Elektrobit

EB tresos® AutoCore Generic 8 RTE documentation

release notes update for the Rte module

product release 8.8.7

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information.

Table of Contents

1. Overview

This document provides you with the release notes to accompany an update to the `Rte` module. Refer to the changelog Section 2.1, “Change log” for details of changes made for this update.

Release notes details

- EB tresos AutoCore release version: 8.8.7
- EB tresos Studio release version: 29.2.1
- AUTOSAR R4.0 Rev 3
- Build number: B602344

2. Rte module release notes

- AUTOSAR R4.0 Rev 3
- AUTOSAR SWS document version: 3.2.0
- Module version: 6.9.3.B602344
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 6.9.3

2023-01-30

- ASCRTE-6918 Fixed known issue: The Rte verifier falsely reports an error for compatible DataConstraints for connected sender-receiver data elements

Module version 6.10.1

2022-11-25

- Added support for OsSpinlockLockMethod config parameter in mode communication scenario
- ASCRTE-6892 Fixed known issue: Rte might use insufficient transformer buffer array if ComSignalEndianness of Com group signal is "BIG_ENDIAN" and OverrideXfBufferComputation is true
- Updated the Rte Verifier to report an error if a Com/LdCom signal is mapped to multiple provided operation instances (inter-ECU client-server communication)
- ASCRTE-6888 Fixed known issue: Rte_OSShutdownEvent is not calculated by Service Needs Calculator, leading to a compilation error
- ASCRTE-6876 Fixed known issue: Rte Generator fails if initial value is specified for ApplicationRecordDataType with optional elements but without availabilityBitField
- Omitted generation of unused server function declaration in case of inter-partition client (connected ports) without Rte_Call() API

Module version 6.10.0

2022-10-28

- Removed unused Rte_EventBuffer variables which were generated under rare circumstances
- ASCRTE-6875 Fixed known issue: Under certain conditions, Rte_IRead() API doesn't read the data sent by Rte_Write() in 1:n sender-receiver communication

Module version 6.9.2

2022-09-30

- Added full support for inter-partition inter-ECU network representation conversion for sender-receiver communication
- Improved data consistency calculations (locks) by evaluating the execution context of Com reception callbacks
- ASCRTE-6813 Fixed known issue: The Rte Generator fails with an error if a write access exists for an ApplicationRecordDataType mapped to an ImplementationDataType with optional elements
- ASCRTE-6782 Fixed known issue: The Rte starts runnables before the Rte_Start() API is called if the SingleScheduleTablePartitionRef is configured
- Added support for I-PDU based time-out according to SWS_Rte_08103
- Updated the Rte Verifier to report a warning if there is no corresponding RteExclusiveAreaImplementation/RteBswExclusiveAreaImpl container for an ExclusiveArea
- Added support for generating implicit sender-receiver APIs as C functions when enableTakeAddress for the port is set to true

Module version 6.9.1

2022-08-18

- ASCRTE-6789 Fixed known issue: The Rte may generate nested memory sections for mode disabling dependency variable declarations
- Improved declaration of DataHandleVariables by declaring them as extern if they are exported as measurement symbols

Module version 6.9.0

2022-07-22

- ASCRTE-6781 Fixed known issue: Rte Editor may erroneously show error RTEAS_129 if multiple instances of the same BSW module exist

Module version 6.8.4

2022-07-15

- Improved the Rte editor to create RteSwComponentType names based on the type short name
- ASCRTE-6794 Fixed known issue: Rte doesn't generate Rte_IsAvailable() and Rte_SetAvailable() APIs for some optional elements
- Improved the initialization of inter-partition mode communication in order to avoid write accesses to the mode variable from multiple partitions

Module version 6.8.3

2022-06-17

- Added support for ImplementationDataType.typeEmitter set to header file name
- Improved sender-receiver buffer sharing by ignoring different aliveTimeout values in case of pure intra-ECU sender-receiver communication
- Added error code macros that enables an application to check the return value of an API (Infrastructure error, Overlayed error and Application error)
- Added support for Std_TransformerForwardCode in Rte_Write and Rte_Send
- Added support for time-triggered execution of runnnables for InternalTriggerOccurredEvent\BswInternalTriggerOccurredEvent
- Added support for optional upper limit of processed client requests per server executable entity within one task cycle
- ASCRTE-6717 Fixed known issue: Under certain conditions Rte calls the Com APIs with wrong type argument
- ASCRTE-6703 Fixed known issue: The Rte schedule table activation offset uses the default value when Rte Os schedule table activation mechanism is set to Next
- ASCRTE-6729 Fixed known issue: The Rte triggers the DataReceivedEvent twice in a ComSignalGroup reception callback when 1:n sender-receiver inter-ECU communication is used
- Improved the generation of Memory Allocation Keywords for struct and pointer variables by considering platform-specific alignment requirements

Module version 6.8.2

2022-05-20

- Added support for the parameter RteBswExclusiveAreaOsSpinlockRef
- ASCRTE-6695 Fixed known issue: Inter-partition mode-disabling dependencies do not work with implicit sender-receiver communication
- Added support to initialize the Rte implicit write buffers at the beginning of an OsTask body according to SWS_Rte_08418
- ASCRTE-6697 Fixed known issue: The value written with the Rte_IWrite() or Rte_IWriteRef() API might get lost if the runnable is mapped to a task with an explicit schedule point and the implicit data handle variable is shared
- Added support for time-triggered execution of runnable entities for AsynchronousServerCallReturnsEvent
- ASCRTE-6705 Fixed known issue: The Rte generates incomplete memory sections for VFB tracing function declarations
- ASCRTE-6721 Fixed known issue: The Rte may falsely generate a direct access to a data handle buffer in an Rte_Read() API function

Module version 6.8.1

2022-04-08

- Fixed violation for MISRA Rule 08.2: function pointer declarations (P2FUNC) shall contain the parameter names
- Changed error message for an empty or invalid RteBswImplementationRef to a warning
- Improved the verification of ImplementationDataType of category STRUCTURE with optional elements

Module version 6.8.0

2022-03-18

- Sorted task list from the Rte Editor's Event Mapping tab in alphanumerical order

Module version 6.7.3

2022-02-23

- ASCRTE-6621 Fixed known issue: The Rte might not invoke the runnable triggered by a periodic DataReceivedEvent in case of inter-partition n:1 sender-receiver communication

- ASCRTE-6651 Fixed known issue: The Rte_Read() API always returns RTE_E_NEVER_RECEIVED in case of intra-partition inter-ECU sender-receiver communication over LdCom TP and handleNeverReceived set to true
- Improved order and grouping of implicit read and write accesses in task bodies
- ASCRTE-6644 Fixed known issue: Rte Editor does not support all scenarios of the n:1 use case for RteBswRequiredSenderReceiverConnection
- ASCRTE-6627 Fixed known issue: The Rte might define the global copy of an implicit sender-receiver communication in a different partition than the one targeted by the RIPS FlatInstanceDescriptor
- ASCRTE-6654 Fixed known issue: Under certain conditions the Rte Generator ignores the timeout specified for an AsynchronousServerCallPoint with a configured AsynchronousServerCallReturnsEvent and non-blocking Rte_Result() API
- Added support for time-triggered execution of runnable entities and schedulable entities for mode switch events
- Improved the definition of the pre-compiler symbol RTE_RUNNABLEAPI_<rn> macros in order to reduce Rte generation time, the size of Application Header Files, and compile time
- ASCRTE-6637 Fixed known issue: The Rte wrongly handles the VSA OUT argument of the ClientServerOperation of an asynchronous client if RTE_E_HARD_TRANSFORMER_ERROR occurred in the result signal callback
- Added Memory Section Allocation Keyword INIT and CLEARED to distinguish between initialized and unitialized variables

Module version 6.7.2

2022-02-04

- Optimized the error prioritization code for implicit sender-receiver communication
- ASCRTE-6605 Fixed known issue: The Rte Generator might fail with an error if multiple Rte_IRead APIs mapped to the same task directly read from Com and network representation conversion is required
- ASCRTE-6620 Fixed known issue: The non-blocking Rte_Result() API does not return RTE_E_TIMEOUT when it is called with a frequency higher than the configured timeout
- ASCRTE-6581 Fixed known issue: The Rte reports a NullPointerException when multiple ClientServerOperations trigger the same server runnable but have different argument names
- Removed possible empty code blocks for implicit write accesses in task bodies
- Added support for ImplementationDataTypes of category FUNCTION_REFERENCE

Module version 6.7.1

2021-12-15

- ASCRTE-6592 Fixed known issue: The Rte might not invoke the runnable triggered by a periodic DataReceivedEvent in case of inter-partition n:1 sender-receiver communication

Module version 6.7.0

2021-11-19

- ASCRTE-6578 Fixed known issue: The Rte_Call() API might apply an inefficient array copy operation if the server handles multiple operations with array arguments of different lengths
- Improved the memory usage of the Rte Generator for software components with many runnables and API functions
- ASCRTE-6579 Fixed known issue: The Rte triggers only one server runnable if there are two or more server runnables with the same short name that are mapped to the same task

Module version 6.6.3

2021-10-20

- Support for time-triggered execution of runnables for DataSendCompletedEvent
- Added support for the generation of review instructions to easily identify memory writes access for freedom from interference analyses (released)
- Added support for the transmission preparation callback (Rte_COMCbkTxPrep_mn) for inter-partition inter-ECU communication. To activate this feature, set the configuration parameter RteGeneration/SendSignalQueueStrategy to "OnePerMainFunction"
- ASCRTE-6563 Fixed known issue: The Rte may not compile if mixed Rte and BSW events are mapped to a mode user task in case of synchronous inter-partition mode communication
- ASCRTE-6556 Fixed known issue: The Rte Generator may not generate the ISR body for events mapped to ISRs if the Event Mapping tab of the Rte Editor is used
- Improved the software component template code generation by adding the proper CODE memory sections

Module version 6.6.2

2021-09-24

- ASCRTE-6513 Fixed known issue: The Rte_Send/Write API uses a wrong buffer if multiple variable data prototypes are mapped to the same system signal and the senders are not located in the Com partition

- Improved the runtime of the Rte generator for software components with many client-server provide ports
- Added a verifier to reject configuration with 1:n client to server connection
- Added a verifier to report an error if the server is synchronously connected to more than 1 client, and at least 1 client is mapped to a different partition than the server and 1 partition-local client is mapped to the same task as server
- The Rte generator now allows direct server calls where the server runnable has a runsInsideExclusiveArea
- ASCRTE-6500 Fixed known issue: Under certain conditions, an unqueued sender-receiver data element with a structure data type is not received intra-partition inter-ECU if it is directly read from Com
- Added support for Rte_IrvIWriteRef API

Module version 6.6.1

2021-08-20

- The Rte will no longer export the alignment attribute of a memory section if the memoryAllocationKeyword-Policy of the SwAddrMethod is set to ADDR-METHOD-SHORT-NAME
- Avoided exception when there is no valid BswOsTaskRef in case of multi-core client/server communication and Det enabled
- ASCRTE-6533 Fixed known issue: The Rte Bswmd generator may fail with an error on attempt to create a McDataInstance for an ApplicationArrayDataType with a DynamicArraySizeProfile VSA_LINEAR or VSA_-SQUARE

Module version 6.6.0

2021-07-28

- Improved verifier in case of single receive signal port runnable not mapped to a task
- ASCRTE-6501 Fixed known issue: Under certain conditions the tasks in a task chain are not activated if all TimingEvents define the same offset greater than 0
- Improved RTE_516 Verifier Warning to be more descriptive to the user (included the minimum Queue length required)
- Changed verification of the BswModuleDescription.moduleID from mandatory to optional

Module version 6.5.3

2021-07-02

- Improved support for variable size arrays
- ASCRTE-6446 Fixed known issue: If one return signal is used for n:1 inter-ECU client-server communication, the response from the server might get lost
- ASCRTE-6476 Fixed known issue: Missing memory sections for VFB tracing function declarations
- ASCRTE-6473 Fixed known issue: The Rte generator fails with error in case of inter-partition inter-ECU SR communication on receiver side where no runnable has access to the data element and a time-out is configured
- ASCRTE-6485 Fixed known issue: Under certain conditions the Rte falsely applies network representation conversion to inter-partition inter-ECU sender-receiver communication on sender side

Module version 6.5.2

2021-05-28

- ASCRTE-6453 Fixed known issue: The Rte Generator may falsely export elements with duplicate IDs in the RteDataModel.xml
- Added option to always invoke the transformer chain for required sender-receiver data elements if ExecuteDespiteDataUnavailability is true. To enable this option, set the configuration parameter Rte/RteGeneration/ExecuteTransformersDespiteLocalDataQueued (for queued communication) resp. Rte/RteGeneration/ExecuteTransformersDespiteLocalDataUnqueued (for unqueued communication) to true
- Improved sharing of implicit sender-receiver buffers by allocating the buffers in highest to lowest status element order

Module version 6.5.1

2021-04-30

- ASCRTE-6439 Fixed known issue: Data type of Rte variable "Rte_AliveTimeoutCounter_<Sig>" is wrongly generated in case of 1:n sender-receiver communication with different aliveTimeout values
- Improved sharing of implicit sender-receiver buffers between sender and receiver with different compatible data types

Module version 6.5.0

2021-04-16

- Improved: The Rte_Write/Rte_Send API in case of LdCom TP usage (EcuC_TpTransmit) and locked tx buffer

- Improved the Rte generator to trigger autonomous error reaction irrespective of the transformer class
- Improved startup and initialization of NvBlockSwComponents: Restore default data if both ROM and initialization callback are configured
- ASCRTE-6407 Fixed known issue: The Rte_Read() API may copy wrong data when inter-partition-inter-ECU sender-receiver communication is used with transformers and function elision is enabled
- ASCRTE-6410 Fixed known issue: The Rte does not apply data conversion to an implicit intra-ECU sender-receiver communication under certain conditions

Module version 6.4.11

2021-03-12

- ASCRTE-6360 Fixed known issue: The Rte does not trigger DataSendCompletedEvent if an Rte_LdComCbkTpTxConfirmation() callback arrives with a value different from E_OK
- The Rte falsely reports a warning for arrays of anonymous structs
- Added measurement support for Nv RomBlocks
- Added support for retrieving partition context for LdCom on a per-PDU basis
- ASCRTE-6355 Fixed known issue: The Rte_Send() API might not return RTE_E_LIMIT if a queue was full in case of 1:n inter-partition queued sender-receiver communication
- Added support for the generation of review instructions to easily identify memory writes access for freedom from interference analyses (draft)
- Added support to generate graphical representation of the internal Rte generator model

Module version 6.4.10

2021-02-17

- Improved DET checking/reporting code in Enter/Exit APIs
- Added support for mapping of variable data prototypes to parameter data prototypes
- ASCRTE-6342 Fixed known issue: The Rte Generator fails with an error if an ArrayImplementationDataType is used for parameter access in combination with RIPS
- ASCRTE-6336 Fixed known issue: The Rte Generator might ignore system data mappings of receive signals/signalGroups to PRPortPrototypes
- Added support for memory sections with alignment when SWAddrMethod or SwAlignment are not configured
- ASCRTE-6339 Fixed known issue: The Rte Generator fails with an error when generating the Rte_LdComCbkRxIndication for a variable size array

- ASCRTE-6327 Fixed known issue: The Rte Generator is not generating the definition of Rte_TransformerError when contract phase is generated
- Added support for retrieving partition context for COM on a per-PDU basis
- Added support for Rte_IReadRef API

Module version 6.4.9

2021-01-22

- ASCRTE-6291 Fixed known issue: Rte Editor does not support the n:1 use case for RteBswRequiredSenderReceiverConnection

Module version 6.4.8

2020-12-18

- ASCRTE-6292 Fixed known issue: The SvcAs configures duplicate XfrmImplementationMappings in case of inter-ECU client-server communication with multiple clients
- ASCRTE-6296 Fixed known issue: Under certain conditions the Rte verifier does not report an error if an NvBlockSwComponentType is connected to another component type from a different partition
- Improved precision of data conversion formulas with floating-point operands
- Added support for synchronous mode switching procedure over multi partitions/cores

