswImplementation is empty or invalid. SwcBswMappingRef in BswImplementation shall be set with valid SwComponentType path.
1005	BehaviorRef in BswImplementation and BehaviorRef in BswImplementation is same. BehaviorRef in BswImplementation and BehaviorRef in BswImplementation shall be different.
1006	SwcBswMappingRef in SwcImplementation is empty or invalid. SwcBswMappingRef in SwcImplementation shall be set with valid SwComponentType path.
1007	SwcBehaviorRef in SwcBswMapping is empty or invalid. SwcBehaviorRef in SwcBswMapping shall be set with valid InternalBehavior path.
1008	BehaviorRef in BswImplementation andSwcBswMappingRef in BswImplementation is mismatching. BehaviorRef in BswImplementation andSwcBswMappingRef in BswImplementation shall be matching.
1009	BehaviorRef in SwcImplementation andSwcBswMappingRef in SwcImplementation is mismatching. BehaviorRef in SwcImplementation andSwcBswMappingRef in SwcImplementation shall be matching.
1010	SwcBswMappingRef in BswImplementation and SwcBswMappingRef in BswImplementation is same. SwcBswMappingRef in BswImplementation and SwcBswMappingRef in BswImplementation shall be different.
1011	SwcBswMappingRef in SwcImplementation and SwcBswMappingRef in SwcImplementation is same. SwcBswMappingRef in SwcImplementation and SwcBswMappingRef in SwcImplementation shall be different.
1013	There is no BswImplementation for the BswInternalBehavior. There shall be a BswImplementation for the BswInternalBehavior.
1014	There is no SwcImplementation for the InternalBehavior. There shall be a SwcImplementation for the InternalBehavior.
1015	There is no BswImplementation for the SwcBswMapping. There shall be a BswImplementation for the SwcBswMapping.
1016	There is no SwcImplementation for the SwcBswMapping.

	There shall be a SwcImplementation for the SwcBswMapping.
1017	SwcBehaviorRef in SwcBswMapping and SwcBehaviorRef in SwcBswMapping is same. SwcBehaviorRef in SwcBswMapping and SwcBehaviorRef in SwcBswMapping shall not be same.
1018	BswBehaviorRef in SwcBswMapping and BswBehaviorRef in SwcBswMapping is same. BswBehaviorRef in SwcBswMapping and BswBehaviorRef in SwcBswMapping shall not be same.
1019	ModeDeclarationGroupPrototypeRef in ManagedModeGroup and ModeDeclarationGroupPrototypeRef in ManagedModeGroup is same. ModeDeclarationGroupPrototypeRef in ManagedModeGroup and ModeDeclarationGroupPrototypeRef in ManagedModeGroup shall not be same.
1020	TriggerRef in BswExternalTriggerOccurredEvent is empty or invalid. TriggerRef in BswExternalTriggerOccurredEvent shall be set with valid RequiredTrigger path.
1021	TriggerRef in IssuedTrigger is empty or invalid. TriggerRef in IssuedTrigger shall be set with valid ReleasedTrigger path.
1022	BswInternalTriggeringPointRef in BswInternalTriggeringPointRefConditional is empty or invalid. BswInternalTriggeringPointRef in BswInternalTriggeringPointRefConditional shall be set with valid BswInternalTriggeringPoint path.
1023	TriggerRef in TriggerRefConditional is empty or invalid. TriggerRef in TriggerRefConditional shall be set.
1024	ModeDeclarationGroupPrototypeRef in ManagedModeGroup is empty or invalid. ModeDeclarationGroupPrototypeRef in ManagedModeGroup shall be set with valid ProvidedModeGroup path.
1025	ModeDeclarationGroupPrototypeRef in ModeDeclarationGroupPrototypeRefConditional is empty or invalid. ModeDeclarationGroupPrototypeRef in ModeDeclarationGroupPrototypeRefConditional shall be set.
1026	SwcRunnableRef in SwcBswRunnableMapping is empty or invalid. SwcRunnableRef in SwcBswRunnableMapping shall be set with valid BswEntity path.
1027	Even though BswModuleEntry is referenced by CalledEntry, CallType in called BswModuleEntry is not REGULAR or CALLBACK. If BswModuleEntry is referenced by CalledEntry, CallType in called BswModuleEntry shall be REGULAR or CALLBACK.
1028	Even though BswModuleEntry is referenced by CalledEntry, ExecutionContext in called BswModuleEntry is not identical to ExecutionContext in caller BswModuleEntry. If BswModuleEntry is referenced by CalledEntry, ExecutionContext in called BswModuleEntry shall be identical to ExecutionContext in caller BswModuleEntry.
1029	BswModuleEntryRef in CalledEntry does not refer to an element declared as OutgoingCallback, ProvidedEntry or as RequiredEntry in BswModuleDependency. BswModuleEntryRef in CalledEntry shall refer to an element declared as OutgoingCallback,

	ProvidedEntry or as RequiredEntry in BswModuleDependency.
1030	HandleOutOfRange in SenderComSpec/ReceiverComSpec is EXTERNAL-REPLACEMENT. HandleOutOfRange in SenderComSpec/ReceiverComSpec shall not be EXTERNAL-REPLACEMENT.
1031	Symbol of RunnableEntity is different with Symbol of BswEntity. Symbol of RunnableEntity shall be same with Symbol of BswEntity.
1032	ModeDeclarationGroupPrototypeRef in AccessedModeGroup is empty or invalid. ModeDeclarationGroupPrototypeRef in AccessedModeGroup shall be set with valid ProvidedModeGroup path.
1033	ModeDeclarationGroupPrototypeRef in AccessedModeGroup and ModeDeclarationGroupPrototypeRef in AccessedModeGroup is same. ModeDeclarationGroupPrototypeRef in AccessedModeGroup and ModeDeclarationGroupPrototypeRef in AccessedModeGroup shall not be same.
1034	RteBswEventRef in RteBswEventToTaskMapping is empty or invalid. RteBswEventRef in RteBswEventToTaskMapping shall be set.
1035	CallType in BswModuleEntry is empty or invalid. CallType in BswModuleEntry shall be set.
1036	PortPrototype and VariableDataPrototype for the RamBlock is connected to SWC Instances of different partitions. PortPrototype and VariableDataPrototype for the RamBlock shall not be connected to SWC Instances of different partitions.
1037	Role in RoleBasedPortAssignment is empty or invalid. Role in RoleBasedPortAssignment shall be set with valid value.
1038	ReadNvData in NvBlockDataMapping is empty. ReadNvData in NvBlockDataMapping shall be exist.
1039	WrittenNvData in NvBlockDataMapping is empty. WrittenNvData in NvBlockDataMapping shall be exist.
1040	Neither RteVariableReadAccessRef nor RteVariableWriteAccessRef in RteImplicitCommunication is exist. RteVariableReadAccessRef or RteVariableWriteAccessRef in RteImplicitCommunication shall be exist.
1041	RteEventRef in RteEventToTaskMapping is empty or invalid. RteEventRef in RteEventToTaskMapping shall be set with valid RteEvent path.
1042	VariableDataPrototype referenced by a SenderReceiverInterface doesn't have a STANDARD or QUEUED SwImplPolicy. VariableDataPrototype referenced by a SenderReceiverInterface shall have a STANDARD or QUEUED SwImplPolicy.

1043	ParameterDataPrototype doesn't have a STANDARD, FIXED or CONST SwImplPolicy. ParameterDataPrototype shall have a STANDARD, FIXED or CONST SwImplPolicy.
1044	HandleOutOfRange in ReceiverComSpec is EXTERNAL-REPLACEMENT. HandleOutOfRange in ReceiverComSpec shall not be EXTERNAL-REPLACEMENT.
1045	There are mismatching function prototypes for the RunnableEntity. Function prototypes for the RunnableEntity shall be consistent. In case only argument name is different, use RunnableEntityArgument Configuration Parameter in RunnableEntity.
1046	There are multiple DataWriteAccesses for the same PortPrototypeRef and TargetDataPrototypeRef. There shall be only one DataWriteAccess for the PortPrototypeRef and TargetDataPrototypeRef.
1047	There are multiple DataSendPoints for the same PortPrototypeRef and TargetDataPrototypeRef. There shall be only one DataSendPoint for the PortPrototypeRef and TargetDataPrototypeRef.
1048	There are multiple DataReadAccesses for the same PortPrototypeRef and TargetDataPrototypeRef. There shall be only one DataReadAccess for the PortPrototypeRef and TargetDataPrototypeRef.
1049	There are multiple DataReceivePointByArguments for the same PortPrototypeRef and TargetDataPrototypeRef. There shall be only one DataReceivePointByArgument for the PortPrototypeRef and TargetDataPrototypeRef.
1050	There are multiple DataReceivePointByValues for the same PortPrototypeRef and TargetDataPrototypeRef. There shall be only one DataReceivePointByValue for the PortPrototypeRef and TargetDataPrototypeRef.
1051	ImplementedEntryRef in BswEntity and ImplementedEntryRef in BswEntity is same. ImplementedEntryRef in BswEntity and ImplementedEntryRef in BswEntity shall be different.
1052	The ComDataInvalidAction in ComSignal/ComSignalGroup and the HandleInvalid in VariableDataPrototype are mismatching. The ComDataInvalidAction in ComSignal/ComSignalGroup and the HandleInvalid in VariableDataPrototype shall be matching.
1053	The ComSignalDataInvalidValue in ComSignal is empty even though the ComDataInvalidAction in ComSignal is set as NOTIFY or REPLACE. The ComSignalDataInvalidValue in ComSignal shall be set if the ComDataInvalidAction in ComSignal is set as NOTIFY or REPLACE.
1054	ComSignalGroup is mapped to dynamic length of array datatype. Dynamic length of array data type shall be mapped to ComSignal.
1055	The ComSignalDataInvalidValue in ComGroupSignal is empty even though the ComDataInvalidAction in ComSignalGroup is set as NOTIFY or REPLACE. The ComSignalDataInvalidValue in ComGroupSignal shall be set if the ComDataInvalidAction in

	ComSignalGroup is set as NOTIFY or REPLACE.
1056	ComSignalDataInvalidValue in ComGroupSignal and InvalidValue in DataType is mismatching. ComSignalDataInvalidValue in ComGroupSignal and InvalidValue in DataType shall be matching.
1057	ComSignalDataInvalidValue in ComSignal and InvalidValue in DataType is mismatching. ComSignalDataInvalidValue in ComSignal and InvalidValue in DataType shall be matching.
1058	Osloc container in the OS Ecud Arxml File exists. Osloc container in the OS Ecud Arxml File shall not exist. Rte creates the Osloc contatiner automatically if necessary.
1059	RTE Generator cannot find OS Ecud Arxml File. Please check whether RTE Generator has OS Ecud Arxml file as an argument.
1060	RTE Generator cannot find DefinitionRef in OS Configuration. Please check whether OS Ecud Arxml File has a DefinitionRef in OS Configuration.
1061	RteUsedOsAlarmRef in RteEventToTaskMapping is empty or invalid. RteUsedOsAlarmRef in RteEventToTaskMapping shall be set with valid OsAlarm Path.
1062	RteComponentTypeRef in RteSwComponentType and RteComponentTypeRef in RteSwComponentType is same. RteComponentTypeRef in RteSwComponentType and RteComponentTypeRef in RteSwComponentType shall be different.
1063	ShortName of OsTask is duplicated. ShortName of OsTask shall be unique.
1064	RteActivationOsAlarmRef in RteUsedOsActivation and RteActivationOsAlarmRef in RteUsedOsActivation is same. RteActivationOsAlarmRef in RteUsedOsActivation and RteActivationOsAlarmRef in RteUsedOsActivation shall be different.
1065	RteActivationOsSchTblRef in RteUsedOsActivation and RteActivationOsSchTblRef in RteUsedOsActivation is same. RteActivationOsSchTblRef in RteUsedOsActivation and RteActivationOsSchTblRef in RteUsedOsActivation shall be different.
1066	Both RteActivationOsAlarmRef and RteActivationOsSchTblRef in RteUsedOsActivation is empty. One of RteActivationOsAlarmRef or RteActivationOsSchTblRef in RteUsedOsActivation shall be set.
1067	Both RteActivationOsAlarmRef and RteActivationOsSchTblRef in RteUsedOsActivation is set. Only one of RteActivationOsAlarmRef or RteActivationOsSchTblRef in RteUsedOsActivation shall be set.
1068	RteActivationOsAlarmRef in RteUsedOsActivation is empty or invalid. RteActivationOsAlarmRef in RteUsedOsActivation shall be set with valid OsAlarm Path.
1069	RteActivationOsSchTblRef in RteUsedOsActivation is empty or invalid.