Module version 6.4.7

2020-10-30

- ASCRTE-6235 Fixed known issue: The Rte_IStatus API might be empty in case of implicit 1:n sender-receiver communication with different aliveTimeout values
- Added support for inter-partition direct server call
- ASCRTE-6228 Fixed known issue: The Rte might not activate the runnable triggered by an AsynchronousServerCallReturnsEvent if a timeout occured in case of non-blocking result reception
- ASCRTE-6241 Fixed known issue: The Rte generates an incorrect call to RIPS APIs Rips_Enter()/Rips_Exit() under certain conditions
- ASCRTE-6204 Fixed known issue: The Rte may generate wrong event handling in case a server uses multiple instantiation
- ASCRTE-6223 Fixed known issue: The Rte Generator fails if an AsynchronousServerCallReturnsEvent exists for an unconnected non-blocking AsynchronousServerCallPoint with time-out

- Added support for mapping of ModeSwitchEvents for initialMode switches to RteInitializationRunnableBatch
- Improved: The Rte now aborts the execution of the SvcAs in case of an error
- ASCRTE-6188 Fixed known issue: The Rte may generate extern definitions of the same data handle variable in multiple C-files in case of implicit inter-partition 1:n sender-receiver communication
- Added: The verification of the mapping constraints (component clustering and separation) for EcuExtract
- ASCRTE-6249 Fixed known issue: The Rte might not activate the runnable triggered by ModeSwitchedAckEvent in case of mode user partitions without ModeSwitchEvents
- ASCRTE-6265 Fixed known issue: Under certain conditions OperationInvokedEvents and DataReceivedEvents are not triggered according to their configured RtePeriod/RteBswPeriod
- ASCRTE-6219 Fixed known issue: Under certain conditions the Rte does not trigger ModeSwitchedEvents
- ASCRTE-6277 Fixed known issue: Rte_IRead() API might not read written value in implicit 1:1 sender-receiver communication with transmission acknowledgement enabled

Module version 6.4.6

2020-09-25

- Added support of multiple receivers of SignalGroups with different subelement mappings to primitives
- ASCRTE-6189 Fixed known issue: The Rte Generator fails with an error if a SwServiceArg of category DATA_REFERENCE is used for SchM client-server communication
- Added support for handling the RTE_E_NEVER_RECEIVED error in case of inter-partition communication

Module version 6.4.5

2020-08-28

- ASCRTE-6187 Fixed known issue: The Rte Generator fails with an error if metadata is used for C/S communication over LdCom TP
- Added support of RIPS for ParameterInterfaces
- Removed the verifier which reports a warning when the ComTimeout value is different from the AliveTimeout value

Module version 6.4.4

2020-07-31

- ASCRTE-6171 Fixed known issue: The Rte Generator fails with an error if an application data type which represents a new world variable size array is used for S/R communication
- ASCRTE-6175 Fixed known issue: The Rte rejects configurations if client-server operations are compatible according to TPS_SWCT_01125 and if an ArgumentDataPrototype is typed by an ApplicationArrayDataType
- Added support of multiple receivers with different aliveTimeout values mapped to the same SystemSignal
- ASCRTE-6177 Fixed known issue: The Rte may generate a wrong alignment suffix for the Memory Allocation Keyword if a SwAddrMethod is configured for a VariableDataPrototype typed by an array of struct data type
- ASCRTE-6185 Fixed known issue: The Rte Generator fails with an error if metadata is configured for a PDU and a signal contained in this PDU is sent via S/R communication over LdCom
- Added missing task end/chain hook in task chains according to [SWS_Rte_06032]

Module version 6.4.3

2020-06-29

- Changed linking of Rte symbols from object files to the newly introduced Rte_src.lib library file
- ASCRTE-6106 Fixed known issue: The Rte_Feedback() API never returns a transformer error in case of inter-partition inter-ECU communication with transformers
- ASCRTE-6109 Fixed known issue: The Rte may generate an empty Rte_IStatus() macro if an implicit S/R communication buffer (data handle buffer) is shared
- Added support for initial value calculation with RuleBasedValueSpecification
- ASCRTE-6124 Fixed known issue: The Rte does not enter/exit an exclusive area in the role runsInsideExclusiveArea of a RunnableEntity that is mapped to a BswSchedulableEntity and started by a BswEvent
- ASCRTE-6125 Fixed known issue: An implicit inter-runnable variable is not updated for a RunnableEntity that is mapped to a BswCalledEntity which is started by an BswOperationInvokedEvent and mapped to a task
- Added a verifier to report a warning when the ComTimeout value is different from the AliveTimeout value
- ASCRTE-6086 Fixed known issue: The Rte generator does not consider the value of the parameter RteRipsPluginFillFlushRoutineFncSymbol
- ASCRTE-6123 Fixed known issue: The Rte may falsely generate a direct access to a data handle buffer in an Rte_Write() API function
- ASCRTE-6117 Fixed known issue: The Rte may update a shared sender-receiver data handle buffer from different partitions
- Added support for DataWriteCompletedEvents and Rte_IFeedback API

- ASCRTE-6147 Fixed known issue: The Rte generates an invalid switch statement in the Com task if a data element to signal fan-out is used and the signals are sent on different Com partitions

Module version 6.4.2

2020-05-27

- Improved argument buffering for synchronous client/server communication
- Added support for synchronized triggers
- ASCRTE-6085 Fixed known issue: The Event Mapping tab of the Rte Editor reports a StackOverflowError when a circular client-server call chain is configured
- Added support for variable size array inter-ECU communication without transformers
- ASCRTE-6092 Fixed known issue: The Rte verifier may falsely report an error about a wrong availability-Bitfield array size for structs with optional elements
- Improved: The Rte shall optionally use the (Ld)Com container's length as transformer buffer length
- ASCRTE-6091 Fixed known issue: The Rte_Read() API generates an incorrect call to Com_ReceiveSignal if DirectReadFromCom applies but LdCom is used

Module version 6.4.1

2020-04-24

- Improved verifier SWCUTILS_SYSDVFY_658 to allow blocking data reception of the same required VariableDataPrototype from different runnables which are mapped to the same task
- Added verifier to check transformer buffer sizes against LdCom signal sizes
- ASCRTE-6049 Fixed known issue: The contract phase generation fails with an error if a Bsw module contains a BswInternalTriggeringPoint or an IssuedTrigger
- Added export of internal Rte generator model to XML when RteDataModelExport is enabled
- ASCRTE-6060 Fixed known issue: The Rte generates an unnecessary cast to void* in the IRead() API if RIPS is enabled
- ASCRTE-6064 Fixed known issue: RTE does not generate calls to RIPS APIs IWBufferRef()/IRBufferRef() for implicit S/R communication if RtePluginSupportsIReadIWrite is set to false
- ASCRTE-6043 Fixed known issue: An AsynchronousServerCallReturnsEvent triggers the non-blocking result runnable twice if a timeout is defined for the AsynchronousServerCallPoint
- Improved the Rte generator to verifiy the consistency of PortAPIOptions and PortDefinedArgumentValues if EventInstances are merged

- ASCRTE-6072 Fixed known issue: The Rte invokes configured Rte_Rips_FillFlushRoutines although the parameter RteRipsSupport is set to RTE_RIPS_OFF
- Added verifier to check restriction to explicit sending semantics for the usage of DataServices in the context of a SwcServiceDependency that aggregates DiagnosticValueNeeds that in turn is referenced by a DiagnosticIoControlNeeds

Module version 6.4.0

2020-03-25

- Improved the RTE generator to call the RIPS APIs for unconnected Sender/Receiver ports
- Added verifier check for implicit S/R 1:n communication where receivers are not compatible and mapped to the same FID and RIPS is managed by plugin
- Added support for generating Bitfield Texttable definitions according to the AUTOSAR 4.4 RTE spec

Module version 6.3.14

2020-02-21

- Added support for LdCom without transformers and inter-partition communication over Ioc
- ASCRTE-5943 Fixed known issue: The Rte triggers ModeSwitchedAckEvents at the end of the transition to the initial mode
- Improved performance of queue accesses by caching the queue index in a stack variable. This also fixes a possible compiler warning: "the order of volatile accesses is undefined"
- ASCRTE-5990 Fixed known issue: The Rte software component header might include a wrong RIPS software component header
- ASCRTE-5988 Fixed known issue: The Rte Editor does not display all BswEvents under certain conditions
- ASCRTE-5981 Fixed known issue: The Rte generates an incorrect subElement mapping under certain conditions
- ASCRTE-5967 Fixed known issue: The Rte does not allow mapping of ApplicationRecordDataType to ImplementationDataType with optional elements
- ASCRTE-5992 Fixed known issue: Under certain conditions the Rte falsely skips data conversion for inter-ecu sender-receiver communication
- ASCRTE-5970 Fixed known issue: The Rte triggers ModeSwitchedAckEvents and BswModeSwitchedAckEvents wrongly in case of multiple mode user partitions
- ASCRTE-5950 Fixed known issue: Subsequent calls to LdCom_TpTransmit() fail if an Rte_LdComCbkTpTxConfirmation() callback arrives very shortly after LdCom_TpTransmit()

- Generate execution condition for the A1 mapped timing events to ISR if periods are not equal
- ASCRTE-5978 Fixed known issue: Rte_LdComCbkTriggerTransmit() copies wrong data in case of inter-partition inter-ECU communication with an autonomous error reaction
- ASCRTE-5979 Fixed known issue: An autonomous error reaction may corrupt data of outstanding server responses
- Added support for RIPS for BSW explicit queued sender/receiver communication and exclusive area protection
- ASCRTE-5963 Fixed known issue: The Rte_Main.c does not include Rte_UserDefinedExclusiveArea.h in case multiple ExclusiveAreas are shared across different partitions
- Updated the check for Rte Det error reporting with partitioning and shared Schm exclusive areas
- Added support for memory optimization for zero initial values
- Improved the Rte verifier to report an error if a server RunnableEntity is mapped to a BswSchedulableEntity. It must be mapped to a BswCalledEntity instead
- ASCRTE-6002 Fixed known issue: The Rte might return a wrong server result to a client if all other connected clients use result-free asynchronous client-server communication
- Improved the Rte generator to allow client tasks to have a higher priority than the server task also in basic intra-partition scenarios
- Improved data consistency optimizations and task mapping verifications by considering Os resources of exclusive areas in the runsInsideExclusiveArea role
- ASCRTE-6004 Fixed known issue: The Rte generates duplicate macros in the Rte_Buffers.h for 1:n implicit S/R RIPS communication

Module version 6.3.13

2020-01-24

- ASCRTE-5942 Fixed known issue: Rte generation fails if an ImplementationDataType is used in a PortInterface and has the same name as one of the standard OsTypes
- Improved: The Rte now adds suffixes for float and double literals if initial values are generated but not explicitly configured
- Improved: The Rte Generator now verifies the partition of Tx Com signals against the partition of the Bsw task if no ComTaskConfiguration containers are configured
- ASCRTE-5954 Fixed known issue: The SvcAs fails with an error in case of a SystemSignal-to-ISignal Tx fan-out with LdCom, even if distinct TransformationTechnology elements are used for each ISignal
- ASCRTE-5945 Fixed known issue: The Rte does not ensure data consistency between LdCom_Transmit() and Rte_LdComCbkTriggerTransmit()

Module version 6.3.12

2019-12-06

- ASCRTE-5912 Fixed known issue: The server might not be called synchronously if intra-partition or inter-partition-intra-core client-server communication is used and a category 2 executable is mapped to the server task
- ASCRTE-5903 Fixed known issue: The SvcAs does not create all CustomXf XfrmImplementationMappings for Tx signals if a transformer chain contains multiple TransformationTechnology elements with transformerClass set to custom
- Added support for configurable output directory of Rte_Bswmd.arxml
- Added support for LdCom without transformers
- ASCRTE-5915 Fixed known issue: The Rte may copy wrong data in Rte_LdComCbkCopyTxData()
- ASCRTE-5914 Fixed known issue: The Rte does not lock the LdCom TP transmit buffer sufficiently
- Updated the verifier to report a warning when an ImplementationDatType of category ARRAY contains an ImplementationDataTypeElement of category STRUCTURE or UNION
- ASCRTE-5904 Fixed known issue: The Rte_Receive() API never returns RTE_E_LOST_DATA in case of inter-partition inter-ECU communication with transformers
- Provided ALIGNMENT info for CONST MEMORY-SECTIONs which are generated in Rte_Bswmd.arxml
- Added support for RteCalibrationSwAddrMethodRef for software calibration
- Added support for mapping timing events to ISRs

Module version 6.3.11

2019-11-08

- Add partitioning support for communication timeout with transformers
- Improved lock optimizations for Rte/SchM APIs that are accessed from non-interruptible tasks only

Module version 6.3.10

2019-10-11

- Improved: Define macro of PR port initial value to be generated once
- Changed the parameter RteSwNvRamMappingRef of the container RteNvRamAllocation to optional
- ASCRTE-5826 Fixed known issue: Under certain conditions the Rte may send/receive incorrect data to/from the Com module
- Improved verifier for inter Ecu C/S communication with client runnable entity not mapped to a task

- Added support for TimingEvent.offset
- ASCRTE-5850 Fixed known issue: The SvcAs does not set ComTimeout in case of inter-partition inter-ECU sender-receiver communication
- Added support for overlayed errors in Rte_Read(), Rte_Receive() and Rte_IStatus() APIs (intra-partition intra/inter-ECU sender-receiver communication only)
- ASCRTE-5851 Fixed known issue: The SvcAs does not set ComSignalInitValue, ComSignalDataInvalidValue and ComDataInvalidAction in case of inter-partition inter-ECU sender-receiver communication
- ASCRTE-5848 Fixed known issue: Rte generates incorrect code in the case of multiple internal triggering points with the same short name triggering the same runnable
- ASCRTE-5849 Fixed known issue: Rte generation fails for inter-core asynchronous client-server communication with InterPartitionCommunication set to Mixed
- ASCRTE-5822 Fixed known issue: The generated Rte may not compile in case of implicit mixed n:1 inter/intra-partition S/R communication with transformers
- Added support for Rte(Bsw)ExclusiveAreaOsResourceRef to reference an OsResource in case RteExclusiveAreaImplMechanism is configured to OS_RESOURCE for this ExclusiveArea
- ASCRTE-5845 Fixed known issue: The Rte_IStatus() API does not return any transformer error in case of inter-partition inter-ECU sender-receiver communication
- ASCRTE-5854 Fixed known issue: The Rte generator may falsely report a warning if multiple NonqueuedReceiverComSpec.initValue of the same port reference data elements with different application data types
- Improved the number of enter/exit critical section calls by grouping the SMC channel read/write accesses of the same OS task by their critical section blocks in case of implicit S/R communication
- ASCRTE-5879 Fixed known issue: The Rte Generator may fail if RteDevErrorDetect is true and a software component has multiple ports with the same port interface
- ASCRTE-5855 Fixed known issue: Det error reporting and partitioning may lead to linkage errors about multiple definitions of Rte_Det_IntLockTask
- Improved data consistency optimizations and task mapping verifications by considering internal Os resources

Module version 6.3.9

2019-09-06

- ASCRTE-5791 Fixed known issue: The function declaration of the RunnableEntity triggered by ExternalTriggerOccurredEvent is not visible for the trigger source if direct function call and function elision apply
- ASCRTE-5814 Fixed known issue: The Rte Generator generates a wrong argument cast in direct server call when server and client use different client-server-interfaces

- ASCRTE-5815 Fixed known issue: The Rte Generator falsely reports an error if an ExternalTriggerOccurredEvent is mapped to an OsIsr in a multi-partition environment
- Added support for intra-ECU sender-receiver sub element mapping to a primitive data element
- Added support for mode switch acknowledgement for basic software

Module version 6.3.8

2019-08-09

- Improved: The Rte now supports casting to compatible data types when calling transformers
- Changed the assignment of void pointer arguments to always use TS_MemCpy, e.g. in the Rte_SetMirror and Rte_GetMirror callbacks

Module version 6.3.7

2019-08-06

- Added support for compu methods specified through the physical props of SystemSignal
- ASCRTE-5752 Fixed known issue: The Rte.bmd file contains an invalid destination reference to the NvMBlockDescriptor for parameter RteNvmBlockRef
- ASCRTE-5775 Fixed known issue: The Rte verifier may falsely report an error about a wrong transformer buffer size if GroupSignals are not byte-aligned
- Added inter-partition support for coherency groups
- Fixed violation of MISRA rule 20.7 by using FUNC_P2CONST/FUNC_P2VAR macro instead of nested compiler abstraction
- ASCRTE-5753 Fixed known issue: The Rte may not properly evaluate the SwDataDefProps if multiple ImplementationDataTypes with the same shortName exist
- ASCRTE-5749 Fixed known issue: The Rte does not protect the inter-partition SchM_Switch API against concurrent mode switch notifications under certain conditions
- ASCRTE-5761 Fixed known issue: The Rte Generator may not call the server function with an activating event argument

Module version 6.3.6

2019-07-12

- Added support for Rte(Bsw)UsedOsEventRef to specify which OsEvent is used to activate the OsTask

Module version 6.3.5

2019-06-14

- ASCRTE-5676 Fixed known issue: Data conversion for network representation without an application data type is not performed
- ASCRTE-5689 Fixed known issue: The Rte may not apply data consistency for the readers of inter-core SMC channels where only implicit writers exist
- Improved EB_FAST_LOCK for multi-core set-ups
- Improved separation of the Rte partitions for COM send signals by adding a new configuration parameter (SendSignalQueueStrategy)
- Added support for inter-partition inter-ECU communication timeout. The RTE will write the data and RTE_E_OK in case of successful reception of data and will write RTE_E_MAX_AGE_EXCEEDED as status in case of reception timeout
- ASCRTE-5710 Fixed known issue: The Rte may not apply data consistency for the implicit writers of inter-partition n:1 sender-receiver communication
- Added limited support for Coherency Groups
- ASCRTE-5907 Fixed known issue: The Rte generation may fail in case of inter-core intra-ECU C/S communication where the server runnable is target of multiple OperationInvokedEvents
- Added support for multiple mode user partitions

Module version 6.3.4

2019-05-17

- Improved Rte generator to support arbitrary Os TaskType definitions
- ASCRTE-5649 Fixed known issue: Potential loss of precision for initial/invalid values defined by ApplicationValueSpecifications
- Removed verifier warnings for diverse SwBaseType native declarations of type "long long"
- Improved verifier for BswCalledEntity SwServiceArgs
- ASCRTE-5670 Fixed known issue: The Rte Generator may fail if a received group signal is mapped to a primitive non-queued sender-receiver data element and DirectReadFromCom applies
- Added Support for time-triggered execution of runnables for data received events
- Improved the Rte generator to not throw a warning message in case R-Port of NVBlockSoftwareComponentType is unconnected and do not define an INIT value
- Improved Rte by providing appropriate suffixes for 64-bit integer values
- Implemented timeout monitoring for asynchronous client/server communication with non-blocking result calls

Module version 6.3.3

2019-04-18

- Enhanced the information displayed for mode switched events in the Rte Editor
- ASCRTE-5616 Fixed known issue: The Rte_Read API does not protect receiving of a Com signal against concurrent access if DirectReadFromCom applies
- Added support for autonomous error reaction on client side
- ASCRTE-5586 Fixed known issue: User-defined ExclusiveArea callouts are not aware of multiple SWC instances
- Improved spinlock allocation for cross-core queues in case of unmapped executables
- ASCRTE-5637 Fixed known issue: The Rte Editor configures wrong trace hook function names for COM callbacks
- Improved client data structure type naming
- Changed the return code of Rte_Read in case of intra-partition inter-ECU sender-receiver communication with data transformation; the RTE_E_SOFT_TRANSFORMER_ERROR and RTE_E_HARD_TRANSFORMER_ERROR of the previous transformer invocation is returned repeatedly unless new data are available and executeDespiteDataUnavailability is false
- ASCRTE-5614 Fixed known issue: The Rte calculates a wrong deserializing transformer buffer size if subElement mappings/partial records and a non-constant bufferComputation are used
- ASCRTE-5607 Fixed known issue: The generated Rte BSWMD file does not contain the OUTGOING-CALLBACK entries for defined RteVfbTraceFunctions if RteGeneratorOutput is set to BSW_SCHEDULER_ONLY

Module version 6.3.2

2019-03-22

- Improved the transformer hard error handling for intra-ECU transformations
- Replaced SMC API by direct buffer access
- Improved allocation of OsResources for ReceiveSignals if no API uses them

Module version 6.3.1

2019-02-15

- Implemented `Rte_IsAvailable` and `Rte_SetAvailable` APIs to query and modify the availability status of optional structure elements

- ASCRTE-5564 Fixed known issue: The Rte generator falsely reports an error if a CompuMethod of category TEXTTABLE contains non-point ranges
- Added support for RteBswRequiredSenderReceiverConnections in the Rte Editor
- ASCRTE-5566 Fixed known issue: The Rte Editor faces delays on the attempt to change the position of an EventToTaskMapping entry
- Improved generation of Rte_DummyDirtyFlag
- Added support for flag ExecuteDespiteDataUnavailability, for unqueued sender-receiver communication
- Improved the Rte Generator by avoiding null pointer exception when allocating a BswModeSwitchEvent with an invalid TargetModeRef
- Implemented verifier to check transformer buffer sizes against Com signal(group) sizes
- ASCRTE-5578 Fixed known issue: The Rte generates an invalid switch statement if a client-server provide port is connected but no ServerCallPoint references the operation
- Improved: The Rte Generator allows non-mapped client executables (i.e. executable with unknown task context) in case of intra-ECU result-free asynchronous client-server communication
- Updated the Rte Verifier to report an error if a VariableDataPrototype instance in the role ramBlock is accessed by SW-C instances of different partitions
- Added verifier to check that the returnSignal and callSignal of a ClientServerToSignalMapping are mapped to Com signals of type UINT8_N or UINT8_DYN
- Improved: The Rte Generator now considers the SwAddrMethod in the memory section of a BswSchedulableEntity/BswCalledEntity, in case it is present
- Added configurable generation of timestamp for generated RTE files
- ASCRTE-5565 Fixed known issue: The Rte generator falsely reports an error if the Init-Value of an ApplicationPrimitiveDataType does not fall in the internal interval of the CompuMethod
- Implemented additional verifier for Implementation data type according to [constr_1383]
- Updated the Rte Verifier to report an error if the mode values of a mode declaration group do not fit into the mapped implementation data type
- Improved verification of compatibility of Com signal type with base type of NetworkRepresentationProps

Module version 6.3.0

2019-01-25

- Improved: The Rte Generator now verifies that no Bsw event triggering any type of Bsw module entity is mapped to the Bsw Os task

- Added support for a new configuration parameter "Respect configured task type" to prevent the Rte from ignoring the configured task type BASIC although an EXTENDED task is required and to report an error message instead
- Added verifier for missing configured BSW module implementation reference
- Added support for data filtering on sender side for inter-ecu sender/receiver communication
- Implemented additional verifier for SwcBswMapping according to [constr_4085] and [constr_4084]
- Updated the Rte verifier to report a warning if a ComSignal has the wrong direction

Module version 6.2.27

2018-12-21

- Added a configuration parameter (HumanReadableBufferNames) to switch between the generation of hashed buffer names and human readable buffer names
- Improved: The Rte Generator shall report warning RTE_526 only for the used data element prototypes of an interface that do not have an initial value defined in the port prototype's ComSpec
- ASCRTE-5539 Fixed known issue: The Rte generates illegal hexadecimal constants on 64-bit architectures under certain conditions

Module version 6.2.26

2018-12-13

- Improved the Rte Bswmd generator to export the McSupportData as self-contained artifact
- Improved ROM consumption of the Rte by removing the unnecessary zero-based initializations of transformer rawValue arrays
- Added support for Lock-free queues
- ASCRTE-5525 Fixed known issue: The Rte Generator fails with an error if a signal uses a data type of category VALUE referencing a SWBaseType
- Fixed known issue: The Rte Generator fails with an error when the NvBlockSwComponentType has an unconnected client port in the role NvMNotifyJobFinished
- ASCRTE-5519 Fixed known issue: The Rte_Write API does not update a data handle variable in an N:1 sender-receiver communication under certain circumstances
- ASCRTE-5527 Fixed known issue: If a server runnable handles multiple inter-partition operations with an array argument of different length, the argument's data may be lost

Module version 6.2.25

2018-11-23

- Improved the access of RTE Implementation Plugins (RIPS) by applying the reference type URI-REFERENCE for RteRipsPluginConfigurationRef
- Improved sorting of column Position on the Event Mapping tab of the Rte Editor
- Improved verifier for constr_1011 in case of SwBaseType with category VOID
- ASCRTE-5503 Fixed known issue: The Rte generates inconsistent memory sections for a per-instance memory that is mapped to an NvM block
- Added support for time-triggered server executables
- Changed queuing support of triggers to strictly adhere to [SWS_Rte_07087] refined with AUTOSAR 4.4.0
- Improved: The Rte now supports including additional type header files for ImplementationDatTypes whose typeEmitter is different from RTE
- Added support for mixed data conversion for intra-ECU SenderReceiverInterfaces (SCALE_LINEAR_AND_TEXTTABLE to TEXTTABLE or LINEAR) and mixed linear conversion for network representation according to rte_sws_3832

Module version 6.2.24

2018-11-02

- Improved queue efficiency and memory consumption (e.g. for queued sender-receiver communication). RTE_E_LOST_DATA is now only reported on first reception after the overflow occurred in case of inter-partition queued sender-receiver communication
- Added the return code SCHM_E_LOST_DATA for the SchM_Receive API
- ASCRTE-5367 Fixed known issue: The Rte does not reset the transformer error status at the beginning of an implicit write
- ASCRTE-5462 Fixed known issue: The Rte creates too small buffers for Com reception and transformers if partial record support is used
- Added support for multiple data elements in the context of a software component port being all mapped to one data element in the context of a NvBlockSwComponentType port using TextTableMappings
- Added warning message in case locking strategy is overruled by the Rte (for OsResources only in case of unmapped runnables)
- ASCRTE-5465 Fixed known issue: Possible corruption of mode disabling dependencies in case of multiple inter-partition mode communications
- ASCRTE-5474 Fixed known issue: The server may not be called synchronously if an inter-partition intra-core client is mapped to a non-preemptive task

- ASCRTE-5461 Fixed known issue: The Rte aborts the execution of all subsequent transformer chains if a transformer returns a hard error
- Added support for the RTE Implementation Plugins (RIPS) for explicit/implicit sender receiver communication
- Implemented support for inter-partition queued triggering for application software components and basic software modules.
- Implemented: Mapping of ExternalTriggerOccurredEvents for category 1 executables to category 2 ISRs
- ASCRTE-5452 Fixed known issue: The SchM_Trigger API may activate the triggered executables multiple times in case of 1:n external trigger communication
- ASCRTE-5433 Fixed known issue: The Rte Generator fails with an error in case of mixed Rte_IWrite/Rte_-Read API and non-mapped reader
- Added support for sender-receiver intra-partition data conversion for CompuMethod categories TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE
- ASCRTE-5445 Fixed known issue: Rte_Read/Rte_IRead/Rte_IStatus might return the wrong value until first reception of data if invalidation for complex types is used
- Added support for InitEvents triggering runnable entities via RteInitializationRunnableBatch
- Added a prefix for Shared Memory Communicator (SMC) inter-core variables

Module version 6.2.23

2018-10-03

- Enabled moving of multiple mapped events / runnables at once and editing of position numbers
- Changed Rte Bswmd generator to a module dependent pre generator
- ASCRTE-5271 Fixed known issue: The Rte may not compile if intra-partition mode management is used and inter-partition events with mode disabling dependencies exist
- Updated verification of [constr_4022] to accept BswModuleDescription.implementedEntry according to AUTOSAR 4.3.1
- ASCRTE-5420 Fixed known issue: The prioritization rules of attribute SwDataDefProps.swImplPolicy are not handled properly for a VariableAccess to a sender-receiver data element
- ASCRTE-5423 Fixed known issue: Os events may be falsely shared for executable with a minimumStartInterval greater zero
- Implemented: Support for multiple dirty flags
- ASCRTE-5421 Fixed known issue: The Rte generation fails when using floating point values for CompuScale limits
- Added support for empty initialization of variable size arrays

- ASCRTE-5446 Fixed known issue: The ROM Block Location Symbols are not generated in the Rte_-NvMData.h if SwcServiceDependency.assignedDatas in the role "defaultValue" are used
- ASCRTE-5432 Fixed known issue: Rte_Cbk.h might not include Nvm_Types.h when partitioning is enabled with NvBlockSwComponents

Module version 6.2.22

2018-08-24

- ASCRTE-5327 Added deviation: A DataReceivedEvent for a queued sender-receiver data element is triggered even if enqueuing of the data failed due to a full receive queue
- Implemented: Component Data Structure (CDS) generation optimization regarding data handle buffers and inter runnable variables
- ASCRTE-5311 Fixed known issue: The generated Rte BSWMD file does not contain all used memory sections

Module version 6.2.21

2018-08-08

- Implemented: A configuration parameter (SpinlockAllocationStrategy) that makes it possible to allocate spinlocks with different strategies
- ASCRTE-5387 Fixed known issue: The Rte generator fails with an error if an event with activationReasonRepresentation is non-exclusively mapped to a task
- Improved: The Rte shall find the NvMBlockDescriptor entry of an NvBlockDescriptor via name matching if no RteNvmBlockRef exists for it
- ASCRTE-5378 Fixed known issue: The Rte does not set Xfrm(Inv)TransformerBswModuleEntryRef correctly if name clashes during the allocation of XfrmImplementationMappings occur

Module version 6.2.20

2018-07-27

- Implemented basic task support for Rte and Bsw timing events mapped to the same task in case OneScheduleTablePerPartition is enabled
- ASCRTE-5328 Fixed known issue: The Rte might not send the result of a server runnable that is started by multiple OperationInvokedEvents which are mapped to the same task
- Changed queueing behavior for asynchronous result-free client-server communication by permitting multiple outstanding server invocations for the same client

- Added new optional configuration parameters RteServerQueueLength and RteBswServerQueueLength as defined by AUTOSAR RfC #79150
- Added configuration check for BSW mode communication with mode users on multiple partitions
- Added support for configurable type of buffer length for transformers

Module version 6.2.19

2018-06-28

- Updated the included memory mapping header file for Bsw module entities in the sense of [SWS_Rte_-07830], i.e. <Msn>[_<vi>_<ai>]_MemMap.h
- Added support for InitEvents triggering runnable entities via OS task
- Removed error message RTE_59 which states that the implementation selection could not be found although there is only one SwcImplementation available and thus does not require a configuration of the RteImplementationRef parameter
- Implemented vendor-specific MetaData support of type SOCKET_CONNECTION_ID_16 for inter-ECU client-server communication
- ASCRTE-5259 Fixed known issue: The Rte overwrites the complete RamBlock despite a configured TextTableMapping if two PR-Ports are connected
- ASCRTE-5151 Fixed known issue: The Rte doesn't allocate consistency mechanism for atomic type in receiver buffer despite enabled serializer
- ASCRTE-5269 Fixed known issue: The Rte_NvMData.h may declare parameter buffers as extern and static if the calibration parameter uses a ServiceDependency
- Removed the declaration of a runnable entity from the Application Header File, if a SwcBswRunnableMapping exists for it (AUTOSAR RfC 79717)
- Added support for swAddrMethod respectively swDataDefProps in 'C' typed PerInstanceMemory
- ASCRTE-5214 Fixed known issue: If a server runnable handles multiple operations with an array argument of different length, the argument's data may be lost
- ASCRTE-5274 Fixed known issue: The Rte Generator may generate duplicate event names if Bsw communication with multiple internal behaviors is used
- Implemented check for LdCom usage without DataTransformation
- Removed reporting of message RTE_612 in case of UINT8_N or UINT8_DYN signal type
- Implemented support for NvM API runnables that can now be mapped to a task
- Changed 'Required' attribute in EventMapping-Tab of Rte Editor for BswOperationInvokedEvents and BswInternalTriggerOccuredEvents to false