	RteActivationOsSchTblRef in RteUsedOsActivation shall be set with valid OsAlarm Path.
1070	Both RteBswUsedOsAlarmRef and RteBswUsedOsSchTblExpiryPointRef in RteBswEventToTaskMapping exist. Only one of RteBswUsedOsAlarmRef or RteBswUsedOsSchTblExpiryPointRef in RteBswEventToTaskMapping shall exist.
1071	Neither RteBswUsedOsAlarmRef nor RteBswUsedOsSchTblExpiryPointRef in RteBswEventToTaskMapping exists. One of RteBswUsedOsAlarmRef or RteBswUsedOsSchTblExpiryPointRef in RteBswEventToTaskMapping shall exist.
1072	RteBswUsedOsAlarmRef in RteBswEventToTaskMapping is empty or invalid. RteBswUsedOsAlarmRef in RteBswEventToTaskMapping shall be set with valid OsAlarm Path.
1073	RteBswUsedOsEventRef in RteBswEventToTaskMapping is empty or invalid. RteBswUsedOsEventRef in RteBswEventToTaskMapping shall be set with valid OsAlarm Path.
1074	OsTask of RteBswUsedOsEventRef in RteBswEventToTaskMapping is not activated by OsAlarm of RteBswUsedOsAlarmRef in RteBswEventToTaskMapping. OsTask of RteBswUsedOsEventRef in RteBswEventToTaskMapping shall be activated by OsAlarm of RteBswUsedOsAlarmRef in RteBswEventToTaskMapping.
1075	OsTask of RteBswMappedToTaskRef in RteBswEventToTaskMapping is not activated by OsAlarm of RteBswUsedOsAlarmRef in RteBswEventToTaskMapping. OsTask of RteBswMappedToTaskRef in RteBswEventToTaskMapping shall be activated by OsAlarm of RteBswUsedOsAlarmRef in RteBswEventToTaskMapping.
1076	OsTask of RteBswMappedToTaskRef in RteBswEventToTaskMapping does not have OsEvent of RteBswUsedOsEventRef in RteBswEventToTaskMapping. OsTask of RteBswMappedToTaskRef in RteBswEventToTaskMapping shall have OsEvent of RteBswUsedOsEventRef in RteBswEventToTaskMapping.
1077	RteExpectedTickDuration in RteUsedOsActivation is not multiples of OsSecondsPerTick in OsCounter. RteExpectedTickDuration in RteUsedOsActivation shall be multiples of OsSecondsPerTick in OsCounter.
1078	RteExpectedTickDuration in RteUsedOsActivation is not within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter. RteExpectedTickDuration in RteUsedOsActivation shall be within a range of 0 and OsSecondsPerTick * OsCounterMaxAllowedValue in OsCounter.
1079	RteEvents/RteBswEvent except BackgroundEvent and BswBackgroundEvent is mapped to Background OsTask RteEvent/RteBswEvent except BackgroundEvent and BswBackgroundEvent shall not be mapped to Background OsTask
1080	The priority of background OsTask is higher than or equal to the priority of nonbackground OsTask. The priority of background OsTask shall be lower than the priority of nonbackground OsTask.

1081	RteUsedOsSchTblExpiryPointRef in RteEventToTaskMapping is empty or invalid RteUsedOsSchTblExpiryPointRef in RteEventToTaskMapping shall be set with valid OsScheduleTableExpiryPoint Path.
1082	RteUsedOsAlarmRef in RteEventToTaskMapping is empty. RteUsedOsAlarmRef in RteEventToTaskMapping shall be set with valid OsAlarm Path.
1083	RteUsedOsSchTblExpiryPointRef in RteEventToTaskMapping is set. RteUsedOsSchTblExpiryPointRef in RteEventToTaskMapping shall not be set.
1084	RteUsedOsEventRef in RteEventToTaskMapping is empty or invalid. RteUsedOsEventRef in RteEventToTaskMapping shall be set with valid OsEvent Path.
1085	RteBswUsedOsEventRef in RteBswEventToTaskMappingis empty or invalid. RteBswUsedOsEventRef in RteBswEventToTaskMapping shall be set with valid OsEvent Path.
1086	RteBswUsedOsSchTblExpiryPointRef in RteBswEventToTaskMapping is set. RteBswUsedOsSchTblExpiryPointRef in RteBswEventToTaskMapping shall not be set.
1087	RteBswProvidedModeGroupRef in RteBswRequiredModeGroupConnection is empty or invalid. RteBswProvidedModeGroupRef in RteBswRequiredModeGroupConnection shall be set with valid ProvidedModeGroup Path.
1088	RteBswRequiredModeGroupRef in RteBswRequiredModeGroupConnection is empty or invalid. RteBswRequiredModeGroupRef in RteBswRequiredModeGroupConnection shall be set with valid RequiredModeGroup Path.
1089	RteBswProvidedModeGrpModInstRef in RteBswRequiredModeGroupConnection is empty or invalid. RteBswProvidedModeGrpModInstRef in RteBswRequiredModeGroupConnection shall be set with valid ProvidedModeGroup Path.
1090	There are multiple SenderRecElementMappings which have same SystemSignalRef in a SenderReceiverToSignalGroupMapping. There shall not be multiple SenderRecElementMappings which have same SystemSignalRef in a SenderReceiverToSignalGroupMapping.
1091	There is no SenderRecElementMapping which have SystemSignalRef in a SenderReceiverToSignalGroupMapping. There shall be a SenderRecElementMapping which have SystemSignalRef in a SenderReceiverToSignalGroupMapping.
1092	OsApplicationCoreAssignment in OsApplication is bigger than or equal to OsNumberOfCores in OsOs. OsApplicationCoreAssignment in OsApplication shall be less than OsNumberOfCores in OsOs.
1093	RteBswUsedOsSchTblExpiryPointRef in RteBswEventToTaskMapping is empty or invalid. RteBswUsedOsSchTblExpiryPointRef in RteBswEventToTaskMapping shall be set with valid OsScheduleTableExpiryPoint Path.
1094	Some VariableDataPrototypes that are connected to the same sender, or connected to the same receiver, have different init values

	All VariableDataPrototype that are connected to the same sender, or connected to the same receiver, must have identical init values
1095	Some VariableDataPrototypes that are mapped to the same ComSignal/ComSignalGroup, have different init values All VariableDataPrototype that are mapped to the same ComSignal/ComSignalGroup, must have identical init values.
1099	TransmissionAcknowledge is used even though multiple sender is mapped to same ComSignal/ComSignalGroup. TransmissionAcknowledge shall not be used if multiple sender is mapped to same ComSignal/ComSignalGroup
1100	There are multiple RunnableEntities which have VariableAccesses to same PortPrototype and VariableDataPrototype if TransmissionAcknowledge is set. There shall be only one RunnableEntity which have VariableAccesses to the same PortPrototype and VariableDataPrototype if TransmissionAcknowledge is set.
1113	There are multiple ProvidedEntries which have the same BswModuleEntryRef in BswModuleDescription There shall not be multiple ProvidedEntries which have same BswModuleEntryRef in BswModuleDescription
1114	Module which BswModuleDependency refers to and whose Id is does not exist. Module which BswModuleDependency refers to and whose Id is shall exist.
2000	The attribute category of EndToEndDescription can have the following values: NONE, PROFILE_01, PROFILE_02. Check wheter the attribute category of EndToEndDescription is configured and is either NONE, PROFILE_01, or PROFILE_02.
2001	In PROFILE_01, the applicable range of values for counterOffset is [0 .. 65535]. For the value of this attribute the constraint value mod 4 = 0 applies. Check wheter the attribute counterOffset of EndToEndDescription is configured, is [0 .. 65535], and mod 4 = 0.
2002	In PROFILE_01, the applicable range of values for crcOffset is [0 .. 65535]. For the value of this attribute the constraint value mod 8 = 0 applies. Check wheter the attribute crcOffset of EndToEndDescription is configured, is [0 .. 65535], and mod 8 = 0.
2003	In PROFILE_01, the applicable range of values for dataIdMode is [0 .. 2 Check wheter the attribute dataIdMode of EndToEndDescription is configured and is [0 .. 2
2004	In PROFILE_01, there shall be only one element in the set of dataIds and in PROFILE_02, there shall be exactly ordered 16 elements in the set of dataIds. Check wheter the attribute dataIds of EndToEndDescription is configured and the number of elements is correct.
2005	In PROFILE_01, the applicable range of values for the element of dataIds is [0 .. 65535] and in PROFILE_02, the applicable range of values for each element of dataIds is [0 .. 255].

	Check wheter the value for each element of datalds is configured and the range is correct.
2006	In PROFILE_01, the applicable range of values for dataLength is [0 .. 240] and in PROFILE_02, the applicable range of values for dataLength is [0 .. 65535]. For the value of this attribute the constraint value mod 8 = 0 applies Check wheter the value for dataLength is configured and the range is correct.
2007	In PROFILE_01, the applicable range of values for maxDeltaCounterInit is [0 .. 14] and in PROFILE_02, the applicable range of values for maxDeltaCounterInit is [0 .. 15]. Check wheter the range of the value for maxDeltaCounterInit is correct.
2008	The ISignalGroupRef of EndToEndProtectionISignalIPdu is not configured or invalid. Check whether the ISignalGroupRef of EndToEndProtectionISignalIPdu is configured or the referenced value is correct.
2009	The ISignalIPduRef of EndToEndProtectionISignalIPdu is not configured or invalid. Check whether the ISignalIPduRef of EndToEndProtectionISignalIPdu is configured or the referenced value is correct.
2011	If the E2E Library is invoked at the level of Data Elements, then a Data Element shall either map to a local intra-ECU communication (without COM involvement) or shall map to a COM I-PDU, but shall not map to both at the same time. Remove either intra-ECU or inter-ECU connections.
2012	If the E2E Library is invoked at the level of Data Elements (e.g. from SW-Cs or from E2E Protection Wrapper), then the communication shall be an explicit sender-receiver communication, in 1:1 and 1:N multiplicities. Remove either N:1 connction or turn off end-to-end protection feature.
2013	A given I-PDU , shall not be at the same protected by means of COM E2E callouts (through association with ISignalIPdu) and by means of E2E Protection Wrapper (through association with E2E Protection Wrapper. Remove either EndToEndProtectionISignalIPdu or EndToEndProtectionVariablePrototype.
2014	The ContextComponentRef in VariableDataPrototypeInSystemInstanceRef is not configured or invalid. Check whether the ContextComponentRef in VariableDataPrototypeInSystemInstanceRef is configured or the referenced value is correct.
2015	The ContextPortRef in VariableDataPrototypeInSystemInstanceRef is not configured or invalid. Check whether the ContextPortRef in VariableDataPrototypeInSystemInstanceRef is configured or the referenced value is correct.
2016	The TargetDataPrototypeRef in VariableDataPrototypeInSystemInstanceRef is not configured or invalid. Check whether the TargetDataPrototypeRef in VariableDataPrototypeInSystemInstanceRef is configured or the referenced value is correct.
9018	Inter Ecu Communication through Inter Partition is not supported.
9019	DataReadAccess/DataWriteAccess is set in a RunnableEntity activated by a direct function call.