- ASCRTE-5294 Fixed known issue: The Rte generates an insufficient counter data type for queued Rte or Bsw trigger communication if the configured queue length is 256
- ASCRTE-5297 Fixed known issue: The Rte generates an insufficient counter data type for data filter oneEveryN if the configured period is 256
- Fixed known issue: Opening and Closing of 'NVRAM Allocation' of the Rte Editor erases the NVRAM Allocations made in the Generic Editor if no Service Dependency exists for the mapping
- ASCRTE-5272 Fixed known issue: The NvBlockDescriptor's RunnableEntity for storeImmediate is triggered too often
- Implemented: The Rte shall find the NvMBlockDescriptor entry of an NvBlockDescriptor via name matching if no RteNvmBlockRef exists for it

Module version 6.2.18

2018-05-25

- Improved memory usage for mixed implicit and explicit sender-receiver communication; the explicit Rte_Read API now reads from an existing data handle buffer under certain circumstances
- ASCRTE-5161 Fixed known issue: The optional transformerError argument is not considered for the intra-ECU asynchronous Rte_Call() API
- ASCRTE-5136 Fixed known issue: Under certain conditions the Rte generates an invalid signature for Rte_ComHook_<ComSignal>_SigTx/SigRx trace functions
- Implemented support of maps and curves (compound primitives)
- ASCRTE-5200 Fixed known issue: Floating point constants cause compiler warning
- Implemented support of transport protocol for serialization
- ASCRTE-5167 Fixed known issue: The Rte calls a runnable in a task of another partition or falsely reports error RTE_571 when software components mapped to different partitions have compatible timing events mapped to the same task
- ASCRTE-5252 Fixed known issue: Dirty Flag storeImmediate NvBlockDescriptor's RunnableEntity is triggered too early

Module version 6.2.17

2018-04-20

- Added support for providing the activating event of an ExecutableEntity
- ASCRTE-5202 Fixed known issue: The generated Rte may not compile if an inter-partition BswOperationInvokedEvent is mapped to a task with many other events

- ASCRTE-5218 Fixed known issue: The Rte Generator may generate an insufficient buffer in a Com callback for a multiple receiver transformer scenario
- ASCRTE-5165 Fixed known issue: The generated Rte BSWMD file contains non-existing measurement symbols if IOC partitioning support is enabled
- Implemented checks for LdCom usage without DataTransformation and for configuring the same ISignalToIPduMapping for multiple Com/LdCom containers.
- Improved the Rte generator with ClientIdDefinitionSet support for inter-ECU client-server communication
- ASCRTE-5049 Fixed known issue: The Rte generates non-compilable code if all BSW module instances are mapped to the same partition and the RteGeneratorOutput is BSW_SCHEDULER_ONLY

Module version 6.2.16

2018-03-16

- Added support for setting the value of Rte_TransformerClass to RTE_TRANSFORMER_UNSPECIFIED and Rte_TransformerErrorCode to E_OK for transformer error argument of an Rte Api to which no data transformation applies
- ASCRTE-5129 Fixed known issue: The transformer error parameter for the Rte_Write()/Rte_Send()/Rte_Call()/Rte_Trigger() function is not updated when the partition is not active
- Implemented support for multiple Com instances
- ASCRTE-4991 Fixed known issue: The generated Rte code may cause an OS_E_INTDISABLE error if the Det checks are enabled
- Improved the check for the error message RTE_803
- Removed Permanent RAM Block consideration for SwcServiceDependency / RoleBasedDataAssignment roles different to "ramBlock" or "ramMirror"

Module version 6.2.15

2018-02-23

- ASCRTE-5112 Fixed known issue: The Rte Generator may fail if multiple IRead APIs are configured for the same data element and no initial value is specified
- Removed unused sender-receiver buffers in case only implicit read accesses and no write accesses are configured
- ASCRTE-4993 Fixed known issue: The Rte verifier does not consider the default value of the swImplPolicy
- Improved Rte Generator by ignoring unmapped runnables for send/receive signal/signal group data consistency. The new warnings RTE_610 and RTE_611 are reported for affected signals/signal groups

- ASCRTE-5089 Fixed known issue: The Event Mapping tab of the Rte Editor reports a NullPointerException if a BswOperationInvokedEvent is configured
- Improved NvBlockSwComponentType dirty flag support: A runnable in the context of a NvBlockSwComponentType can now handle multiple NvRamBlocks
- Improved Rte Generator to support LINEAR CompuMethods with a factor equal to zero
- ASCRTE-5080 Fixed known issue: The values of the predefined error codes are generated with a cast to the type Std_ReturnType
- Improved the filtering in Rte Editor for required event mapping for DataReceivedEvent(s) of NvBlockSwComponentType(s)
- Improved generation of function like macro Rte_WaitGetClearEvent by calling the Rte trace hooks even if Os defines its own WaitGetClearEvent function
- Improved receive buffer sharing for explicit intra-partition 1:n sender-receiver communication
- Added support for debounced activation of executable entities
- ASCRTE-5135 Fixed known issue: BSW Os task shutdown may be delayed
- Improved data conversion by considering the baseType of the NetworkRepresentationProps

Module version 6.2.14

2018-01-19

- Improved generation of very large numerical value specifications and data filter attributes
- Improved the Rte generator to accept a missing BswImplementation.ResourceConsumption that is mandatory by means of the strict AUTOSAR schema
- ASCRTE-5066 Fixed known issue: The Rte generator may fail if the number of elements of an ApplicationRecordDataType of a require port is less than the number of signals of the signal group mapped to the provide port
- ASCRTE-4691 Fixed known issue: The timing event execution offset may be shifted after a partition restart

Module version 6.2.13

2017-12-20

- ASCRTE-5019 Fixed known issue: The generated Rte may not compile if the Det checks are enabled and multiple clients call the same server by a direct function call
- Added support for external replacement invalidation for sender-receiver communication
- Added support for handling of Section Name Prefix for BswSchedulableEntitys

- ASCRTE-4965 Fixed known issue: The Rte reports an error if a variable size array is initialized with an application value specification without a constant mapping

Module version 6.2.12

2017-12-11

- ASCRTE-4979 Fixed known issue: The Rte contract phase does not compile if implicit sender-receiver communication using array type is configured
- Improved Rte generation time for data types, especially for single-pointered or double-pointered calibration parameter reference tables
- ASCRTE-4995 Fixed known issue: The Rte generator fails if an operation argument is named "Status" and inter-partition-intra-ECU client-server communication is used
- ASCRTE-4975 Fixed known issue: Subsequent calls of Rte_Call() always return RTE_E_LIMIT if RTE_E_HARD_TRANSFORMER_ERROR occurred in the result signal callback (inter-partition)
- ASCRTE-4689 Fixed known issue: Com_SendSignal/Com_SendSignalGroup function may be accessed concurrently for the same Com signal/Com signal group
- ASCRTE-4973 Fixed known issue: Os schedule table offset is not always taken into account in case of OSEK compatibility mode
- Updated Memory Mapping: No additional type qualifier is added to the Memory Allocation Keyword if a SwAddrMethod is configured
- Added support for implicit communication for directly called ModeSwitchEvent runnables in the context of asynchronous mode communication

Module version 6.2.11

2017-11-17

- ASCRTE-4976 Fixed known issue: The Rte generator fails if PossibleErrorRefs are used for asynchronous inter-partition-inter-ECU client-server communication
- Removed falsely reported warnings for constr_9082
- ASCRTE-5010 Fixed known issue: The access to the event bit mask of the internal Rte event buffer mechanism may be out-of-bounds
- ASCRTE-4326 Fixed known issue: Some inter-partition channels are not protected against concurrent access
- Added support for intra partition Bsw client/server calling chains
- ASCRTE-4969 Fixed known issue: Enumerations of ImplementationDataTypeElements of category VALUE are not considered

Module version 6.2.10

2017-10-20

- Improved task mapping verification for trigger communication
- ASCRTE-4948 Fixed known issue: The Rte generates non-compilable code in case of intra-partition sender-receiver communication with complex data type and handleInvalid set to REPLACE
- ASCRTE-4964 Fixed known issue: Subsequent calls of Rte_Call() always return RTE_E_LIMIT if RTE_E_HARD_TRANSFORMER_ERROR occurred in the result signal callback

Module version 6.2.9

2017-10-06

- ASCRTE-4846 Fixed known issue: The Rte_Enter and Rte_Exit API return a non-void value if OsResource is used and function elision is enabled
- ASCRTE-4782 Fixed known issue: The RTE generates incorrect code for sender/receiver communication with transformers and partial records but no subElementMappings
- ASCRTE-4928 Fixed known issue: The Rte does not propagate the transformerError to Rte_Read() for unqueued inter-partition inter-ECU communication
- Added support for implicit communication for directly called triggers
- ASCRTE-4924 Fixed known issue: Subsequent calls of Rte_Call always return RTE_E_LIMIT if RTE_E_HARD_TRANSFORMER_ERROR or RTE_E_COM_STOPPED was returned before
- ASCRTE-4847 Fixed known issue: Implicit inter-runnable variable buffer is not updated for a RunnableEntity that is mapped to a BswModuleEntity and started by a BSW event
- Implemented support for measurement of Nv RAM Block and non volatile data communication
- Added dirty flag support for NvBlockSwComponentType
- Added support for the Rte_NvMNotifyJobFinished callback
- ASCRTE-4899 Fixed known issue: The Rte does not generate any code if an invalid value is specified for a deeply nested element
- Implemented memory mapping initialization strategy
- ASCRTE-4925 Fixed known issue: The Rte may generate a non-existing memory section in the SchM module interlink header
- Added support for Bsw client/server communication

Module version 6.2.8

2017-08-25

- ASCRTE-4830 Fixed known issue: An out of bounds access may occur in the APIs Rte_Enter / Rte_Exit if the Det check is enabled in the Rte
- ASCRTE-4850 Fixed known issue: The Rte Generator fails if a blocking Rte_Result without timeout references an asynchronous server call point with a timeout
- ASCRTE-4893 Fixed known issue: The Rte_Main.h may declare parameter buffers as extern and static if partitioning is enabled
- Added support for SenderReceiverToSignalGroupMapping/SenderRecArrayTypeMapping to byte arrays
- Added support to configure if the Os supports OsSpinlockLockMethod which allows the Rte to generate optimized spinlocks
- ASCRTE-4860 Fixed known issue: The Rte Editor configures wrong trace hook function names for BSW Schedulable entities
- ASCRTE-4892 Fixed known issue: The Rte Generator fails if partitioning is disabled and a basic software module instance or software component instance is mapped to an Os application
- Added deviation to document that the Rte allows multiple inter-partition synchronous client server calls for the same client

Module version 6.2.7

2017-07-28

- Improved sharing of inter-partition S/R channels
- ASCRTE-4834 Fixed known issue: The Rte generator may apply a wrong event setting mechanism in case of inter-partition inter-ECU client-server communication
- Removed unused Rte_SetEvent macros from Rte_Intern<Partition> header files
- ASCRTE-4857 Fixed known issue: ParameterElementGroup types are not generated in Rte_Main.h
- ASCRTE-4773 Fixed known issue: The Rte_Read API does not overwrite the whole OUT parameter while reading from a NvRamBlockElement with a bitfield texttable mapping

Module version 6.2.6

2017-06-30

- ASCRTE-4737 Fixed known issue: Rte could report a misleading error message for events with mode-disabling dependencies in certain conditions
- ASCRTE-4687 Fixed known issue: The Rte_Read API does not protect receiving of a Com signal group against concurrent access if DirectReadFromCom applies
- ASCRTE-4781 Fixed known issue: The Rte considers a sender/receiver partial record communication as incompatible if the matching structure elements are not in the same order

- Added user-defined exclusive area implementation mechanism
- ASCRTE-4768 Fixed known issue: The RTE ignores the PortAPIOption errorHandling for the generation of direct function calls to a server runnable
- Added support for intra-ECU sender-receiver communication with data transformation
- ASCRTE-4819 Fixed known issue: Implicit S/R communication is not working for a RunnableEntity that is mapped to a BswModuleEntity and started by a BSW event

Module version 6.2.5

2017-05-31

- ASCRTE-4747 Fixed known issue: The Rte Generator may fail with an OutOfMemoryError if many direct server calls are configured
- ASCRTE-4802 Fixed known issue: The Rte may falsely report the error RTE_538 during execution of the SvcAs
- Improved the order of intra-partition unqueued sender/receiver receive buffers definitions to reduce the amount of MemMap.h inclusions

Module version 6.2.4

2017-05-05

- Removed filter flag from sender/receiver buffer
- Added support for Os alarm and schedule table allocation with Os counter on different core. This allows to use a single Os counter on a partitioned multi-core system
- ASCRTE-4665 Fixed known issue: Rte_Read does not return the initial value of a complex inter-ECU communication data under certain conditions
- ASCRTE-4683 Fixed known issue: The Rte does not send any data if the S/R data queue overflows in case of inter-partition-inter-ECU communication
- ASCRTE-2026 Fixed known issue: If inter-partition mode management is used with mode disabling dependencies, the Rte may not compile
- ASCRTE-4695 Fixed known issue: The synchronous client may not wait for the server to finish while an exclusive area is active
- ASCRTE-4760 Fixed known issue: The Rte Generator fails if a structure ImplementationDataType contains a pointer element with an invalid target type
- Removed the receive buffer structure data types for intra-partition unqueued sender/receiver communication. Each receive buffer element is now a discrete global variable

- Updated the Rte Verifier to report an error if a BSW requiredTrigger is connected to more than one BSW providedTriggers

Module version 6.2.3

2017-03-31

- Added support for autonomous error reaction for inter-ECU client/server communication with transformers
- Added support for TransformerHardErrorEvents for inter-ECU client/server communication
- ASCRTE-4698 Fixed known issue: DataReceiveErrorEvents are not triggered if inter-ECU S/R invalidation is handled by the Rte and no data receive point exists
- ASCRTE-4651 Fixed known issue: The Rte may not compile if a schedulable entity and a runnable entity with a blocking API are mapped to the same task
- ASCRTE-4692 Fixed known issue: The Rte doesn't create the memory section in the Rte_Bswmd.arxml for a RamBlock of a NvBlockSwComponentType
- ASCRTE-4704 Fixed known issue: The Rte generates uncompilable MemMap definitions if SectionInitializationPolicies are used according to AUTOSAR specifications
- Improved data consistency by locking the interrupts when a spinlock is used
- ASCRTE-4621 Fixed known issue: The RTE does not provide NULL to all transformers in a chain if executeDespiteDataUnavailability is enabled
- Added suport for bitfield mapping of multiple require variable data prototypes to one provide variable data prototype
- Implemented Asynchronous Mode Switch
- Added allocation of init and invalid values for ComGroupSignals and ComRxDataTimeoutAction for ComGroupSignals/ComSignals via the service needs assistant
- Removed unused Rte_IsModeDisablingDepSet functions from Rte_<partition>.c files

Module version 6.2.2

2017-03-03

- ASCRTE-4064 Fixed known issue: Implicit communication with data conversion is not working for send signals/signal groups
- Improved measurement support for inter-partition sender/receiver communication in conjunction with the SharedMemory communication mechanism
- ASCRTE-4664 Fixed known issue: The Rte Editor does not provide available NvM blocks

- Improved queued triggering to use basic task where extended task is not mandatory
- Improved generated code relaxing the restriction regarding the SwcInternalBehavior in case of NvBlockSwComponentType
- ASCRTE-4531 Fixed known issue: DataFilter oneEveryN is not taking into account the offset and period properly