9020	Rte does not support that RteEvents/BswEvents except for TimingEvent in an ExtendedTask reference the same OsEvent.

※ Regarding Error 45

Parameter name	Container
Symbol	RunnableEntity
Native Declaration	SwBaseType
Period	BswTimingEvent
Behaviour Ref	BswImplementation
ImplEntryRef	BSW SchedulabelEntity
BswEveRef	BswEventToTaskMapping
BswImplRef	BswModuleInstance
Period	Timing Event
EventRef	EventToTaskMapping
OsAlarmCntrRef	OsAlarm
OsAlarmActivationTaskRef	OsAlarmActivateTask
OsAlarmSetEventTaskRef	OsAlarmEvent
OsAlarmSetEventRef	OsAlarmEvent
OsSecPerTick	OsCounter

8. Apendix

This Chapter describes considerations when designing an application using RTE, and includes tips that can be used during design.

However, among the contents below, if it is mentioned that there is a restriction on use in Limitation and Deviation for the current RTE Module version, the related information (in 4.5) supersedes what's mentioned in this Chapter.

8.1 Interrupt Decoupling Guide

8.1.1 General Description

8.1.1.1 Scope

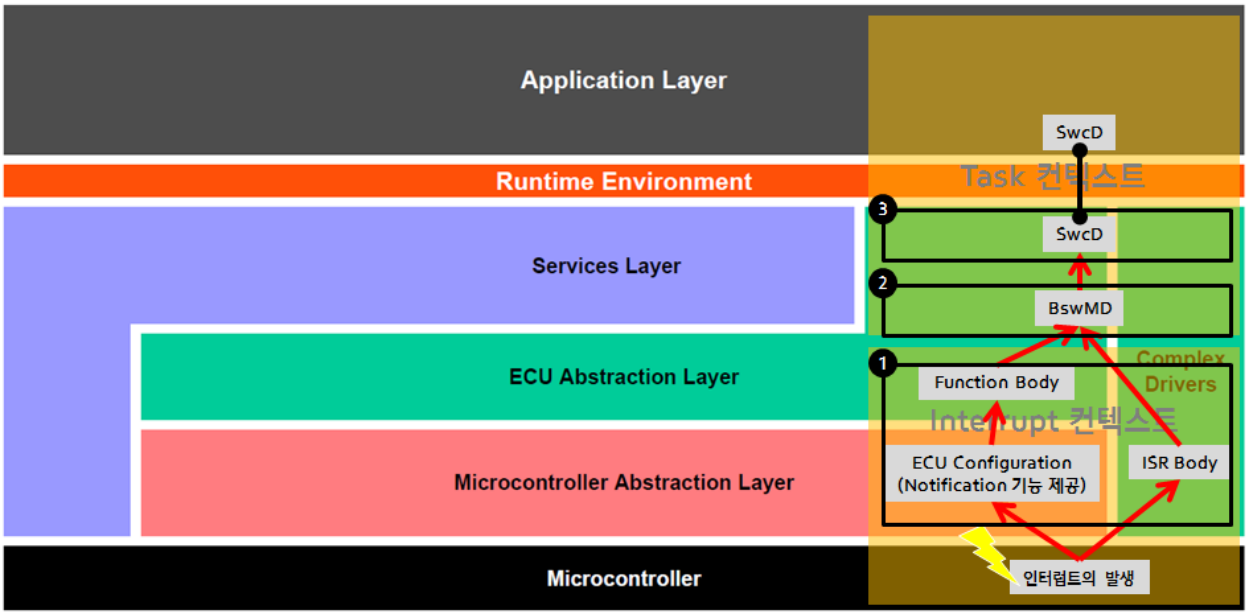
Decoupling method of Interrupt which is provided by the platform is described.

8.1.1.2 Overall Description

In general, Interrupt Service Routine (ISR) is run when Interrupt occurs. ISR is provided by MicroController Abstraction Layer (MCAL) as a Notification Function or can be implemented. At this time, since operation of ISR can affect the entire system, it should be designed so that it performs the minimum role and the rest of operations is performed through decoupling.

The Interrupt processing methods can be classified as follows, and details are described in each chapter.

1. Interrupt Service Routine: The case of performing all actions in Interrupt context. It includes executing the notification function provided by MCAL. (8.1.1.2.1)
2. Decoupling on BSW level: Interrupt is decoupled and run within the BSW (including CDD). (8.1.1.2.2)
3. Communication with ASW: Interrupt is decoupled and needs communication with ASW through Rte during the run in the BSW. (8.1.1.2.3)



8.1.1.2.1 Interrupt Service Routine

If all actions are performed in Interrupt context, separate Software Component Description or Basic Software Module Description is not required as there is no action through Rte.
ISR can be directly implemented through codes or the relevant function can be implemented by configuring the notification function in MCAL.

See the manuals on individual module or Os of MCAL for more information.

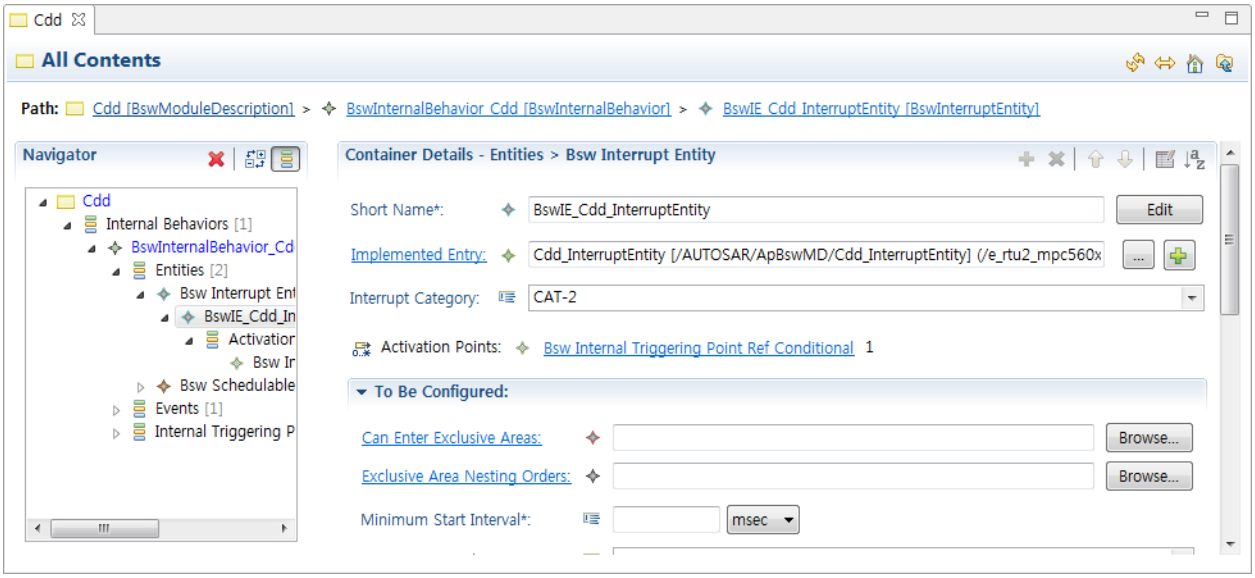
8.1.1.2.2 Decoupling on BSW level

Interrupt is decoupled at the BSW level by default.
In order to decouple the interrupt at BSW level, design through the Basic Software Module Description (BswMD) in each BSW module (including CDD).
Decoupling is possible through Internal Trigger from Interrupt Entity to Schedulable Entity in the Basic Software Module Description.

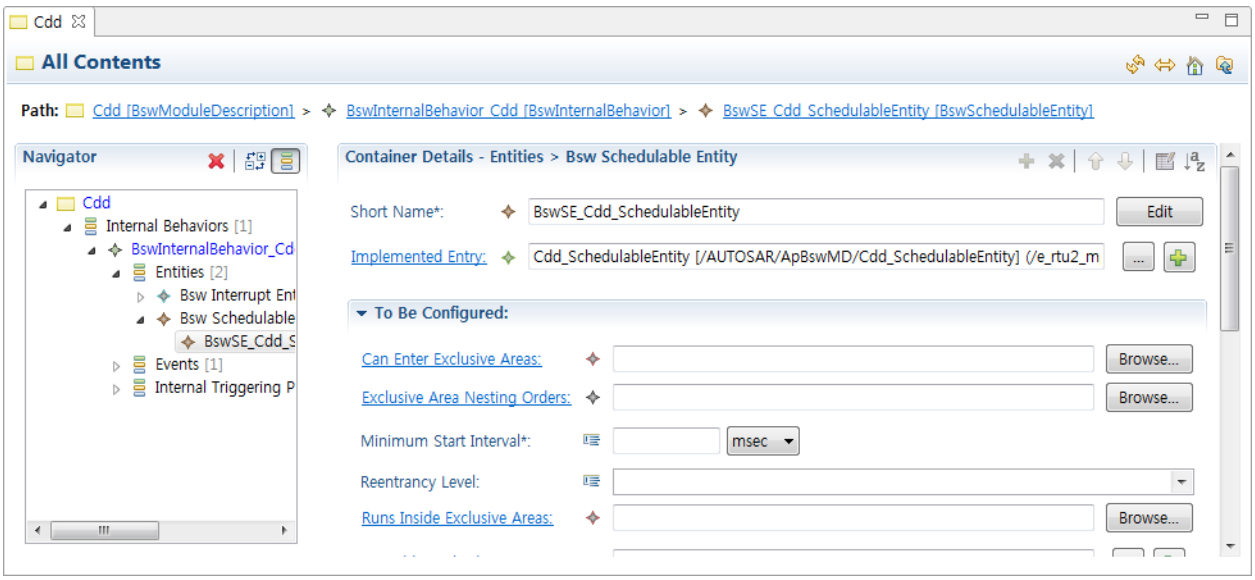
Methods and procedures are as follows.

1. Basic Software Module Description

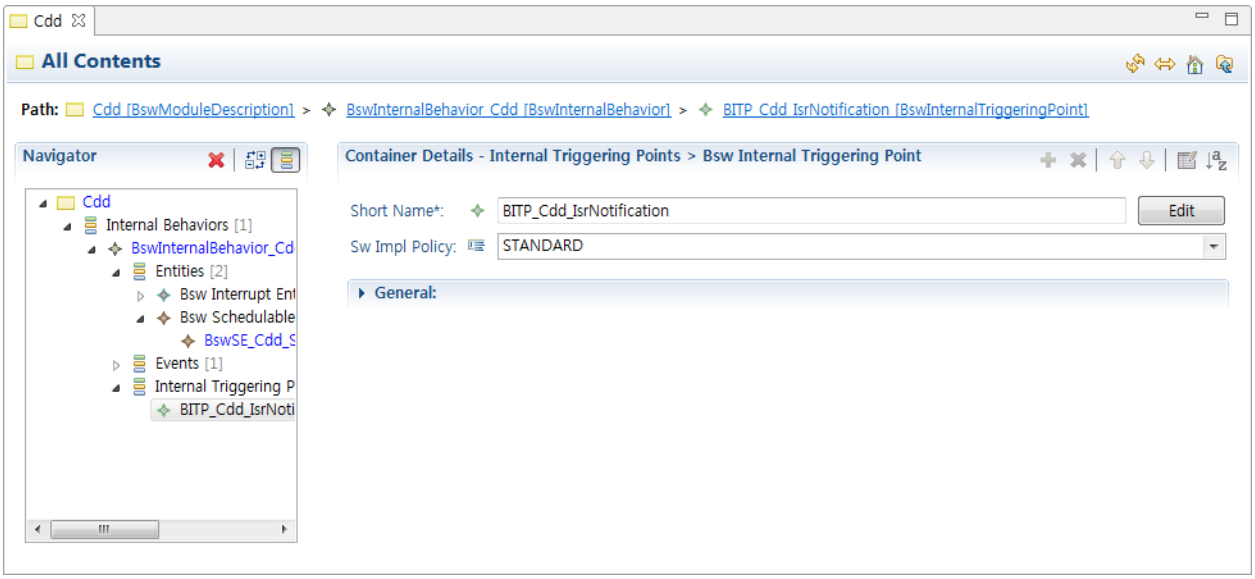
   A. Add Bsw Interrupt Entity: Create Interrupt Entity for the notification function registered in MCAL or directly implemented ISR in the BswModuleDescription.