Module version 6.2.1

2017-02-03

- ASCRTE-4131 Fixed known issue: The Rte does not consider the initial value for an unconnected parameter port and reports an error
- ASCRTE-4547 Fixed known issue: The Rte does not generate a Rte_IrTrigger API for each RunnableEntity if the InternalTriggeringPoints have equal short names
- ASCRTE-4599 Fixed known issue: An exclusive area which uses COOPERATIVE_RUNNABLE_PLACEMENT may be disabled by another exclusive area which uses OS_RESOURCE
- Improved generated code for Rte_Feedback API in case "function elision" is enabled
- Added custom constraint checks for LINEAR CompuMethods in the sense of constr_1375
- Improved task mapping scenarios for runnables with a synchronous server call point that is implemented as direct function call. The runnable may now be mapped to a basic task regardless of the configured timeout value
- ASCRTE-4409 Fixed known issue: The component data structure may not be initialized if the SWC contains two provide ports which refer the same trigger interface
- Updated the inter-partition mode receive emptyQueue

Module version 6.1.172

2017-01-05

- Added support for 64 bit datatypes in sender/receiver, client/server and inter-ECU communication

Module version 6.1.171

2016-12-14

- ASCRTE-4554 Fixed known issue: DataReceiveErrorEvents are not triggered in case of inter-partition inter-ECU S/R invalidation

- Consider memory mapping for SMC inter-partition communication
- ASCRTE-4525 Fixed known issue: DataFilter newIsWithin is more restrictive than specified
- ASCRTE-4528 Fixed known issue: TaskType BASIC leads to an extended task for Task Mapping Scenario B1
- ASCRTE-4551 Fixed known issue: If a timeout is specified for an asynchronous server callpoint and no waitpoint is defined, the Rte Generator does not setup an alarm in Rte_Call
- ASCRTE-4558 Fixed known issue: The Rte generator fails for inter-ECU client/server communication if an ApplicationRecordDataType argument is mapped to an ImplementationDataType argument
- Improved merge of OsEvents for multiple TriggerOccuredEvents
- ASCRTE-4573 Fixed known issue: The Rte generator fails if a complex ApplicationArrayDataType with variable size semantics is used
- ASCRTE-4483 Fixed known issue: The Rte generates empty Rte_Invalidate API in case of complex inter-ECU sender/receiver communication
- ASCRTE-4524 Fixed known issue: Invalidation of primitive inter-partition inter-ECU sender/receiver communication is not working
- ASCRTE-4553 Fixed known issue: Inter-ECU S/R invalidation of signal groups does not consider the parameter InterECUInvalidationHandledByRte
- ASCRTE-4482 Fixed known issue: The Rte generates a non-blocking instead of a blocking Rte_Feedback API under certain conditions
- Implemented function elision for the Rte_Trigger and Rte_IrTrigger API
- Added support for memory mapping considering SwAddrMethod, SwAlignment and mapping of RunnableEntities in the Application Header File
- Improved declaration of shared memory API in Smc.h by generating partition-specific header files

Module version 6.1.170

2016-11-08

- ASCRTE-3346 Fixed known issue: The Rte does not generate Rte_Main.[ch] if partitioning is enabled but only one partition is configured
- ASCRTE-4434 Fixed known issue: If COOPERATIVE_RUNNABLE_PLACEMENT is configured for an exclusive area, the Rte might allocate more than one internal resource for a task
- ASCRTE-4176 Fixed known issue: Port interface mappings may not be applied for client server interfaces
- ASCRTE-4497 Fixed known issue: If limits are defined for a signed data type, the Rte generates an unsigned suffix
- Fixed violation for MISRA rule 8.8 by moving external declaration of implicit buffers to Rte_Partitioning.h

- ASCRTE-4210 Fixed known issue: The Rte Editor may create duplicate Bsw module instances
- Added support for asynchronous inter-ECU client server communication
- ASCRTE-4532 Fixed known issue: The Rte generates an invalid EventMaskType typedef under certain circumstances
- Extended data consistency for inter-core C/S communication related to bugfix ASCRTE-4445
- Fixed MISRA rule 5.5 violations which are related to the partition state and execution counter variables
- ASCRTE-4110 Fixed known issue: Some APIs do not consider data conversion correctly
- Implemented memory mapping by the size of intra-partition variables if no explicit SwAddrMethod is given
- Added inter-partition-inter-ECU-inter-core communication optimization: one Com signal IP channel for each core
- ASCRTE-4388 Fixed known issue: The Rte expects a parameter data prototype mapped to a per instance memory in the same RoleBasedDataAssignment
- Improved verifier for constr_1094 in case of runnables of NvBlockSwComponentType

Module version 6.1.169

2016-10-10

- Improved the resolution of SwDataDefProps for implementation data types of category TYPE_REFERENCE.
- ASCRTE-4487 Fixed known issue: The Rte uses a wrong buffer size for Com/LdCom signal reception callbacks when client/server inter-ECU communication is used
- Relaxed the verifier for implementation data types with the same short name and DataConstr between C/S arguments
- Added configuration support for trigger queue lengths in the Rte Editor
- Added support for SubElementMapping for inter-ECU S/R Data Serialization

Module version 6.1.168

2016-09-09

- Added support for sharing implicit S/R communication buffers (data handle buffers) between partitions
- ASCRTE-4590 Fixed known issue: The elided Rte_Write API macro does not update data handle buffers of implicit readers
- ASCRTE-4389 Fixed known issue: The Rte Editor may falsely warn that the schedule table expiry points exceed 5000 entries

- ASCRTE-3906 Fixed known issue: The Rte does not consider the activation mechanism for the SchM schedule table

Module version 6.1.167

2016-08-05

- ASCRTE-4424 Fixed known issue: The Rte Generator may fail if a runnable is started by multiple OperationInvokedEvents for different client/server operations
- Added queueing support for the SchM_ActMainFunction API
- ASCRTE-4433 Fixed known issue: The Rte Generator fails if incompatible inter-ECU client/server operations are called
- ASCRTE-4337 Fixed known issue: The runnable based scope for application header files might not detect a wrong runnable context
- Improved inter-partition intra-core synchronous Rte_Call to not wait for Os events anymore if the client's task priority is lower than the server's task
- ASCRTE-4426 Fixed known issue: The Rte Bswmd generation fails if measurement is enabled on a PRPortPrototype
- ASCRTE-4443 Fixed known issue: The Rte uses a wrong buffer size for Com signal reception callbacks when S/R inter-partition/inter-ECU communication is used
- ASCRTE-4444 Fixed known issue: The Rte Generator might not set the isUpdated flag in a Com/LdCom signal reception callback
- ASCRTE-4445 Fixed known issue: Inter-core inter-partition shared-memory communication wrongly shares Os resources
- ASCRTE-4384 Fixed known issue: The Rte resets Bsw modes during Rte_Start

Module version 6.1.166

2016-06-30

- ASCRTE-4360 Fixed known issue: The Rte Bswmd generation fails if an McDataInstance refers a CompuMethod without a Unit
- ASCRTE-4385 Fixed known issue: Implicit InterRunnableVariable communication might not work
- ASCRTE-4387 Fixed known issue: The enhanced Rte_Mode API returns a wrong mode if the mode provider partition is not started and the ModeDeclarationGroup has category EXPLICIT_ORDER
- Removed unused global buffers for unaccessed parameter data prototypes
- ASCRTE-4397 Fixed known issue: The Rte may not apply data consistency for inter-core SMC channels

- Added support for queued SchM_Trigger API
- ASCRTE-4362 Fixed known issue: The Rte might use the same Os event for an Bsw and Rte event
- ASCRTE-4393 Fixed known issue: Shared memory with inter-core-inter-ECU transformer leads to non-compilable code
- ASCRTE-4412 Fixed known issue: The Rte_Trigger API does not return a value for an unconnected queued trigger

Module version 6.1.165

2016-06-07

- Added use of SvcAs feature to configure VariableLength parameter of IocChannels
- ASCRTE-4329 Fixed known issue: Possible infinite wait in inter-core synchronous Rte_Call
- Added use of SvcAs feature to configure XfrmOsApplicationRef parameter of XfrmImplementationMappings
- ASCRTE-4288 Fixed known issue: Rte_Read might return RTE_E_OK although the API is unconnected
- Added support of SubElementMapping with Transformer
- Improved asynchronous inter-partition client/server communication so that the client runnable may now be mapped to a basic task
- ASCRTE-4358 Fixed known issue: A client-server request's sequence counter value may be truncated
- Added buffer length check for copying raw data to local buffer for re-transformation in (Ld)Com callback

Module version 6.1.164

2016-04-29

- ASCRTE-4279 Fixed known issue: Software component without Internal Behavior results in the failure of RTE
- Improved allocation of XfrmImplementationMapping so that only a single entry for the same tx signal (group) path is created and XfrmVariableDataPrototypeInstanceRef is left disabled
- ASCRTE-4200 Fixed known issue: The verifier wrongly reports an error about incompatible categories between two ImplementationDataTypeElements
- Changed dependency to the Platforms plugin from mandatory to optional
- ASCRTE-4295 Fixed known issue: The Rte generator fails if calibration support of a ParameterSwComponentType is different than for the connected software component
- Added support for the AutosarVariableIRef instance reference in NvRamBlockElements
- Implemented relaxation of constr_1175

- Added support for inter-partition/inter-ECU client server communication
- ASCRTE-3895 Fixed known issue: The enhanced Rte_Mode API does not return the initial mode if no ModeSwitchEvents exist for inter-partition mode management
- Removed unused mode variable from the mode provider partition in case of inter-partition mode management with enhanced Rte_Mode API
- ASCRTE-4317 Fixed known issue: The verifier wrongly reports an error if an ApplicationArrayDataType is mapped to a variable-size array ImplementationDataType
- Added support for variable-size arrays (VSA_LINEAR) for inter-partition queued sender-receiver communication with SMC
- ASCRTE-4261 Fixed known issue: An Rte client server reception callback function may retransform invalid raw data

Module version 6.1.162

2016-04-01

- ASCRTE-4243 Fixed known issue: The Rte generator fails if incompatible initial values are configured for certain data prototypes with measurement support
- ASCRTE-4223 Fixed known issue: Rte.c does not include NvM.h leading to compilation error when a software component uses NvMService through a NvBlockSwComponent
- Improved allocation of expiry points for timing events with an offset greater than 0
- ASCRTE-4252 Fixed known issue: The Rte does not configure the LdComRxIndication for client/server call signals
- ASCRTE-4248 Fixed known issue: An internal error is reported if a task chain is created of non periodic tasks
- ASCRTE-4233 Fixed known issue: The Rte creates nested MemMap defines
- ASCRTE-4269 Fixed known issue: ComIPChannel identifier defines may not be generated if senders are located on more than one non-Bsw partition
- Implemented the definition of partition specific and shared memory sections within the Rte's BSWMD
- ASCRTE-4260 Fixed known issue: The Rte does not consider invalid values on array application data types
- Shortened names of allocated XfrmImplementationMappings
- ASCRTE-4266 Fixed known issue: The Rte Generator does not generate code if inter-core C/S communication with SharedMemory is used
- ASCRTE-4281 Fixed known issue: If a task chain is configured, the Rte allocates expiry points for all tasks in the chain
- Added support for inter-partition/inter-ECU S/R communication

Module version 6.1.161

2016-03-04

- Reworked generation of template/example code for each software component.
- Added comment next to global variables which shows the original name before MD5 hashing is applied
- Improved the verification of self referencing datatypes to check indirect referencing
- ASCRTE-4228 Fixed known issue: The Rte editor generates wrong Vfb trace hook function names for certain API functions

Module version 6.1.160

2016-02-05

- Added use of SvcAs feature to configure LdComIPdus
- Added support for fire-and-forget (result-free) asynchronous client/server calls as specified by RfC 70295.
- Improved handling of the Rte basic software module description: The Basic Software Scheduler does not consider an existing Rte Bswmd as subject for code generation anymore

Module version 6.1.159

2016-01-15

- ASCRTE-4069 Fixed known issue: The Rte editor may enable unused configuration parameters
- Removed Det RTE_E_DET_UNINIT logging for the SchM_Enter and SchM_Exit Apis
- ASCRTE-4075 Fixed known issue: Incorrect warning regarding unconnected client server ports
- ASCRTE-4000 Fixed known issue: If transformerError argument exists the Rte template code does not compile
- ASCRTE-4072 Fixed known issue: Rte falsely reports an error when SwcServiceDependency is defined on a PortPrototype which is involved in Nv data management
- ASCRTE-4125 Fixed known issue: Under certain conditions the Rte does not generate enumeration definitions
- Adapted the validation of ModeRequestTypeMaps for support of multiple BswInternalBehaviors
- Updated the verification of constraint constr_4071
- ASCRTE-4145 Fixed known issue: The measurement symbols exported to the Rte Bswmd file may not exist in the generated source code
- Added support for mixed usage of IOC for inter-core and Shared Memory Communicator (SMC) for intra-core communication

- Added system description verifier check for constr_1386 (PortDefinedArgumentValue shall only be defined for AbstractProvidedPortPrototype)
- ASCRTE-4152 Fixed known issue: If inter-partition invalidation is used, it might happen that the receiver reads old data
- ASCRTE-4142 Fixed known issue: SchM_Send and SchM_Receive APIs to do not return a value when a port is unconnected
- ASCRTE-4114 Fixed known issue: DataSendCompletedEvents are not properly triggered under certain conditions
- ASCRTE-4159 Fixed known issue: An extended Os task may not be activated on startup
- ASCRTE-4178 Fixed known issue: The Rte Editor reports a NullPointerException under certain conditions

Module version 6.1.158

2015-11-10

- Added use of SvcAs feature to configure Nv blocks and callbacks
- Changed parameter name XfrmTransformationBswModuleEntryRef for reading from the Xfrm Ecu configurations to XfrmTransformerBswModuleEntryRef (see AUTOSAR RfC #68531)
- Added support for inter-partition task chains and multiple task chains
- Improved configuration check regarding the mapping of Bsw events to tasks belonging to the configured Bsw Os application
- Improved error messages for value specifications that are not compatible to the data type
- ASCRTE-4068 Fixed known issue: The Rte editor does not show any error or warning under several conditions
- Extended verifier check for constr_1075 (point 1.h is now checked according to RfC#67491/ASR 4.2.2)
- ASCRTE-4073 Fixed known issue: The Rte generator ignores initial values for an arTypedPerInstanceMemory defined by application value specifications
- ASCRTE-4071 Fixed known issue: A value specification for a multi-dimensional ApplicationArrayDataType can not be applied to a data prototype
- Added use of SvcAs feature to configure XfrmImplementationMappings
- ASCRTE-4080 Fixed known issue: Under certain conditions the Rte generator fails if measurement is enabled for a data prototype of complex application data type
- Removed erroneous warning if init value of primitive data element and mapped Com signal match
- ASCRTE-4041 Fixed known issue: The Rte writes array initial values to the Com configuration with an "U" suffix
- Updated the Rte Verifier to report a warning if a timeout is configured and LdCom is used

- Removed support for system signal/signal group mappings in the Rte editor. The functionality has been moved to the System Signal Mapping Editor.
- Removed Service Port Mapping tab from Rte editor. The functionality has been moved to the Connection Editor.
- Improved implicit read/write access for server runnables which are triggered by a direct call. The client(s) can be mapped on different tasks now as long as they do not preempt each other.
- Updated the Rte Verifier to report an error if a software component name conflicts with a standard Rte header file

Module version 6.1.157

2015-10-09

- ASCRTE-3984 Fixed known issue: Inter-partition mode switch events may lead to a deadlock on startup
- Added serialization support for QUEUED data reception
- Added supported for queued Bsw sender receiver communication
- Added support for specification of XfrmVariableDataPrototypeInstanceRef (multiple receivers)
- ASCRTE-4016 Fixed known issue: If no initial value is specified for a C typed per instance memory, the Rte generates non-compilable code
- Added support for the use of NumericalValueSpecifications to define the invalidValue of an array data type
- Updated the definition of the Rte_Cs_TransactionHandleType according to RfC#69581
- ASCRTE-4012 Fixed known issue: Rte miscalculates buffer consumptions for transformers in case the CompuNumerator contains a linear factor
- ASCRTE-4014 Fixed known issue: Blocking Rte_Receive may return undefined data even if the queue is empty
- Removed the SomeIp returnValue parameter when a server runnable has no application errors
- ASCRTE-4023 Fixed known issue: The transformerError argument passed to a runnable may contain undefined values
- Updated the system description verifier to reject configurations where executeDespiteDataUnavailability=true and signal fan-in is configured
- Exchanged partition IDs which are used for the naming of Rte_Smc_Data_<id>.c files with partition names
- Fixed the Rte verifier, which might wrongly report the warning that a category 2 runnable is not exclusively mapped to a task
- ASCRTE-4040 Fixed known issue: It is required to specify a DataTypeMap for single elements of an ApplicationCompositeDataType
- ASCRTE-4047 Fixed known issue: The Rte editor may remove data mappings from the system model