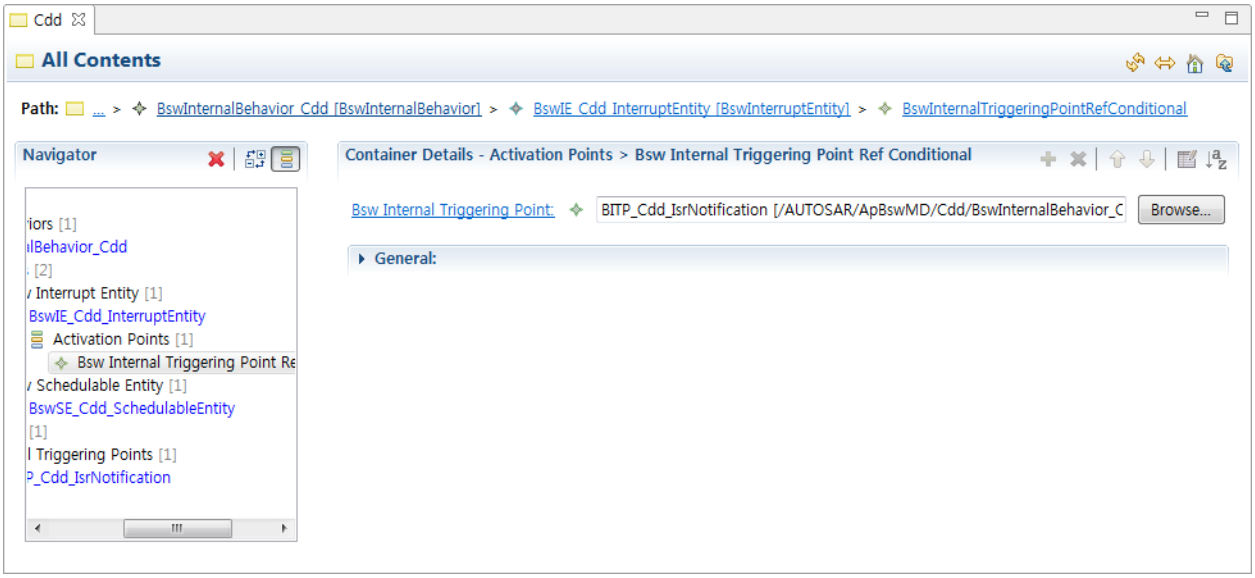
B. Add Bsw Schedulable Entity: Create SchedulableEntity to be run when Interrupt is decoupled in the BswModuleDescription.



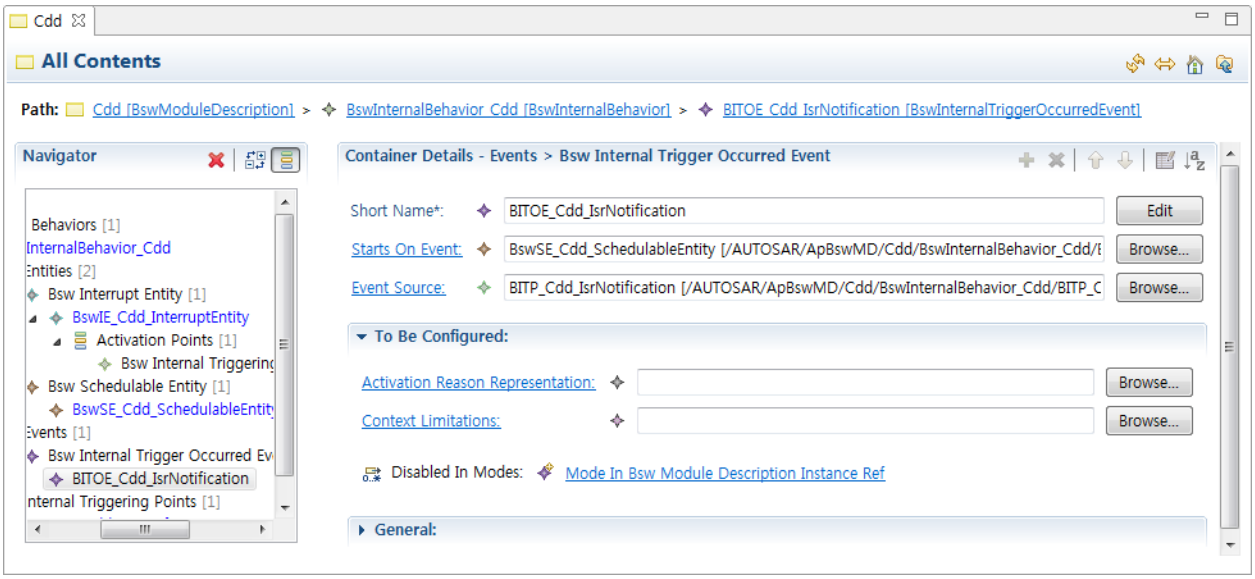
C. Add Internal Triggering Point: Create Internal Triggering Point to run Interrupt Decoupling in the BswModuleDescription.



D. Add Activation Point (Bsw Internal Triggering Point Ref Conditional) in the Bsw Interrupt Entity: Create Activation Point (Bsw Internal Triggering Point Ref Conditional) to start the decoupling in Interrupt.

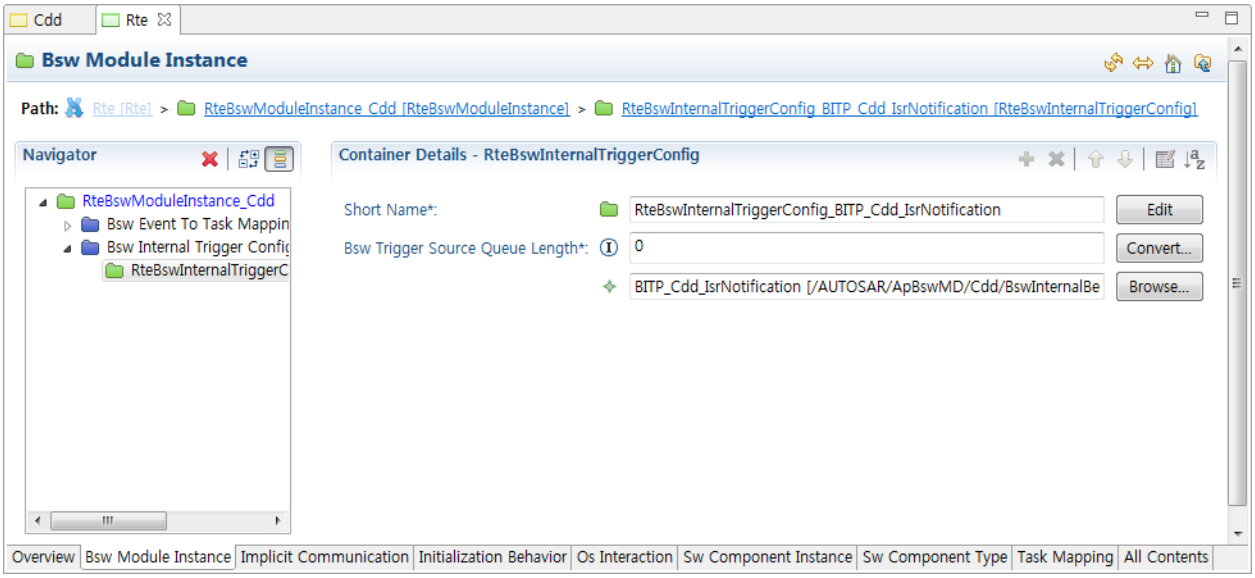


E. Add Bsw Internal Trigger Occurred Event: When Internal Trigger is executed for decoupling, Bsw Internal Trigger Occurred Event is created to call Schedulable Entity.

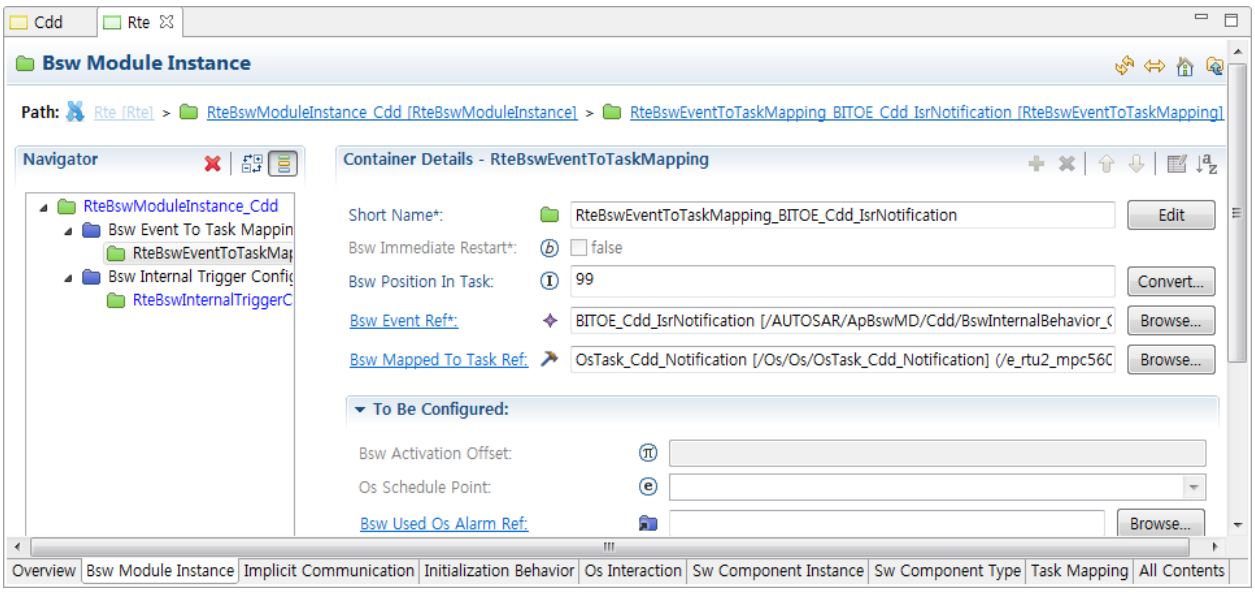


2. Ecu Configuration (Rte)

A. Add RteBswInternalTriggerConfig: Create RteBswInternalTriggerConfig in RteBswModuleInstance of the BSW module (including CDD) in the Ecu Configuration (Rte) for the Internal Triggering Point created in 1.C.



B. Add RteBswEventToTaskMapping: Create RteEventToTaskMapping for the BswInternalTriggerOccuredEvent created in 1.E.



3. Code

A. InterruptEntity: To start decoupling in the notification function or ISR, call SchM_ActMainFunction API (for more details, see 6.3.3.9).

```
#include “SchM_Cdd.h”

void Cdd_Isr (void)
{
  …
  SchM_ActMainFunction_Cdd_BITP_Cdd_IsrNotification();
  …
}
```

8.1.1.2.3 Communication with ASW

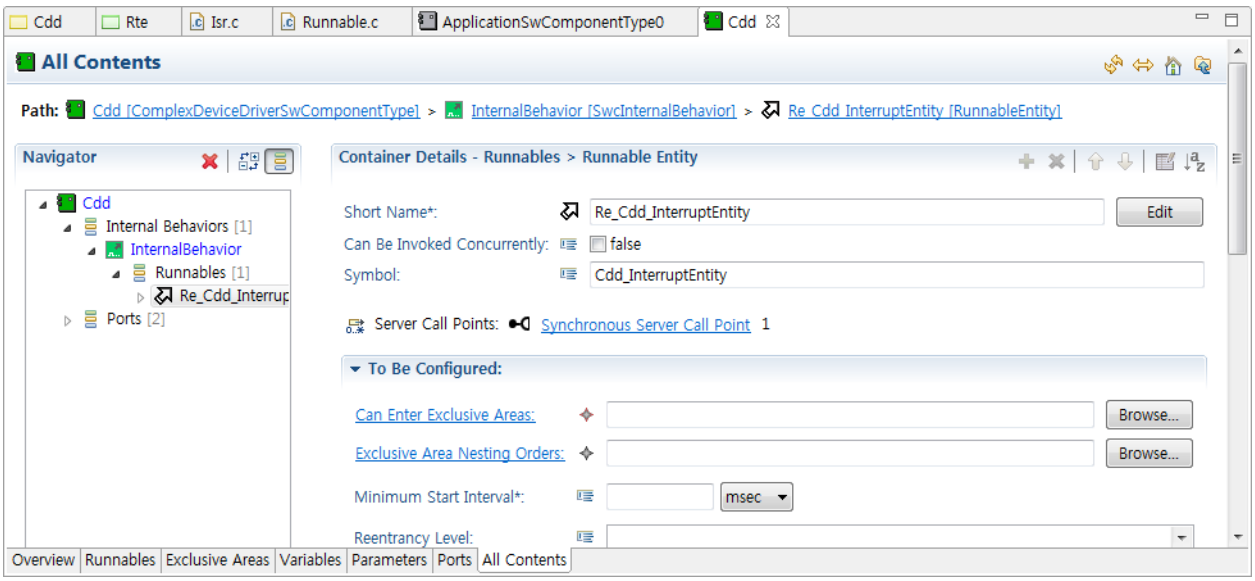
Communication between ASW of Rte and the BSW module (including CDD) can be done through Software Component Description for the BSW module.

Design the software component for the Basic Software Module that decouples the interrupt through 8.1.1.2.2 and communication between the relevant software component and the ASW software component.

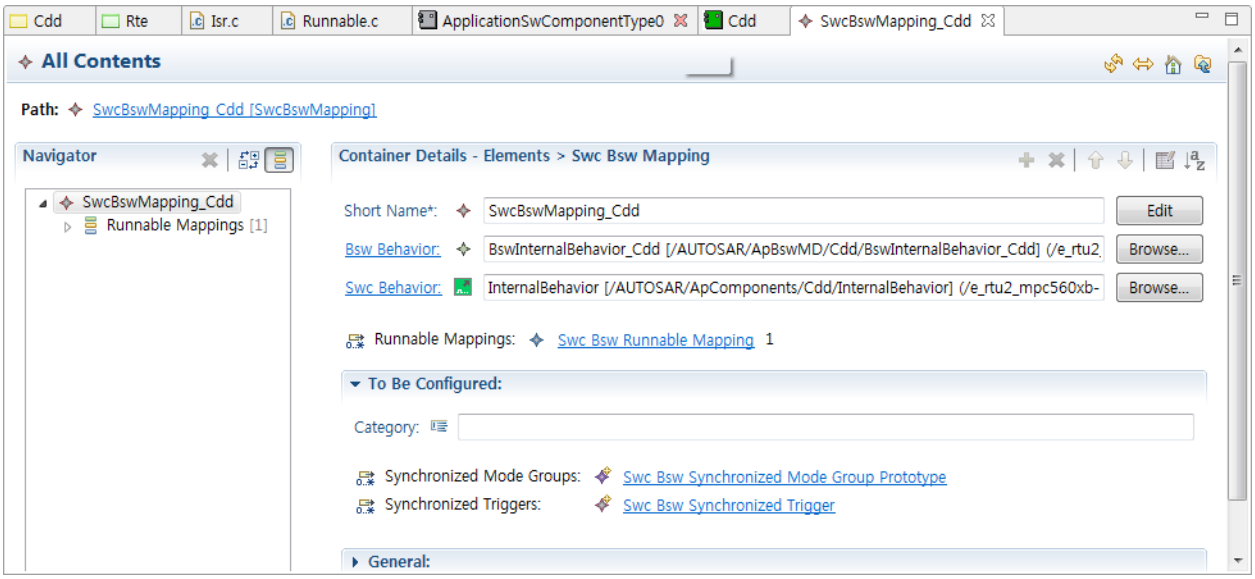
Software Component for the Basic Software Module can be designated through SwcBswMapping.

Methods and procedures are as follows.

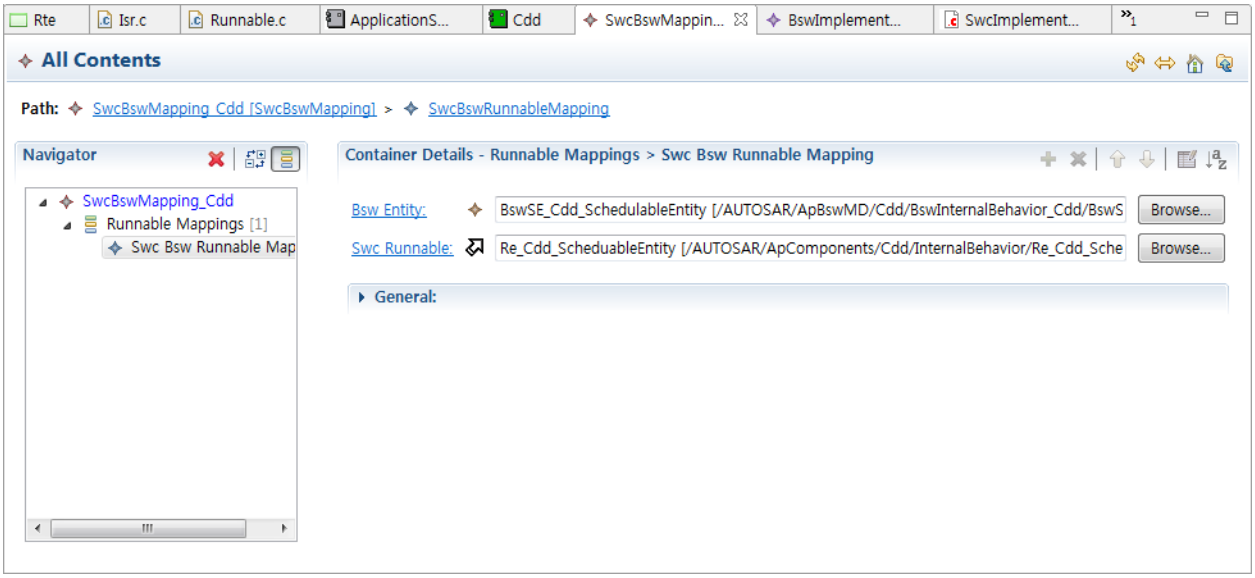
1. Create RunnableEntity of the same name for SchedulableEntity which is run after interrupt decoupling in the SW-C for the BSW module (including CDD) in 8.1.1.2.2.



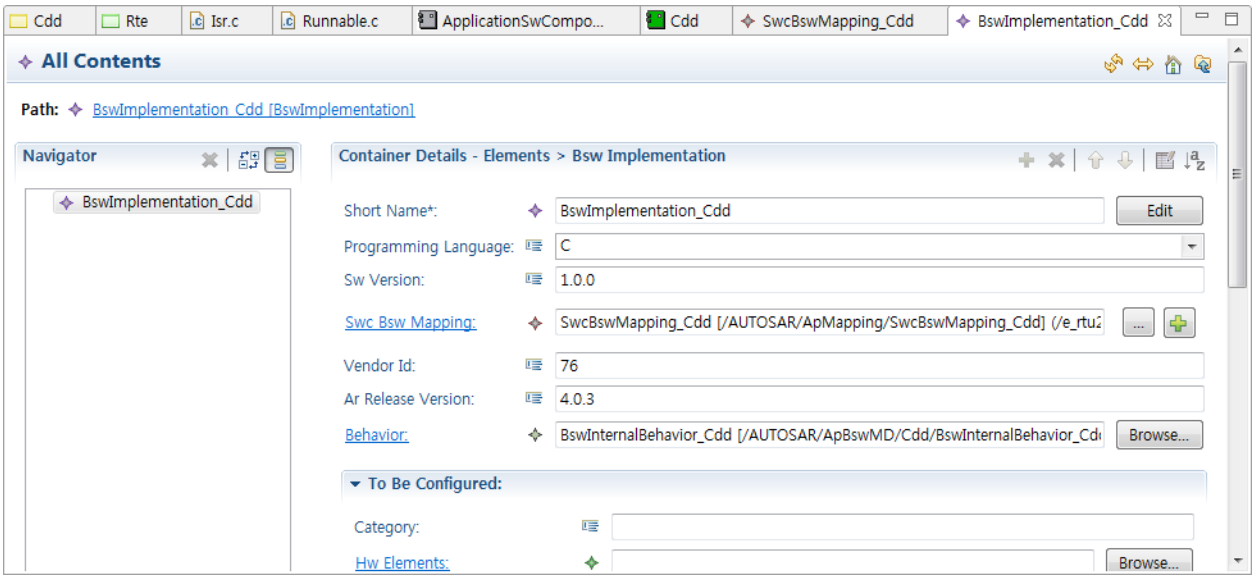
2. Create SwcBswMapping in ARPackage.

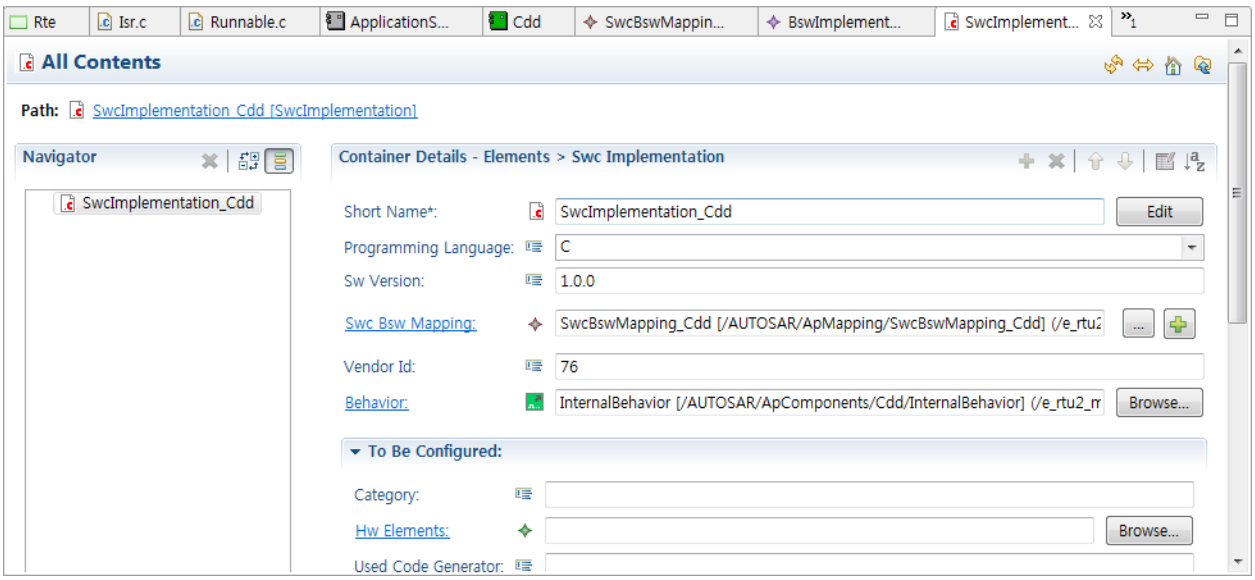


3. Create SwcBswRunnableMapping in SwcBswMapping.



4. Add SwcBswMappingRef to BswImplementation and SwcImplementation.





5. Code: After setting the communication (Sender-Receiver communication, etc.) in the Software Component Description for the BSW module (including CDD), write code to communicate with ASW using Rte API in the code of Schedulable Entity (Runnable Entity).

```
#include "Rte_Cdd.h"

void Cdd_ScheduableEntity (void)
{
  ...
  Rte_Send_...(...);
  ...
}
```

8.1.2 Specific Description depending on MCU

8.1.2.1 Scope

This section describes special notes resulted from Target MCU characteristics in addition to General Description (Chapter 8.1.1).

8.1.2.2 Bolero Family (MPC560x)

When developing an ECU based on the Bolero MCU, the provided AUTOSAR OS does not support multiple activation of task. (I.e. Duplicate Activation feature of a single OsTask is not provided (meaning the Task's Ready Queue = 1) Therefore, multiple activation error can take place, calling the Error Hook function.