- Added support for the inclusion of multiple instances of a Bsw module into the same partition

Module version 6.1.156

2015-08-24

- Added support for the generation of the Bsw Scheduler in multiple partitions
- ASCRTE-3974 Fixed known issue: Undefined behavior for mode disabling dependencies which are defined in multiple partitions
- Added support for C/S calls to NvMServices
- Added verifier checks for Variable-Size Array ImplementationDataTypes
- ASCRTE-3983 Fixed known issue: An Rte_Call function may not dereference an INOUT operation argument
- ASCRTE-3923 Fixed known issue: Rte template code does not compile
- ASCRTE-3980 Fixed known issue: If the Rte contract phase is executed from the command line, the data type verifier is not executed
- ASCRTE-3905 Fixed known issue: The Rte does not consider the receiver's initial value if the invalidation policy is set to REPLACE
- ASCRTE-3991 Fixed known issue: High memory consumption when many complex application data types are defined
- ASCRTE-3955 Fixed known issue: Missing critical section for inter-partition-inter-ECU communication

Module version 6.1.155

2015-07-21

- ASCRTE-3933 Fixed known issue: The Rte Editor does not properly handle data mappings of application array data types within other complex types
- Updated verification check for constr_1060 according to RfC#68247
- ASCRTE-3989 Fixed known issue: Rte Api function arguments may reference an undefined data type
- ASCRTE-3963 Fixed known issue: The Rte verifier incorrectly evaluates the invalidation policy of sender receiver interfaces
- ASCRTE-3961 Fixed known issue: If an array of a structure implementation data type is specified the generated code does not compile

Module version 6.1.154

2015-06-19

- ASCRTE-3725 Fixed known issue: If the invalid value equals the init value, Rte_IStatus()/Rte_Read() returns RTE_E_OK for inter-partition invalidation
- ASCRTE-3798 Fixed known issue: A McDataInstance entry is not generated for ParameterDataPrototype elements of a ParameterSwComponentType
- ASCRTE-3823 Fixed known issue: Rte_Mode does not return the initial mode if no mode switch point exists for the provided mode declaration group
- ASCRTE-3705 Fixed known issue: If an executable entity runs inside several exclusive areas with different interrupt locking mechanisms, the Rte considers only one
- ASCRTE-3836 Fixed known issue: The Rte Editor does not store event mappings if software component and basic software instance names are ambiguous
- ASCRTE-3862 Fixed known issue: The Rte incorrectly requires the mapping of operation invoked events
- Removed the additional measurement buffer in case of 1:1 intra-partition sender/receiver communication
- Replaced buffer data structure by flat hierarchy
- ASCRTE-3852 Fixed known issue: The initial value of a data element may be wrong in case of unqueued 1:n inter-partition sender/receiver communication
- Improved the Rte Bswmd generator to only modify the system model (file SystemModel2.tdb) if necessary
- ASCRTE-3535 Fixed known issue: The Rte_Read API may return the last received value from Com instead of the initial value if HandleTimeoutType is set to REPLACE
- ASCRTE-3867 Fixed known issue: Rte_Write does not write to a subelement of a complex Nv ram block if the RootVariableDataPrototype is not set
- Removed constraint check constr_3131 (see RfC#69207)
- ASCRTE-3870 Fixed known issue: The generated Rte code does not compile if background events are used with partitioning enabled
- ASCRTE-3776 Fixed known issue: The Rte_Read API returns RTE_E_INVALID if the invalidation policy is set to replace and an alive timeout is specified or handle never received is enabled
- ASCRTE-3769 Fixed known issue: The Rte may still activate runnable entities on extended tasks after Rte_Stop has been called
- ASCRTE-3830 Fixed known issue: The Rte generates duplicate internal IDs for development error tracing if two software components define the same API function
- ASCRTE-3814 Fixed known issue: Inter-partition S/R communication lead to compiler warnings if several receivers use different structure types
- ASCRTE-3689 Fixed known issue: Mixing of enhanced and non-enhanced Rte_Mode API in inter-partition mode management fails

- ASCRTE-3290 Fixed known issue: The Rte may incorrectly report the error message RTE_503
- ASCRTE-3875 Fixed known issue: The Rte does not stop the BSW inter-partition task
- ASCRTE-3913 Fixed known issue: The Rte ignores incoming events during a blocking API call
- ASCRTE-3931 Fixed known issue: If an array implementation data type is referenced by an implementation data type of category TYPE_REFERENCE an internal error is reported

Module version 6.1.153

2015-04-27

- ASCRTE-3287 Fixed known issue: Sender receiver record element mappings to array implementation data type elements will not be recognized by the Rte editor
- ASCRTE-3799 Fixed known issue: The BSW main function auto-mapping functionality of the Rte editor does not re-create removed Os tasks
- ASCRTE-3800 Fixed known issue: Inter-partition inter-ECU sender-receiver signal fan-out does not work
- ASCRTE-3808 Fixed known issue: The Rte generator fails if a base type without a native declaration is used
- Added support for constant (e.g. initial) value specifications for complex application data types using RecordValueSpecification and ArrayValueSpecification
- Added support for inter-ECU client server communication
- ASCRTE-3819 Fixed known issue: Compiler error related to the definition of the data type Rte_TransformerError
- ASCRTE-3812 Fixed known issue: The inter partition Rte_Result API returns the wrong value after starting the partition
- ASCRTE-3821 Fixed known issue: The Rte may pass invalid data to the LdCom
- Added support for network representation conversion of complex sender-receiver data elements.
- Modified the allocation of the Rte_Result Api, an AsynchronousServerCallResultPoint is now required to generate the Rte_Result Api
- ASCRTE-3827 Fixed known issue: The Bsw Os Task may be unnecessarily activated during Rte_Start
- ASCRTE-3820 Fixed known issue: The Rte shall support definitions of limits in certain numerical formats
- Added support for SubElementMappings of VariableAndParameterInterfaceMappings

Module version 6.1.152

2015-02-23

- ASCRTE-3662 Fixed known issue: Update flag is not cleared when Rte_RestartPartition is called
- Added support for Internal Trigger Event Communication
- Implemented call of Com[Send/Receive]DynSignal only for ComSignalType UINT8_DYN
- Improved generated code for Sender / Receiver Serialization (use-case: usage of LdCom only)
- Corrected generated code for Sender / Receiver Serialization (use-case: LdComCbkTriggerTransmit)
- ASCRTE-3702 Fixed known issue: The Rte generator aborts with a fatal error if a Bsw trigger event is configured
- ASCRTE-3701 Fixed known issue: The Rte_Write API behaves like the Rte_Invalidate API if invalidation is configured and the invalid value is sent
- Added OS_SPINLOCK as possible parameter for selection of exclusive area implementation mechanism.
- Improved generated code for Sender / Receiver Serialization (use-case: transformerErroHandling enabled)
- Improved generated code for Sender / Receiver Serialization (use-case: transformerError prioritization)
- Improved generated code for Sender / Receiver Serialization (use-case: Transformer Soft/Hard error in Rte_Feedback/IStatus)
- Improved generated code for Sender / Receiver Serialization (use-case: macro definition for RTE_E_TRANSFORMER_LIMIT)
- Improved generated code for Sender / Receiver Serialization (use-case: consistent (de-)serialization error in case of multiple receiver software components)
- Improved generated code for Sender / Receiver Serialization (use-case: compiler warning in case of (de)-serialization and array data types)
- Improved generated code for Sender / Receiver Serialization (use-case: startup behavior (Rx - RTE_E_NEVER_RECEIVED+InitValues, Tx - RteFeedback))
- Improved generated code for Sender / Receiver Serialization (use-case: transformerError arg enabled + intra Ecu receiver)
- Added support for TextValueSpecifications to specify enumerations of the corresponding CompuMethod (category TEXTTABLE or BITFIELD_TEXTTABLE)
- Added support for Bitfield Textable PortInterfaceMapping for NvDataInterfaces.
- Added support for NvBlockSwComponentType NvBlockDescriptor romBlock and callback function Rte_NvMNotifyInitBlock.
- Added support for network representation conversion of primitive sender-receiver data elements.

Module version 6.1.151

2015-01-08

- ASCRTE-3576 Fixed known issue: The Rte contract phase does not compile if implicit interrunnable variables are configured
- ASCRTE-3580 Fixed known issue: The Rte may try to lock an Os resource from within a partition that is not permitted to access the resource
- ASCRTE-3573 Fixed known issue: Rte_ModeDisablingDep_Type may use an inadequate type when using inter-partition mode management
- ASCRTE-3581 Fixed known issue: The Rte reports an internal error if update is enabled for n:1 S/R mixed inter- and intra-partition communication
- Improved generation of interrupt locks for shared buffers which are not required for tasks which have the highest priority
- ASCRTE-3457 Fixed known issue: The order of invalid values in Application Types header file is arbitrary
- Added support for server runnables which are triggered by a direct call that can now have implicit read/write access
- ASCRTE-3592 Fixed known issue: If a task chain is configured and the first task is of task mapping scenario A2, the Rte code does not compile
- Added algorithm to generate Buffer names based on element names to ensure predictable symbols
- ASCRTE-3375 Fixed known issue: If SchM_Trigger or SchM_ActMainFunction is called the Rte ignores implicit exclusive areas
- ASCRTE-3625 Fixed known issue: The software component might receive old data if S/R data transformation is used
- Implemented [SWS_Rte_08408] so that a runnable with both implicit read and write access to a variable data prototype in the context of a PR-port uses a single shared read/write buffer
- ASCRTE-3638 Fixed known issue: The Rte does not generate code for an implicit exclusive area if EB_FAST_LOCK is configured
- ASCRTE-3621 Fixed known issue: Rte_SwitchAck returns RTE_E_TRANSMIT_ACK although the transition status isn't done
- ASCRTE-3657 Fixed known issue: The Rte does not consider precedence of data properties for the generation of the McSupportData
- ASCRTE-3653 Fixed known issue: The Rte allocates memory for constant memory if it is referenced by a parameter access
- ASCRTE-3661 Fixed known issue: Inter-partition S/R communication lead to compiler warnings if used with two different but compatible structures
- ASCRTE-3664 Fixed known issue: A RunnableEntity might be triggered by a BswModuleEntity even if the Rte is not started
- Added support for External Trigger Event Communication
- Added basic support for NvBlockSwComponentType.

Module version 6.1.150

2014-10-13

- ASCRTE-3474 Fixed known issue: The Rte contract phase generator fails with an error if implicit sender receiver communication is configured
- ASCRTE-3466 Fixed known issue: If the service needs assistant is executed the Rte treats disabled configuration parameters as enabled
- ASCRTE-3475 Fixed known issue: The Rte does not generate the BSW Module Description entities for measurement correctly in case of application array and application primitive data types
- Added invalidation support for inter-partition communication
- Added configuration parameter `DisableInvalidationDataConsistency` to disable data consistency for invalidation if invalidation policy is set to keep
- ASCRTE-3505 Fixed known issue: Complex inter-partition inter-ECU data may be corrupted if the received Com signal group is mapped to different data elements
- ASCRTE-3501 Fixed known issue: The Rte contract phase does not compile if implicit sender/receiver communication is configured
- ASCRTE-3508 Fixed known issue: The Rte generator fails in attempt to optimize the buffer handling for an implicit receiver connected to an explicit sender
- ASCRTE-3506 Fixed known issue: The Rte may exchange corrupted data in case of inter-partition inter-ECU sender/receiver communication using complex data types
- ASCRTE-3533 Fixed known issue: The generation of the Rte Bswmd file may take very long
- ASCRTE-3538 Fixed known issue: The Rte contract phase generator fails with an error if per-instance memory is configured
- ASCRTE-3539 Fixed known issue: The Bsw main function auto-mapping functionality maps ComM main functions to an inadequate task if multiple Flexray channels are configured
- ASCRTE-3543 Fixed known issue: The VFB Tracing tab of the Rte Editor does not list trace hook functions for the Rte_SwitchAck API
- ASCRTE-3458 Fixed known issue: If two or more SwBaseTypes with the same name exist, the Rte only considers one
- ASCRTE-3512 Fixed known issue: The Rte Generator throws an exception if data conversion with floating point coefficients is configured
- ASCRTE-3524 Fixed known issue: The Rte Editor removes assembly connectors between service components and application components
- Added support for combined provide and require ports as described by AUTOSAR 4.1.3
- Added support for data conversion of constants defined using an ApplicationValueSpecification in the context of an ApplicationPrimitiveDataTypes of category VALUE

- ASCRTE-3551 Fixed known issue: Missing parenthesis around Rte_IWriteRef definition
- Added partial support for serialized inter-ECU sender-receiver communication.
- ASCRTE-3561 Fixed known issue: The conversion of the schedule table activation offset into ticks may be imprecise
- ASCRTE-3547 Fixed known issue: Under some circumstances the Rte_Write API does not write the measurement buffer
- ASCRTE-3565 Fixed known issue: The Rte Bswmd generator removes certain information from data types and data prototypes which are enabled for measurement

Module version 6.1.102

2014-08-08

- ASCRTE-3416 Fixed known issue: If the callback Rte_COMCbkRxTOut_<sn> is called by Com, the Rte may not set a configured data receive error event
- ASCRTE-3386 Fixed known issue: If multiple events have been set for a BSW task, the Rte considers only one
- Added support for measurement of sender/receiver communication (intra-partition and inter-ECU), AUTOSAR-typed per-instance memory, inter-runnable variables, and parameter data prototypes
- ASCRTE-3355 Fixed known issue: The Rte Generator rejects configurations containing a `CompuMethod` of category `TEXTTABLE` that references a `Unit`
- Updated the Rte Generator to support chaining of tasks
- Updated the Rte Generator to optimize the allocation of receive buffers if they are not required for implicit communication
- Added function to disable the partition active checks
- Updated the Rte configuration to generate only one Rte schedule table for all partitions
- Updated the configuration of the Bsw task used for sending Com signals/signal groups to be time-triggered by setting the parameter `BswOsTaskPeriod`
- Added configurability to hook functions before/after Bsw schedulable entities are triggered. Additionally the Rte generates hook function before task termination
- ASCRTE-3423 Fixed known issue: Under certain conditions the software component header file does not include a required Rte header for development error checks
- ASCRTE-3415 Fixed known issue: Two client-server interfaces are not compatible if the server only defines a possible error with code 0 and the client defines other possible errors
- ASCRTE-3421 Fixed known issue: The Rte illegal invocation development error detection may use an inadequate unsigned integer type

- Removed generation of interrupt locks for Shared Memory IP channels if they are accessed by Com callbacks only and Com callbacks are configured as not interruptible
- Removed the generation of function `Rte_Det_CheckIllegalInvocation` from those partitions where the illegal invocation check is not applicable
- ASCRTE-3452 Fixed known issue: If inter-partition C/S communication is used with application errors, `Rte_Call`/`Rte_Result` return an undefined error code
- ASCRTE-3453 Fixed known issue: Under certain conditions the component data structure is not initialized