8.2 Execution of Runnable mapped in Non-Periodic OsTask

8.2.1 General Description

8.2.1.1 Scope

This section describes considerations for the design through the non-periodic OS task-based Runnable that can be applied when designing AUTOSAR-based SW.

8.2.1.2 Non Periodic OsTask and Runnable

In AUTOSAR, it is possible to design a system based on non-periodic RTE Events. For example, in a specific SWC, it is possible to request run of Runnable in other SWC (i.e. SINK Runnable) through Rte_Trigger Interface. At this time in RTE, the OsTask to which the relevant Runnable is mapped is activated, and the code is generated so that the related Runnable can be run.

8.2.1.2.1 Basic principles

Non-periodic RTE Events can be implemented based on Basic/Extended OsTask, and this will be determined by the RTE configuration.

8.2.1.2.2 Non Periodic OsTask in Basic and Extended OsTask

Basically, in the case of RTE Event execution, there is no difference in Runnable execution according to Basic/Extended OsTask settings. In the case of RTE, to run the relevant Runnable, the OS-provided API will be used. In doing so, the necessary OsTask will be changed to Ready state. Then it runs Runnable required for relevant OsTask.

However, there is a difference in the operating mechanism inside the OS for executing Runnable based on Basic OsTask and Extended OsTask.

How to process Runnable based on Basic OsTask

1) RTE requests the OS to run a specific OsTask through ActivteTask (Event A).
2) At this time, if the OsTask can be executed, the relevant task will be run but if not, it waits for its own turn.
3) At this time, if another OsTask requests additional execution for the OsTask requested in 1) (Event B), Multiple Activation Error may occur depending on the queue setting for this

OsTask.

4) In the situation of 3) (the OsTask is set to Queue > 2, so multiple activation errors do not occur), when the actual execution point is reached, this OsTask is activated twice at the moment. At this time, Event A and Event B are sequentially executed (depends on PositionIntask) at the first OsTask execution time, and at the second OsTask execution time since two events have already been performed the OsTask is terminated without executing Runnable by a specific event.

How to process Runnable based on Extended OsTask

1) RTE requests the OS to run a specific OsTask through SetEvent (Event A).
2) At this time, if the OsTask can be executed, the relevant task will be waken up from Waiting but if not, it waits for its own turn.
3) At this time, if another OsTask requests additional execution (Event B) for the OsTask requested in 1), the same action as in 2) will be taken.
4) When the requested OsTask reaches the run time, the relevant OsTask transitions from the Waiting state to the Run state, and then sequentially runs the connected Event A and Event B with reference to the requested OsEvent.
   At this time, unlike the Basic OsTask-based Runnable processing method, as the OsTask is not activated twice, the task is run only once and then terminated.

8.2.2 Specific Description depending on MCU

8.2.2.1 Scope

This section describes special notes resulted from characteristics of Target MCU, in addition to General Description (Chapter 8.2.1).

8.2.2.2 Bolero Family (MPC560x)

When developing an ECU based on the Bolero MCU, the provided AUTOSAR OS does not support multiple activation of task. (I.e. Duplicate Activation feature of a single OsTask is not provided (meaning the Task's Ready Queue = 1)

8.3 Configuration Guide for Client-Server Communication

8.3.1 General Description

Client-server communication is a communication method in which the server runs a service according to the client request and the client responds with the result.

8.3.1.1 Scope

This section describes operation of the client-server communication and how to set up related Ecu Configuration.

8.3.1.2 Synchronous Client-Server Communication

Synchronous client-server communication is a method in which Request and Response operate simultaneously in Rte_Call() API.

If the application calls Rte_Call()API,

1. execution of the server runnable is requested in Rte_Call() API, and wait for server response.
2. The server runs the service at the request of the client.
3. After the server run is completed, the Rte_Call() API in waiting receives the result and sends it to application.

Synchronous client and server communication has two methods: Direct Function Call and Task. (Rte events are basically mapped to Task through RteEventToTaskMapping, and OperationInvokedEvent for the client-server communication is also based on the Task method, but if simultaneous run is possible (CanbeInvokedConcurrently is true), the direct function call method without mapping to task can be used).

8.3.1.2.1 Direct Function Call

The server is not executed in a separate task, but it is performed in the form of directly calling the server Runnable from the Rte_Call() API of the client.

In general

1. the Runnable corresponding to server sets CanBeInvokedConcurrently to true.
2. For OperationInvokedEvent that runs the Runnable, if RteEventToTaskMapping does not set RteMappedToTaskRef

code will be generated in the form of Direct Function Call.

Rte_Call() API needs to wait for response in the above three steps of "request-service run-response". Since the direct function call method directly executes the server runnable with a function call within the Rte_Call() API, it waits until the server runnable is completed by the call stack. Once the server runnable is returned, the rest of actions are run and the result is sent to application.

The server's runnable runs in the client's task context, so it can be run in multiple contexts at the same time. Also, users cannot use features such as queues and timeouts.

8.3.1.2.2 Task Activation

This is how the server is run in a separate task. Like Direct Function Call, it runs in three steps: request-service run-response.

Unlike the Direct Function Call method, the Task Activation method requests service run in the form of activating the task to which the server runnable belongs to run the server runnable within the Rte_Call() API. Through this, the server's Runnable cannot be run concurrently, and a queue is used when there are multiple requests.

However, since the client (Rte_Call() API) and the server (server runnable) are performed in separate task contexts, Rte_Call() may return prior to the run of the server task depending on the priority. Therefore, after the Rte_Call() API requests the server to run the service it is necessary to set (priority and Wait Point settings) to wait for the server to complete the execution.

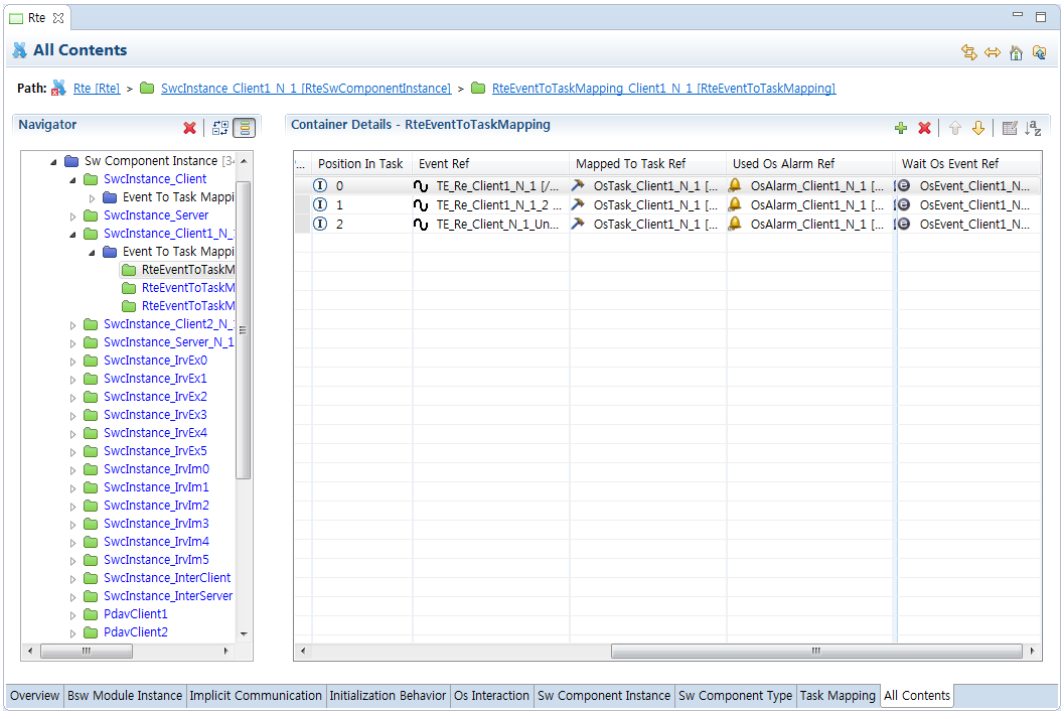
1. Wait Point

   Define the wait point within Rte_Call() API to wait for server response. Wait Point uses Event

of Os. After requesting the server to run the service (ActivateTask() / SetEvent()), it waits for the server's EVENT to occur through the WaitEvent() API.

After the server completes the service, it generates an EVENT for the requested client (SetEvent() API) so that the result can be delivered to the correct client after the server is shut down.

Therefore, the event to be used in the client should be configured. At this time, in synchronous client-server communication, there is no need to separately designate the location of the Wait Point as all operations are performed in the Rte_Call() API. (As the location of the Wait Point becomes the synchronous server call point) **the OsEvent used in the client task must be designed through RteWaitOsEventRef**. (RteWaitOsEventRef is in RteEventToTaskMappng, and it is set in RteEventToTaskMapping for events (e.g. Timing Event, etc.) that run RunnableEntity (Client Runnable) where SynchronousServerCallPoint is defined. It is sufficient to specify the same OsEvent for the same task. See the figure: When three Rte events operate in the same task, RteWaitOsEventRef specifies that they are all the same. (The figure below is for reference only and should not be simply applied to the project.)



2. Prioritization

As it operates using the Wait Point as above, **the priority of the task to which the client belongs is set higher than the priority of the task to which the server belongs.**

3. Exception

   For cases where conditions 1 and 2 cannot be satisfied (SC1 support, etc.), a setting that satisfies all three conditions below is allowed. (Synchronous)

   A. Intra-partition communication
   B. The priority of the server is set higher than that of the client.
   C. RteWaitOsEventRef is not configured.

8.3.1.3 Asynchronous Client-Server Communication

In asynchronous client-server communication, request and response are separated, so a service run is requested in the Rte_Call() API and the result is received in the Rte_Result() API.
In application,

1. when Rte_Call() API is called, run of the server runnable is requested in Rte)Call() API.
2. The server runs the service by executing Runnable codes
3. and then Rte_Result() sends the response to application.

At this time, the application can receive a response by calling the Rte_Call() API, running another action, and calling the Rte_Result() API at a desired time.

In the case of asynchronous method, the server's completion of run is sent to the client in the same way as below (Activation of Runnable Entity).

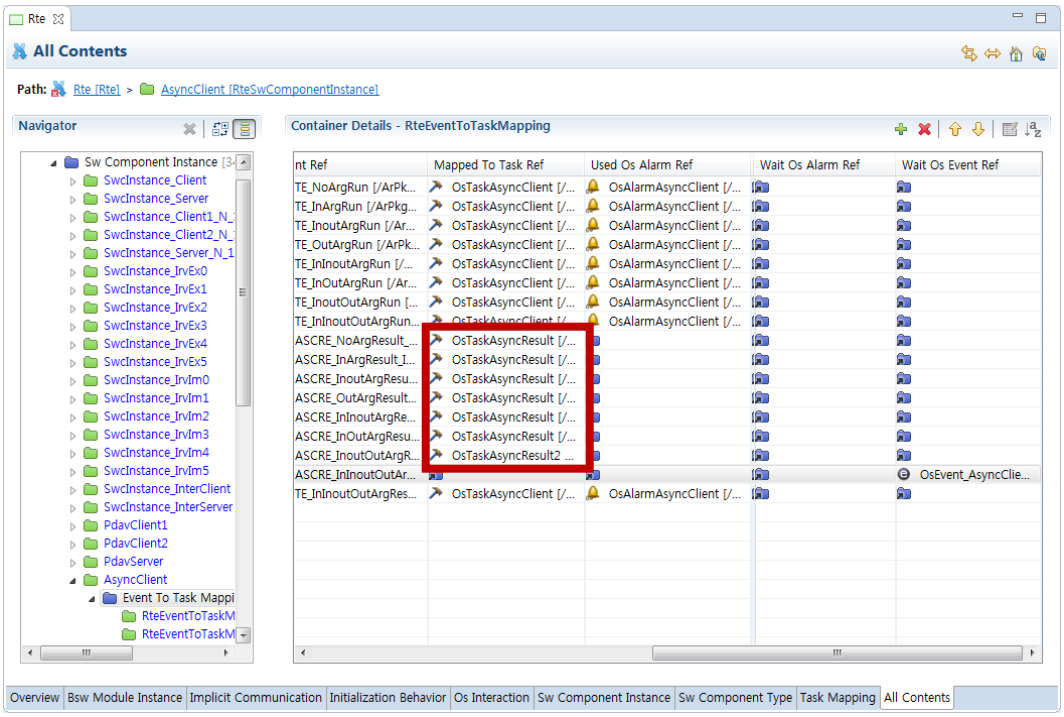
8.3.1.3.1 Activation of Runnable Entity

Once the service run is completed in the server, the designated runnable is activated. After activation of the runnable, response can be received through Rte_Result() API. By specifying StartOnEventRef in AsynchronousServerCallReturnsEvent it can be operated in this way.

1. Asynchronous Server Call Returns Event

   In this method, AsynchronousServerCallReturnsEvent starts Runnable. Therefore, **specify RteMappedToTaskRef in RteEventToTaskMapping** for AsynchronousServerCallReturnEvent to

set task to be run by Runnable. See the Figure (the figure below is for reference only and cannot be simply applied to a project).



2. Prioritization

   Priority of tasks to which Runnable that Rte_Call() API operates belongs > Priority of tasks to which Runnable that Rte_Result() API operates belongs >

   It is based on the priority of the task to which the server Runnable belongs.

8.3.1.4 Inter-Partition Communication

8.3.1.4.1 OsEvent

To use Client-Server Communication between partitions, RteWaitOsEventRef must be set for both Sync/Async. At this time, for OsEvent set with RteWaitOsEventRef, OsEvent with Postfix of PartitionTerminated must be added to ShortName of OsEvent.

For example

1. OsTask: Assume that OsTaskInterClient is the OsTask the Client runs on.
2. OsEvent: Assume that OsEventInterClient is the OsEvent configured as RteWaitOsEventRef. (The OsEventRef in OsTaskInterClient, too, should reference OsEventInterClient.)

☞ Then, an OsEvent of OsEventInterClient**PartitionTerminated** should be added to the Os setting.
(The OsEvent of OsEventInterClient**PartitionTerminated,** too, should be referenced by OsEventRef in OsTaskInterClient.)

This is a necessary setting for a feature that satisfies the conditions required by the AUTOSAR specifications (when the server partition is terminated, the client must be notified immediately) in order to prevent the client from waiting indefinitely when the server partition is terminated. (The client is notified of the termination of the server partition through OsEvent along which PartitionTerminated is tagged)

8.3.1.4.2 Scalability Class

To use Client-Server Communication between partitions, the scalability class must be set as SC3 or SC4 for both Sync/Async. For client-server communication between partitions, the Rte_Call/Rte_Result API must use the Os CheckTaskMemoryAccess() API according to the AUTOSAR specifications, as the CheckTaskMemoryAcces() API is only provided in SC3/4.

8.3.1.5 Clients on Interrupt Level or Invoked by Direct Function Call

If the RunnableEntity (client) running the Rte_Call API is interrupt level or run as a direct function call, the Rte Generator does not print an error message based on priority comparison, and generates a code after printing a warning message (WRN0087). Therefore, if a warning message is printed, check the settings directly to confirm and take action for normal client-server communication.

For example,

1. if the client is run in interrupt (notify the application of the interrupt)
   Use Trigger, etc. to use the client-server communication in RunnableEntity (Task Level) which gets decoupled after interrupt decoupling in BSW, or remove the argument of Operation if interrupt directly uses Rte_Call. Make sure not to configure RteWaitOsEventRef and use only for simple operation of server (application)'s RunnableEntity.
2. If the client runs direct function call

The client finds that tasks that are actually running, compares the priority between each task and the server task, and checks whether it is set according to the priority conditions in 8.3.1.2 and 8.3.1.3.

8.3.1.6 Conclusion

	Synchronous		Asynchronous	
	Direct Function Call	TASK[1)]	Activation of Runnable Entity	Wake Up of Wait Point
RteWaitOsEventRef	Not configured	Client Runnable[2)]	Not configured	ASCRE[3)]
Prioritization	N/A	Client > Sever	Rte_Call, Rte_Result API > server[4)]	

1) Allow setting the priority to Client < Server without setting RteWaitOsEventRef for Intra-Partition Communication for synchronous TASK method.
2) Rte_Call is set in RteEventToTaskMapping for the Rte event running Runnable that operates
3) Set up in RteEventToTaskMapping for Asynchronous Server Call Returns Event
4) Rte_Call and Rte_Result mean the task for which Rte_Call and Rte_Result operate.

8.3.2 Specific Description depending on MCU

8.3.2.1 Scope

This section describes special notes resulted from Target MCU characteristics, in addition to General Description (Chapter 8.3.1).

8.3.2.2 Bolero Family (MPC560x)

When developing an ECU based on the Bolero MCU, the provided AUTOSAR OS does not support multiple activation of task. (I.e. Duplicate Activation feature of a single OsTask is not provided (meaning the Task's Ready Queue = 1)

8.4 Mode Instance Initialization on inter partition mode switching

8.4.1 General Description

8.4.1.1 Scope

This section describes synchronization and initialization for the mode instance in the initialization process of each partition when using the inter-partition mode switch feature.

8.4.1.2 Initialization of Mode Instance in inter partition

When using the mode switch feature between multi-partitions, the mode instance is initialized in the partition where the Mode Manager is located. In the partition where Mode User is located, initialization of Mode Instance is performed differently according to the following two states.

- **The Mode Manager partition is in the starting state**
  See the Mode Instance initialized by the Mode Manager.
- **The Mode Manager partition is in the stopping or restarting state**
  Run self-initialization based on the from configuration for the mode instance.

Runnable execution of OnEntry Event in Initial Mode is performed by itself at the time when the partition where the Mode User is located is started (in SchM_Init or Rte_Start after completing the initialization for the Mode Instance). Follow instance initialization information associated with the two states mentioned above.

8.4.1.2.1 Basic principles

When using the multi-partition-based, mode switch feature, the initialization time of each partition may differ (especially when designing a multi-core-based multi-partition system). When the initialization time of each partition and the initialization sequence between partitions are managed by BswM (or EcuM), (i.e. the initialization sequence between partitions is fixed if the Mode Stack provided by Hyundai AutoEver (EcuM and BswM) and Init Policy are used) design should consider the mode instance processing in each partition according to the partition initialization sequence between Mode Manager/User.

8.4.2 Specific Description depending on MCU

8.4.2.1 Scope

This applies to all MCU in common.

8.5 Precautions on Rte API Usage

8.5.1 General Description

8.5.1.1 Scope

This section includes matters to be noted when designing SW based on the API provided by RTE.

8.5.1.2 Rte API in Runnable

The API provided by RTE is valid only in the Runnable to be used. In other words, when designing SW, the Rte API to be used in a specific Runnable must be reflected in the system (SWC arxml file). RTE creates related Rte API based on the settings to be used in the specific Runnable in the system, and this API must be used within the runnable. If Rte API is used in other runnable than the relevant one, an unwanted error may occur at runtime.

8.5.2 Specific Description depending on MCU

8.5.2.1 Scope

This applies to all MCU in common.

8.6 Synchronized Offset

8.6.1 General Description

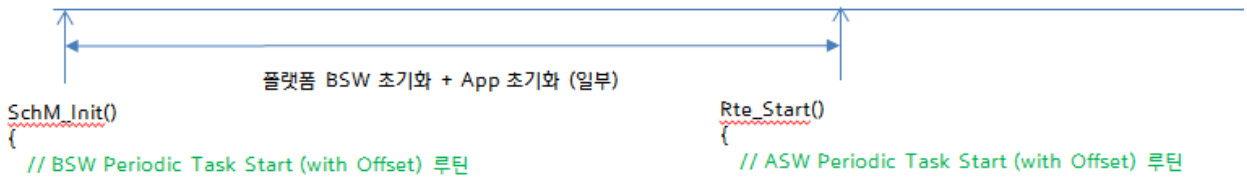
8.6.1.1 Scope

This section describes the necessary settings, contents and cautions when setting the Offset for task run between BSW and ASW based on Synchronized Offset.

8.6.1.2 Description of Synchronized Offset Feature

If synchronization of the Offset of Timing Event that operates using a specific OsCounter-based OsAlarm is required, this feature can be used. This feature is not proposed by the AUTOSAR standard. It is a feature added to enable efficient distribution of run time of OsTask used in BSW and ASW in Hyundai AutoEver.

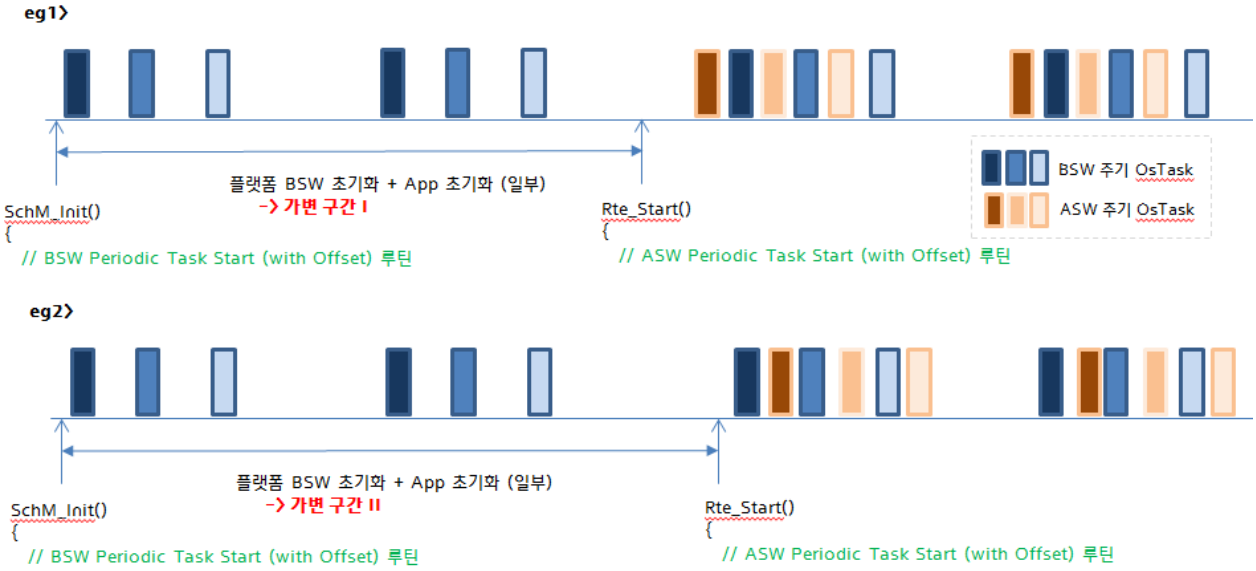
Basically, in AUTOSAR, the cycle OsTask used in BSW and the cycle OsTask start time used in ASW are performed separately within the SchM_Init/Rte_Start function, respectively. Therefore, the synchronization of Offset settings between the cycle OsTask of BSW/ASW is not guaranteed. In other words, settings of the Offset between ASW Tasks and the Offset between BSW Tasks work effectively only between compatible OsTasks (i.e. between ASW cycle OsTask and between BSW cycle OsTask).



In the end, to make the offset settings between ASW cycle OsTasks and BSW cycle OsTasks interoperate with each other, the Synchronized Offset feature must be used. Then OsTask run time can be distributed based on Offset valid for all OsTasks without distinction of BSW and ASW (yet the OsTask activated through OsAlarm should be linked to the same OsCounter.)