Module version 6.1.101

2014-07-14

- Added support for data conversion for primitive type data in the context of intra partition sender receiver communication and the definition of constant values in the context of application data types that are transparent to the application
- Updated verification of more elements such as AutosarDataTypes, CompuMethods and SwPointerTargetProps if the RteGeneratorOutput is set to BSW_SCHEDULER_ONLY
- Added function to configure whether only the first notification from Com shall be taken into account for transmission acknowledgment after a `Rte_Send` or `Rte_Write` request
- ASCRTE-3371 Fixed known issue: The return value of a blocking Rte_Feedback API is wrong when either a shutdown notification or a timeout occurred
- Updated the Rte Editor to correctly set the parameter `RteNvmRomBlockLocationSymbol` if the rom block location is empty
- ASCRTE-3374 Fixed known issue: If a component supports multiple instantiation the Rte implements only the Rte_Enter/Rte_Exit APIs of the first instance correctly
- ASCRTE-3379 Fixed known issue: If an array data element is used in queued sender receiver communication the Rte uses an incorrect type for the queue implementation
- ASCRTE-3156 Fixed known issue: The Rte generates non-compileable code if the short name of the BSW module description contains an underscore
- ASCRTE-3376 Fixed known issue: If the short name of certain elements are not unique the internal IDs for development error tracing may be duplicated
- ASCRTE-3387 Fixed known issue: If Rte_Stop is called the Rte may terminate a task that executes Bsw schedulable entities
- ASCRTE-3386 Fixed known issue: If multiple events have been set for a BSW task, the Rte considers only one
- ASCRTE-3381 Fixed known issue: The Rte may generate shared buffer groups incorrectly if a shared receive buffer must provide a status field

Module version 6.1.100

2014-04-24

- ASCRTE-3244 Fixed known issue: The Rte may generate mode disabling dependencies for mode switch events even if they are not specified in the software component description
- ASCRTE-3282 Fixed known issue: The `Rte_Read` API may return the wrong initial value if multiple receive ports are connected to the same provide port and different initial values are defined at the receive ports
- ASCRTE-3285 Fixed known issue: The Rte Verifier does not report an error if two variable data prototypes with equal names but incompatible data types are connected
- ASCRTE-3288 Fixed known issue: The Rte Verifier does not check the compatibility of connected variable data prototypes if the categories of the data types are different
- ASCRTE-3284 Fixed known issue: The Rte may generate incorrect code if an implementation data type of category `TYPE_REFERENCE` is used to define an invalid value in context of a sender/receiver connection
- ASCRTE-3281 Fixed known issue: If an union type is used, the Rte Generator produces code that cannot be compiled
- ASCRTE-3307 Fixed known issue: No data consistency mechanism is applied to a variable data prototype with a complex data type that is sent inter-partition inter-ECU

Module version 6.1.57

2014-03-24

- ASCRTE-3217 Fixed known issue: The `Rte_Read`/`Rte_IRead` API returns the invalid value if the invalidation policy is set to `KEEP` and the data element is invalidated
- ASCRTE-3296 Fixed known issue: The Rte does not send structure data to an intra-partition connected receiver if the names of the struct members are different on sender and receiver side
- Added support for reporting of development errors to the DET
- ASCRTE-3250 Fixed known issue: The Rte may not execute time triggered runnable entities if several timing events with both equal and different periods and mode disabling dependencies are mapped to the same task
- Removed the query of the current task ID in `Rte_Call` and `Rte_Result` APIs at runtime if the client task is well-known at code generation

Module version 6.1.56

2014-02-14

- ASCRTE-3211 Fixed known issue: The Rte Generator does not consider mappings from calibration parameters to per instance memories if an `arTypedPerInstanceMemory` is used

- ASCRTE-3200 Fixed known issue: If the timeout of a waitpoint for a data received event is zero, the Rte_-Receive API performs timeout monitoring with zero ticks.
- Changed the parameter names for the runnable function signature, if runnable entity or service arguments are specified.

Module version 6.1.55

2014-01-21

- Updated the event type (e.g. `BswTimingEvent`) to be shown in a separate table column on the Rte Editor's **Event Mapping** tab
- Implemented error report in Rte Verifier if multiple client server operations trigger the same runnable entity but define incompatible possible errors
- ASCRTE-2513 Fixed known issue: If two data received events referencing the same variable data prototype are mapped to different tasks, the Rte only considers one
- ASCRTE-3161 Fixed known issue: The Verifier wrongly reports an error if a server port of a composition is not implemented by a runnable entity
- ASCRTE-3177 Fixed known issue: If synchronous intra-partition client/server-communication is used and a server runnable of category 2 is mapped to a task, the client may receive a wrong result
- Added missing trace hook functions (e.g. for `Rte_IsUpdated`) and changed trace hook signature of `Rte_IRead`
- Updated compiler abstraction macros
- ASCRTE-3168 Fixed known issue: If the Rte editor is opened and the elements mode declaration, mode declaration group or mode declaration group prototype are not complete, an exception is reported

Module version 6.1.54

2013-11-15

- ASCRTE-3099 Fixed known issue: If a runnable entity implicitly sends a variable data prototype with invalidation policy set to keep or replace in case of inter-partition inter-ECU communication, then the Rte Generator fails
- ASCRTE-2843 Fixed known issue: If a runnable entity with a blocking Rte API function is not mapped to a task, the Rte Generator crashes with an exception
- ASCRTE-2801 Fixed known issue: If an implementation data type of category structure contains an implementation data type element referencing its parent implementation data type the Rte Generator causes a stack overflow error

- ASCRTE-3106 Fixed known issue: An executable entity is never executed at runtime if it is triggered by a timing event that has an activation offset equal to its period which in turn is a multiple of the cycle length of the task to which it is mapped to
- Improved functionality that starting a schedule table with relative activation mechanism no longer fails if the configured schedule table offset is shorter than one Os counter tick. The minimal offset value `1` is used in this case
- ASCRTE-2947 Fixed known issue: The Rte does not support sender receiver to signal group mappings for array/record application data types
- Fixed known issue: Violations against the AUTOSAR meta model might be reported twice or elements are verified that are not used by the Rte generator.
- Updated the table columns **Period** and **Offset** on the Rte Editor's **Event mapping** tab to use numerical instead of alphabetical sort algorithm

Module version 6.1.53

2013-10-22

- Updated the Rte Verifier to allow client/server application errors with `errorCode 0`. If the short name of the application error is not `E_OK`, though a warning is reported
- Updated the Rte Verifier to allow operation-invoked events triggering the same runnable which have array typed arguments of different length (see TPS_SWCT_1125)
- Changed the Rte types header file to use the compiler abstraction macros
- ASCRTE-3049 Fixed known issue: The Rte Generator shares internal buffers for non-queued implicit inter-partition sender/receiver communication across partition boundaries which leads to non-compilable code
- ASCRTE-3053 Fixed known issue: Mode-disabling dependencies are not working if more than eight mode-disabling dependencies for different events are specified for an inter-partition mode machine instance

Module version 6.1.52

2013-09-18

- ASCRTE-2963 Fixed known issue: If the enhanced `Rte_Mode` API is configured for a software component which supports multiple instantiation or uses the indirect API, the generated Rte code does not compile
- ASCRTE-2931 Fixed known issue: The Rte editor removes connections between two service components
- ASCRTE-2920 Fixed known issue: The Rte Editor does not group elements of `requiredModeGroup` correctly in the BSW Mode Mapping window

- ASCRTE-2944 Fixed known issue: If `TS_IntDisableEnable` is configured as the interrupt blocking function in a shared memory inter-partition environment, the generated code does not compile
- ASCRTE-2934 Fixed known issue: If only the Bsw scheduler is generated and the Os parameter `OsSecondsPerTick` is not configured, then the Rte fails with an `ArithmeticException`
- ASCRTE-2976 Fixed known issue: The Rte Generator fails if an inter runnable variable is typed by an application data type which is not mapped to a corresponding implementation data type
- ASCRTE-3004 Fixed known issue: The Rte generates the API infix in upper case letters in memory mapping and compiler abstraction macros
- Updated the Rte Generator to support the category `EXPLICIT_ORDER` for mode declaration groups

Module version 6.1.51

2013-08-20

- ASCRTE-2921 Fixed known issue: If only on-transition runnables are configured for one mode machine instance and no mode disabling dependencies are configured, the generated Rte code does not compile
- ASCRTE-2911 Fixed known issue: If data mappings are distributed among different system mappings, the Rte Editor does not save the data mapping correctly
- ASCRTE-2988 Fixed known issue: If timing events have a huge period, the Rte configures an empty schedule table in the Os ECU configuration
- ASCRTE-2914 Fixed known issue: If a system signal is mapped to a data element, but no Com signal references this system signal, the Rte Generator fails
- ASCRTE-2932 Fixed known issue: The **Data Mapping** tab of the Rte Editor does not show application record elements of category `BOOLEAN`
- ASCRTE-2940 Fixed known issue: The generated `Rte_IrvRead` and `Rte_IrvWrite APIs` ignore the given parameter if the inter-runnable variable is of complex type and the APIs are implemented as a C-macro
- Updated the Rte Generator to support `Rte_IsUpdated` for inter-partition and inter-partition/inter-ECU communication
- The Rte Verifier now reports an error if not all mode users of a provided mode declaration group are on the same partition
- ASCRTE-2938 Fixed known issue: The Rte Generator does not generate empty module interlink header files

Module version 6.1.50

2013-06-25

- ASCRTE-2824 Fixed known issue: In task mapping scenario B2, the generated Rte code might not contain all events required by `Rte_WaitGetClearEvent()`
- ASCRTE-2697 Fixed known issue: Initial values for data elements of array types are not written to the Com configuration via the service needs assistant
- ASCRTE-2860 Fixed known issue: The generated Rte may not compile if the value of a text value specification contains a number of characters that is greater or equal to the array size of the underlying data type
- ASCRTE-2814 Fixed known issue: The Rte Editor reports an error if data prototype names collide with Rte Editor internal short names
- Updated the generated template code in the folder `src_appl` to use a pointer to the array base type in case of array data types

Module version 6.1.13

2013-05-17

- Updated the Rte editor to ignore a local variable access which does not refer a valid local variable when opening the **VFB Tracing** tab
- ASCRTE-2815 Fixed known issue: If at least two mode switch points reference the same mode declaration group instance and one requires a blocking `Rte_SwitchAck` API while the other one is not referenced by a mode switch acknowledge event, the Rte generates no code

Module version 6.1.12

2013-04-17

- Updated the Rte Editor tab **Service Port Mapping** to allow to connect one application port to multiple service ports and vice versa
- ASCRTE-2555 Fixed known issue: `Rte_Pim()` incorrectly returns a pointer to array type instead of pointer to the array base type for a per instance memory typed by an AUTOSAR data type
- ASCRTE-2529 Fixed known issue: The code which is generated in the contract phase depends on EB-specific header files
- ASCRTE-2640 Fixed known issue: The Rte does not generate any data filters if the underlying C-type is a char type
- Updated the Rte Verifier to report a warning if the resolution of the parameter `OsSecondsPerTick` of the used Os counter is higher than nano-seconds
- Updated the Rte Verifier to report an error if a configured data filter lacks required attributes, e.g. the `MIN` and `MAX` attributes

- ASCRTE-2672 Fixed known issue: The Rte generates incorrect code in case of intra-partition mode switch procedure
- ASCRTE-2450 Fixed known issue: If inter-partition mode management with IOC is used and no mode switch event is specified, the mode disabling dependencies are not set by the `Rte_Switch` API
- Updated the Rte mode defines to generate even if the corresponding mode declaration group is not used, but an included mode declaration group set is referencing the mode declaration group
- ASCRTE-2672 Fixed known issue: The Rte generates incorrect code in case of intra-partition mode switch procedure
- ASCRTE-2675 Fixed known issue: If a mode-disabling dependency references a mode declaration that is not part of the referenced mode declaration group prototype, the Rte Generator aborts with an exception
- ASCRTE-2685 Fixed known issue: If multiple clients are connected to the same server and an unconnected port triggers the same server runnable or the client operations consist of different argument, the Rte might report a null pointer exception or the generated code does not compile
- ASCRTE-2699 Fixed known issue: If the Rte Generator is executed multiple times without restarting EB tresos Studio, the Rte generates incorrect mode-disabling dependency macros
- ASCRTE-2745 Fixed known issue: The enhanced `Rte_Mode` API function does not set the given previous and next mode parameters if function elision is configured
- Updated the Rte to support the execution of runnable entities and BSW schedulable entities on the transition of two modes
- Added the Rte Editor and the plugin `SwcUtils` to the Rte. The release notes for both plugins are now within the release notes of the Rte
- Updated the combo boxes **Bsw Os application** and **BSW Os task** to always show the values of the Rte configuration after opening the Rte Editor page **Partitioning**.
- ASCRTE-2727 Fixed known issue: The Bsw mode machine instance is not correctly initialized and `Rte_Start()` overwrites the state of the mode machine instance
- ASCRTE-2733 Fixed known issue: The Rte does not generate enumeration macros for application data types that are referenced by an included data type set

Module version 6.1.11

2013-02-08

- Updated the system description verifier to check for a proper Bsw mode management configuration as described by constr_4022, constr_4059, and constr_4063
- Added support of the BSW Scheduler name prefix
- ASCRTE-2523 Fixed known issue: If an initial value is specified for an implementation data type that references a structure, the Rte Generator wrongly reports an error

- Added support of background events
- Added support of Rte mode switch acknowledgment
- ASCRTEAS-431 Fixed known issue: The Rte Editor does not show elements of `SwcServiceDependency` which are mapped to per-instance memory
- ASCSWCUTILS-191 Fixed known issue: If an implementation data type of category `value` does not specify a base type, the Rte Generator reports a null pointer exception
- Added support of the enhanced mode API for inter-partition mode management
- ASCRTE-2478 Fixed known issue: If function elision is active and the port for a specific API function is unconnected, the Rte generates the return value `RTE_E_OK`

Module version 6.1.10

2012-12-14

- Updated the Rte to use the pointer to array type passing scheme by default if the configuration parameter `ASR32RteWrapper`is set to true,
- ASCRTE-2458 Fixed known issue: If the enhanced Mode API is used and the corresponding port is unconnected, the Rte Generator does not generate any code
- ASCRTE-2306 Fixed known issue: The Rte Generator may produce duplicate code blocks for reading a signal group from COM within a reception callback
- ASCRTE-2452 Fixed known issue: The Rte Generator does not generate any code if a per instance memory is typed by an application data type
- Updated the Rte Generator to support elements of `InterRunnableVariable` with complex data types
- Added generation of enumeration constants for referenced application primitive data types to the application types header file
- ASCRTE-2479 Fixed known issue: The Rte cannot allocate more than one Os schedule table expiry point per millisecond

Module version 6.1.9

2012-11-19

- ASCRTEAS-435 Fixed known issue: Rte Editor removes configured BSW Module instances from Rte configuration under certain circumstances
- Updated the editor fields **BSW Os task**, **BSW send signal queue length**, and **BSW send signal group queue length** on the **Partitioning** tab to be properly enabled depending on the check box value **BSW Os task required**

- ASCSWCUTILS-175 Fixed known issue: The compatibility check of two elements of `ImplementationDataType` or `ImplementationDataTypeElement` of category `ARRAY` is incorrect
- ASCRTE-2172 Fixed known issue: The Rte Generator generates non-compilable code when a local parameter is mapped to per-instance memory, but no parameter access is defined
- ASCRTEAS-439 Fixed known issue: If the Rte configuration is saved and the **Measurement and Calibration** tab was not touched, the calibration support configuration for parameter software component types is removed
- ASCRTE-2441 Fixed known issue: If partitioning is enabled, the Rte does not generate invocations of executable entities mapped to the BSW Os task
- ASCRTE-2442 Fixed known issue: If basic software (BSW) events are mapped to an OS task that is not part of the BSW partition, the OS task implementation is added to the BSW partition
- Updated the Rte Generator to support data element invalidation if the initial value equals the invalid value
- Updated enumeration constants for elements of `CompuMethod` to be generated as defined by AUTOSAR 4.0.3

Module version 6.1.8

2012-10-15

- ASCSWCUTILS-167 Fixed known issue: If it is checked whether a variable data prototype is queued or not, the data type map should not be considered if the variable data prototype references an implementation data type
- ASCRTE-2408 Fixed known issue: The behavior of the Rte Generator is undefined when invalid `SwBaseType` native declarations are present in the data model
- ASCRTE-2386 Fixed known issue: The system model is not validated if the Rte Contract Phase Generator is executed via the command line
- Updated the system description verifier to check if the native declaration of a `SwBaseType` contains a valid C-data type
- Updated the Rte Generator to support the Basic Software Scheduler Mode Management as defined by AUTOSAR 4.0
- Updated the Rte Editor to support connecting of BSW required triggers with BSW released triggers
- ASCRTE-2412 Fixed known issue: Partial record support does not function in combination with system signal mappings
- Updated macro definitions for range data types, enumeration constants and invalid values according to AUTOSAR 4.0.3
- Fixed several issues with the component template generator