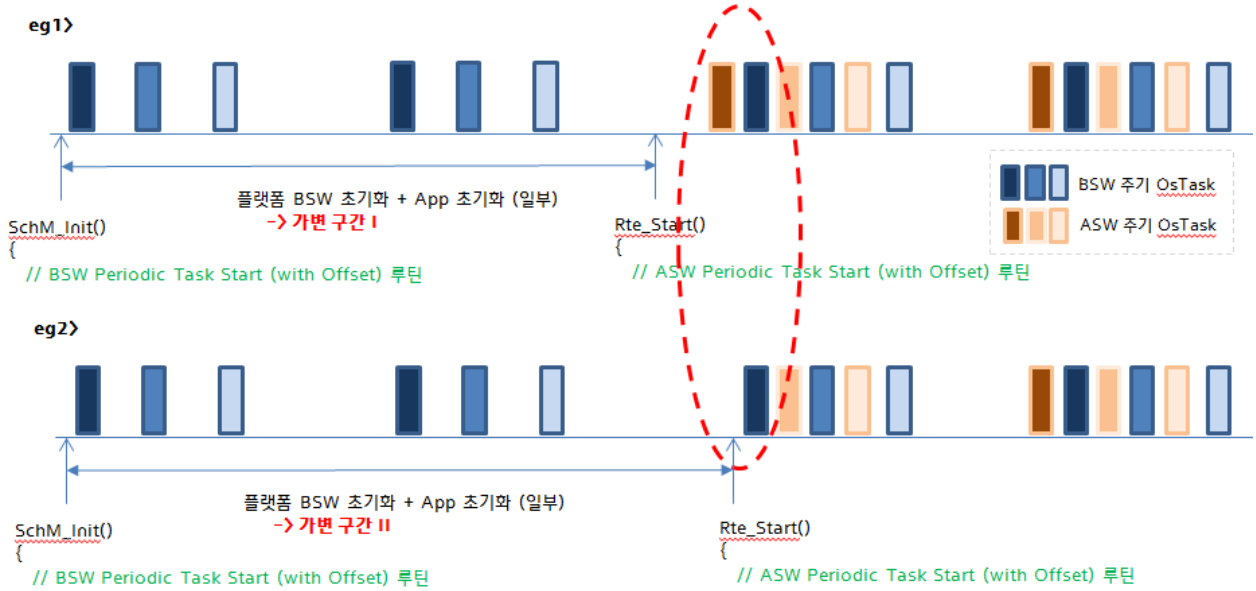
※ Regarding the synchronized offset feature, as the offset information for the Timing Event set in BSW and ASW is not synchronized with each other, it is recommended to use it if users want efficient distribution of resources that run periodic timing events based on the synchronized offset.

If the Synchronized Offset feature is not used, the variable zone (i.e. the zone I and II in the Figure) can determine how the offset between ASW and BSW cycle OsTasks is reflected as shown below.



In the case of using Synchronized Offset feature, as shown in the figure below, although the variable zone (i.e. the zones I and II in the figure) can be changed depending on the situation, the offset between ASW and BSW cycle OsTasks is mutually synchronized and based on a constant offset, OsTask can be run.

Yet task run between the initial ASW and BSW cycle OsTasks may differ depending on the situation. (See the red circle in the figure below)



8.6.1.3 How to Use this Feature

In order to use the synchronized offset feature, OsAlarm information linked to one OsCounter is

required, and the order of each OsAlarm run should be defined (related parameterRteExpectedActivationPosition).

8.6.1.4 Cautions regarding Task Activation Design based on Synchronized Offset Feature

Users should be well aware of the following precautions for using the synchronized offset feature before they design ASW.

① As the order of the first OsAlarm run changes depending on when SchM_Init and Rte_Start are called, the above Offset feature does not guarantee the first run of OsAlarm. (Refer to the figure in 8.6.1.2 (see the red oval part))

② After the first run, OsAlarm occurs according to the cycle, but the order of OsTask run can be changed according to the Os situation at the time of activation immediately after OsAlarm occurs (e.g. OsTask priority, task run time, etc.), the Offset feature does not guarantee the order of OsTask run.

③ To prevent the OsCounter tick value between SchM_Init ~ Rte_Start from exceeding the MaxAllowedValue (Wrap around) MaxAllowedValue should be set to the largest value supported by HW Counter used by OsCounter. (Offset synchronization is not guaranteed after Wrap around)

④ The Offset or cycle of OsAlarm should not exceed (MaxAllowedValue – 1) value of OsCounter.

⑤ There are restrictions in designing depending on the OS Counter type. In some cases, a timing gap may be found in relation to offset synchronization. Please make sure to use it after preliminary review.

Counter type	Impact on SW behavior in Condition	details	Action
SW Counter	NR	Offset gap is not found. Yet as the counter value does not increase within the period of DI~EI[1] it is possible to delay the overall timer by the corresponding time.	NR
HW Counter	Counter 1 Tick value > Rte_DisableInterrupt ~ Rte_EnableInterrupt run time	A gap may be found during offset synchronization as much as the time corresponding to Counter 1 Tick. (For example, if 1 Tick is 1ms and the duration of Rte_DisableInterrupt ~ Rte_EnableInterrupt is 50us, 1ms offset gap may be found)	Please make sure that **"expected offset gap range depending on conditions * 2"** is considered when designing offset for task distribution. That is, if 50us of offset gap is expected, distribution between tasks should be designed with a margin of at least 100us.[2]
	Counter 1 Tick value < Rte_DisableInterrupt	A gap may be found during offset synchronization as much as the time corresponding to	

	~ Rte_EnableInterrupt run time	Rte_DisableInterrupt ~ Rte_EnableInterrupt run time. (For example, if 1 tick is 1us and the duration of Rte_DisableInterrupt ~ Rte_EnableInterrupt is 50us, 50us offset gap may be found)	

* 1) Rte_DisableInterrupt ~ Rte_EnableInterrupt run time: After measuring the run time of the relevant routine in the Rte_Start / SchM_Init function, take the long time as a basis.

Duration of Rte_DisableInterrupt ~ Rte_EnableInterrupt is measured as follows.

※ How to measure the run time

① How to measure the duration of Rte_DisableInterrupt ~ Rte_EnableInterrupt in the Rte_Start function: Measure the run time immediately before Rte_DisableInterrupt(); until immediately after Rte_EnableInterrupt(); within the Rte_Start function (location in the Rte.c file).

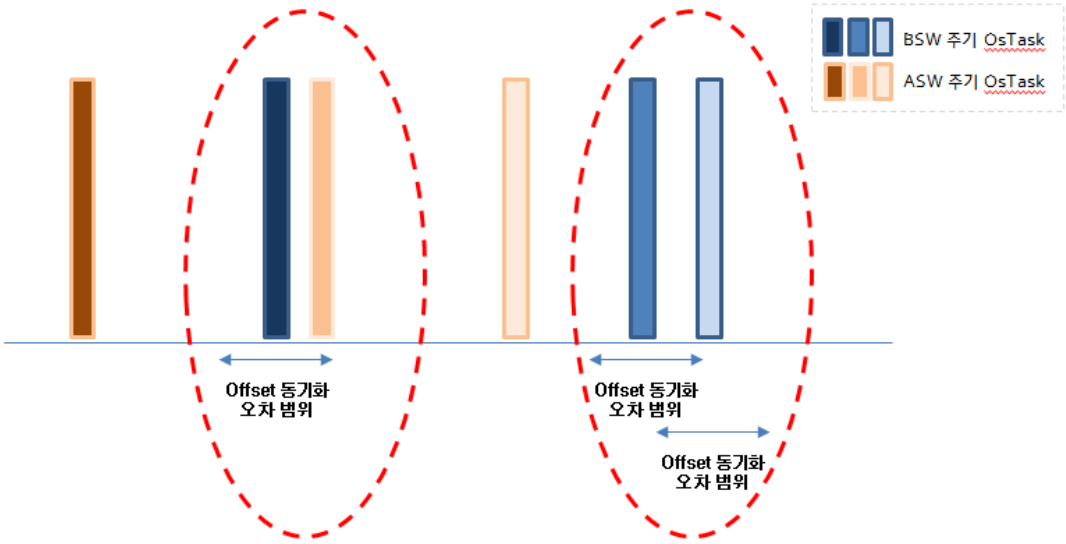
② How to measure the duration of Rte_DisableInterrupt ~ Rte_EnableInterrupt in the SchM_Init function: Measure the run time immediately before Rte_DisableInterrupt(); until immediately after Rte_EnableInterrupt(); within the SchM_Init function (location in the Rte.c file).

③ Final gap in the duration of Rte_DisableInterrupt ~ Rte_EnableInterrupt: The larger of ① and ② will be used as the final gap for the duration of Rte_DisableInterrupt ~ Rte_EnableInterrupt section.

(In the future, it must be assessed together with AutoEver for technical review based on the actual measurements. If the MCU Clock is changed, the run time may be changed. Therefore it is required to check the validity of the Offset after re-measurement.)

** 2) How to interpret the distribution between tasks considering the margin of error: The Offset of OsTask should not be set within the error margin of offset synchronization as in the case of the red dotted oval shape in the figure below.

※ As Synchronized Offset is a feature for sync between BSW <-> ASW, the error margin of offset between ASW <-> ASW and BSW <-> BSW needs not be considered.



8.6.2 Specific Description depending on MCU

8.6.2.1 Scope

This applies to all MCU in common.

8.7 Guide for RteEvent

8.7.1 General Description

8.7.1.1 Scope

This section describes configuration and precautions for Task Mapping and Direct Function Call of the Runnable Entities run by RteEvent.

8.7.1.2 Task Mapping

The Runnable Entity (s) that is (are) run by RteEvent is usually run through the task mapping. In this case, Runnable Entity is run within the task mapped.

8.7.1.3 Direct Function Call

Some RteEvents may be run as Direct Function Call if a specific condition is met in each communication.

8.7.1.3.1 RTE and Basic Software Scheduler and BswExecutionContext

RTE and Basic Software Scheduler support running the triggered ExecutableEntity through a direct function call only in special cases. Nonetheless, it prevents the ExecutableEntity of a specific execution context from calling Triggered ExecutableEntity which requires an execution context with more privileges. The table below shows combinations supported.

BswExecutionContext of the caller	BswExecutionContext of the called				
	task	**interruptCat2**	**interruptCat1**	**hook**	**unspecified**
task	Supported	Supported	Supported		Supported
interruptCat2		Supported	Supported		Supported
interruptCat1			Supported		Supported
hook					
unspecified	Supported				Supported

For example (the 4th Column), if the caller's BswExecutionContext is task, interruptCat2 or interruptCat1, ExecutableEntity call in BswModuleEntiry interruptCat1 BswExecutionContext can be implemented as a direct function call.

It applies to calling of triggered ExecutableEntity based on SchM_Trigger, SchM_ActMain or

Rte_Trigger API, or OnEntry ExecutableEntity, OnTransition ExecutableEntity, OnExit ExecutableEntity or mode switch acknowledge ExecutableEntity based on SchM_Switch or Rte_Switch.
For reference, the execution context of RunnableEntity is considered as a task.

8.7.1.3.2 Client-Server Communication

In client-server communication, RTE generator supports direct function call if at least the following conditions are met.

- The canBeInvokedConcurrently property of server runnable is set to True.
- The client and the server are run in the same partition. In other words, it is intra-partition communication between the client and the server.
- ServerCallPoint is synchronous.
- OperationInvokedEvent is not mapped to OsTask.

8.7.1.3.3 Modes

The RTE generator support direct function call if calling of OnEntry ExecutionEntitys, OnTransition ExecutableEntitys, OnExitExecutableEntitys and mode switch acknowledge ExecutableEntity in Modes meets all of the following conditions.

- Set up asynchronous mode switch behaviour.
- The minimum start distance is not configured in OnEntry ExecutionEntitys, OnTransition ExecutableEntitys, OnExitExecutableEntitys and mode switch acknowledge ExecutableEntity
- The Mode manager and the Mode user in the same partition
- Meet the combination in 8.7.1.3.1

8.7.1.3.4 Trigger Sink

The RTE Generator supports direct function call of Triggered ExecutableEntitys if all of the following conditions are met.

- The ‘minimum start distance’ of Triggered ExecutableEntitys is not defined
- Trigger Sink and Trigger Source in the same partition
- BswTriggerDirectImplementation is not defined
- Meet the combination in 8.7.1.3.1
- Set up Trigger Source other than Queue

8.7.2 Specific Description depending on MCU

8.7.2.1 Scope

This applies to all MCUs in common.