- Updated the Rte Editor to support mapping of provided BSW mode declaration group prototypes to required BSW mode declaration group prototypes
- Updated the Rte Generator to verify the Bsw Scheduler ECU configuration
- ASCRTEAS-425 Fixed known issue: The **Service Port Mapping** tab removes all port interface mappings if a service connector is modified
- Updated the Rte Generator to support the enhanced mode API for intra-partition mode management

Module version 6.1.7

2012-09-18

- ASCRTE-2387 Fixed known issue: If an inter-runnable variable is typed by an application data type which does not have a mapped implementation data type, the Rte Generator will prematurely terminate without errors
- ASCRTE-2363 Fixed known issue: The system signal mapping is not working when a port interface mapping is specified
- ASCRTE-2313 Fixed known issue: The Rte Generator may produce code which causes compiler warnings like *Rte_IsModeDisablingDepSet_X defined but not used*
- Updated the Bsw Scheduler to support the `SchM_ActMainFunction` and `SchM_Trigger` API functions

Module version 6.1.6

2012-08-17

- Updated the Bsw Scheduler to support Bsw called entities and Bsw interrupt entities
- Updated the names of the preprocessor defines for the Com Handle IDs according to AUTOSAR 4.0 Rev 3 naming scheme
- Updated the system description verifier to check Bsw schedulable entities, Bsw called entities and Bsw interrupt entities
- ASCRTE-2365 Fixed known issue: The Rte Generator may incorrectly initialize Rte data structures when the initial value is greater than `0x7FFFFFFF`
- ASCRTE-2368 Fixed known issue: The Rte Generator throws a null pointer exception when a software base type is incorrectly defined
- Improved the evaluation of task mapping for asynchronous client/server calling chains
- Updated the system description verifier to report an error if two different interfaces with the same short name have incompatible application errors or if they are referenced by a software component which supports multiple instantiation or the indirect API attribute is set to true for at least one port

Module version 6.1.5

2012-07-16

- Removed configuration switch for the AUTOSAR 3.1 SchM Exclusive Area API support
- ASCRTE-2154 Fixed known issue: Inter-partition mode disabling dependencies do not work with non-timing events
- Updated the Bsw Scheduler to generate an AUTOSAR 3.1 SchM Exclusive Area API wrapper for individual Bsw module instances
- Updated the Rte Editor to disable exclusive areas with the implementation mechanism option *Disabled Exclusive Area*
- Removed the AUTOSAR 2.1 to AUTOSAR 3.1 Rte module transformer from the Rte add-on
- ASCRTE-2341 Fixed known issue: Wrong registration of transformer causes error log entry in Studio when using module upgrade functionality

Module version 6.1.4

2012-06-21

- ASCRTEAS-411 Fixed known issue: The Rte Editor does not report any errors if the BSWMD is invalid
- ASCRTE-2332 Fixed known issue: Warnings from the Rte Generator are not reported to the user
- Updated the Rte Generator to support data mappings to system signals
- Updated the Rte to support the usage of partial records (signal degradation) for sender-receiver communication as specified by AUTOSAR 3.1
- Updated the Rte editor to support the mapping of system signals and system signal groups to data elements
- ASCRTE-2322 Fixed known issue: If a runnable entity makes an inter-partition server call and the runnable is not mapped to a task, the Rte Generator reports an error even if a SWC to BSW mapping exists for that runnable
- Updated the Rte Generator to provide the generation of the AUTOSAR 3.2 Rte Wrapper to support AUTOSAR 3.2 software components
- Updated the Rte Editor to offer a configuration switch to enable or disable the AUTOSAR 3.2 Rte Wrapper feature
- Removed legacy support of the ASR 3.1 SchM Exclusive Area API

Module version 6.1.3

2012-05-16

- ASCRTE-2288 Fixed known issue: The generated Rte will not compile if a basic software module description name and a software component type name are equal
- Updated the Rte Generator to support the generation of empty `Rte_Start`/`Rte_Stop` functions if the configuration switch **Generate empty Rte_Start/Rte_Stop stubs** is enabled
- Updated the Rte Editor to offer a configuration switch to enable or disable the generation of an empty `Rte_Start`/`Rte_Stop function`
- ASCRTE-2297 Fixed known issue: The Rte Generator may produce duplicate code blocks for implicit reads and writes within the same task
- Updated the system description verifier to handle partial records and subelement mappings
- ASCRTE-2307 Fixed known issue: The Rte Generator may not consider the task position of timing events mapped to the same task

Module version 6.1.2

2012-04-23

- ASCRTEAS-398 Fixed known issue: The configuration of the partitioning support may not be possible, even if the partitioning support is enabled
- ASCRTEAS-399 Fixed known issue: The drop-down list in the tables of the **Implementation Selection** and **Partitioning** tabs show the wrong list of items
- ASCSWCUTILS-116 Fixed known issue: The system description verifier does not generate an error when the offset and the period of a `ONEEVERYN` data filter are equal
- ASCRTE-2192 Fixed known issue: The Rte Generator will generate a non-blocking `Rte_Result` API function, if an asynchronous server call result point is defined, but no corresponding asynchronous server call returns event exists
- ASCRTE-2154 Fixed known issue: Inter-partition mode disabling dependencies do not function with non-timing events
- ASCRTE-2215 Fixed known issue: The Rte Generator will produce a wrong type definition for an implementation data type of category `STRUCTURE` that contains an implementation data type element of category `ARRAY`
- Changed implementation data type elements of category `ARRAY` with the basetype `uint8` to be grouped to byte arrays
- ASCRTE-2287 Fixed known issue: The memory mapping in the Module Interlink header files may cause linker errors

Module version 6.1.1

2012-03-23

- ASCRTE-2178 Fixed known issue: If a data write access is specified for a variable data prototype which is not of category `ARRAY`, the resulting `Rte_IWriteRef` API will not return a pointer to the variable
- ASCRTE-2009 Fixed known issue: The Rte Generator may produce code which causes compiler warnings like *a value of type `x` cannot be used to initialize an entity of type `y`*
- ASCRTE-2189 Fixed known issue: The Rte Generator may generate code that will not compile due to unresolved symbols
- ASCRTE-2102 Fixed known issue: If a timeout is specified for an asynchronous server call point and no wait point is defined for the corresponding asynchronous server call result point, then the Rte Generator will produce non-compilable code
- Updated the Rte Generator to support the generation of the legacy AUTOSAR 3.1 SchM Exclusive Area API as function-like macros
- ASCRTE-2193 Fixed known issue: The generated Rte code will produce a compile error, if a Com signal group is mapped to a variable data element at a require port and the data can be directly read from Com
- Updated the Rte Editor to offer a configuration switch for enabling or disabling the AUTOSAR 3.1 SchM Exclusive Area API support
- ASCRTE-2196 Fixed known issue: The Rte Generator may produce a `NullPointerException`, if two tasks from different partitions hold runnable entities that specify a timeout for their `Rte_Call`, `Rte_Feedback`, `Rte_Receive` or `Rte_Result API`
- ASCRTE-2200 Fixed known issue: The `Rte_Start()` API will not correctly initialize the Rte on a multi-core system
- ASCRTE-2177 Fixed known issue: The Rte Contract Phase will fail without any error if an application data type is not mapped to an implementation data type
- Reception signal timeout callbacks no longer read Com signal values when the attribute *replace by initial value* is not configured for a Com signal
- Updated the Rte Editor to provide the feature to auto-map all Bsw timing events to a task
- Updated the Rte Generator to support the `SwcBswMapping` to determine the calling context for Bsw schedulable entities
- ASCSWCUTILS-130 Fixed known issue: The merge of the ECU resource properties fails with an error during the Rte generation or when running the Service Needs Assistant
- ASCRTE-2206 Fixed known issue: The generated Rte code will not compile if two or more runnable entities are mapped to different tasks and each of these has a synchronous server call point for the same operation on the same port
- ASCRTE-2197 Fixed known issue: Within the Rte default schedule table, the Rte Generator will not allocate any expiry point, which activates an Os task if there already exists an expiry point activating any other Os task with the same period and offset
- Updated the Rte Generator to support the generation of the BSW scheduler and its API functions as defined by AUTOSAR 4.0

- Implemented generic BSW scheduler verifier
- Updated the Rte Editor to support configuring the BSW scheduler specific parts in the Rte

Module version 6.1.0

2012-02-17

- ASCRTE-2113 Fixed known issue: If a client port is connected to a server port, which belongs to the same software component instance, the generated Rte code will produce compiler warnings
- Updated the system description verifier to consider the port interface mapping if the compatibility between two interfaces is checked
- Updated the Rte Generator to support the port interface mapping as defined by AUTOSAR 4.0
- ASCRTE-2173 Fixed known issue: `Rte_NvMData.h` can cause compile errors due to the duplicate definition for the same ROM default value variable
- Added support for the `ImplementationDataType` attribute `typeEmitter`
- Removed support of the Rte Editor for the generation of the service component description via the **Service Port Mapping** tab.

Module version 6.0.12

2012-01-20

- ASCSWCUTILS-107 Fixed known issue: If a delegation connector connects ports of different types, the system description verifier will stop the verification of the system model and report a class cast exception
- ASCRTE-2013 Fixed known issue: If inter-partition sender/receiver communication is used with data filters configured with the data filter algorithm `NEVER`, the Rte Generator produces code which causes compiler warnings
- ASCRTE-2115 Fixed known issue: The Rte Generator will report an error for application data types that refer to a valid application value specification
- ASCRTE-2121 Fixed known issue: The Rte will not update the mode of a mode machine instance after the execution of a related mode switch event
- ASCRTE-2150 Fixed known issue: The Rte Generator will produce compiler errors like *parse error before '}' token* if a runnable entity defines a server call or server call result point for an operation, for which no operation invoked event exists on server side
- Updated the system description verifier to check `CompuMethod` elements of the category `TEXTTABLE` according to the software component template constraint constr_1134
- Implemented partitioning support in the variant *Shared Memory*

Module version 6.0.11

2011-12-09

- Added additional consistency checks
- ASCRTE-1193 Fixed known issue: An `Rte_Write`, `Rte_Feedback` or `Rte_Invalidate` might behave as if the port was not connected
- ASCRTE-2088 Fixed known issue: When a DBC, FIBEX or LDF file is imported with the option *Enable system model import* enabled, the Rte Generator will fail to generate the Rte
- Added generic Rte Verifier

Module version 6.0.10

2011-11-11

- ASCRTE-2007 Fixed known issue: The Rte Generator produces non-compilable code when inter-ECU invalidation shall be handled by the Rte and the invalid value of a COM signal which shall be invalidated is specified in binary or array format
- ASCRTE-2020 Fixed known issue: If there are `CompuMethod` elements with category `TEXTTABLE` where at least one `CompuScale` defines no upper or lower limit, the Rte Generator will produce a null pointer exception
- ASCRTE-2018 Fixed known issue: If inter-partition client/server communication is used, the API function `Rte_Result()` will not return
- ASCRTE-2031 Fixed known issue: If optimization function elidation is enabled, the Rte will transmit a primitive data element only within its own partition
- ASCRTE-2008 Fixed known issue: When the optimization *use bit fields for internal buffers* is enabled, the Rte Generator may produce non-compilable code
- ASCSWCUTILS-91 Fixed known issue: The system description verifier fails to report an error for incorrect instance references, which refer to an entity that is not part of the enclosing software component type
- Updated the Rte Generator to support the update flag feature defined by AUTOSAR 4.0
- Updated the Rte Generator to support the Rte_`DRead` API function as defined by AUTOSAR 4.0
- Changed the value of the error code `RTE_E_SHUTDOWN_NOTIFICATION`
- Updated the Rte Generator to support the never received status for sender/receiver communication
- ASCRTE-2063 Fixed known issue: The Rte Generator may produce code that causes compiler warnings if `USE-VOID` is used as `ServerArgumentImplPolicy` for a complex `IN` argument of an operation
- ASCRTEAS-340 Fixed known issue: The Rte Editor will not save any changes which have been applied to the **General** tab if the partitioning support, event mapping, or service port mapping is configured afterwards

- Changed the content of the API mapping and declaration in the component header file to not be re-ordered if the configuration is unchanged

Module version 6.0.9

2011-09-02

- Initial AUTOSAR 4.0 version

2.2. New features

- The Rte now supports multiple Com instances.

  It is now possible to send/receive signal(group)s over multiple Com instances which are mapped to different partitions.

- The Bsw Scheduler now supports client/server communication.

  Bsw modules can now exchange variable data over the Bsw Scheduler by using the SchM_Call and SchM_Result APIs.

- The Rte now supports autonomous error reaction for inter-ECU client server communication with data serialization

  For client server communication it is now possible on the server side to trigger an autonomous error reaction which generates the response of the client server communication automatically without involvement of any runnable.

- The Rte now supports the asynchronous mode switch

  It is now possible to configure the asynchronous mode switch communication. If all intra-partition require mode declaration group instances support the asynchronous mode switch behavior and if the mode switch events are either mapped to no task or if all of them are mapped to the task of the `Rte_Switch`, then the invocation of runnable entities will be made via direct function call.

- The Rte now supports intra-ECU sender-receiver communication with data transformation

  For intra-ECU sender-receiver communication it is now possible to specify a DataTranformation between two interfaces by using a DataPrototypeMapping

- The Rte now supports the dirty flag mechanism for NvBlockSwComponentTypes

  It is now possible to enable the `dirtyFlagSupport` for a `NvBlockDescriptor`. Different writing strategies like `storeImmediate`, `storeCyclic` and `storeAtShutdown` can be configured in the `NvBlockNeeds`.

- The Rte now supports memory mapping initialization strategy

For inter runnable variables, NvRam blocks, per instance memory, receive buffers and Smc buffers it is now possible to specify an initialization strategy mapped to a policy described by a SwAddrMethod corresponding to the variable data prototypes.

- The Rte now supports the external replacement invalidation for sender-receiver communication according to AUTOSAR 4.3.0.

  For sender-receiver communication it is now possible on reception side to specify another variable data prototype to be used for replacement.

- The Rte now supports the feature to provide the activating event of an ExecutableEntity according to AUTOSAR 4.3.1.

  It is now possible to configure `ActivationReason` for an `RTE/BSW Event`. When configured, `RTE` shall pass an additional argument `activation` to the `ExecutableEntity`.

- The Rte now supports debounced activation of executables.
- The Rte now supports compound primitives

  It is now possible to use compound primitives as long as their implementation is an array.

- The Rte now supports InitEvents triggering runnable entities via OS task according to AUTOSAR 4.3.1.

  It is now possible to configure `InitEvents` triggering runnable entities via `OS task` for initialization purposes, i.e. for starting and restarting a partition.

- The Rte now supports Transport Protocol for Serialization for LdCom.
- The Rte now supports the implementation of a basic task for tasks shared by Rte and Bsw timing events.

  It is now possible to enable a `OneScheduleTablePerPartition` option to implement a basic task for tasks shared by Rte and Bsw timing events.

- The Rte now supports mapping of external trigger events for category 1 executables to category 2 ISRs.
- The Rte now supports RTE Implementation Plugins (RIPS) for explicit/implicit sender receiver communication.
- The Rte now supports InitEvents triggering runnable entities via RteInitializationRunnableBatch according to AUTOSAR 4.3.1.

  It is now possible to configure `InitEvents` triggering runnable entities for initialization purposes via `RteInitializationRunnableBatch`. i.e. for starting and restarting a partition.

- The Rte now supports the generation of human readable global variable names instead of cryptic ones

  It is now possible to set the new configuration parameter `HumanReadableBufferNames` to true to let the Rte generator use a human readable naming approach for the generated global variables.

- The Rte now supports timeout monitoring for asynchronous client/server communication with non-blocking result calls.

- The Rte now supports partitioning for communication timeout in case of inter-partition inter-ecu communication
- The Rte now supports multiple mode user partitions.

  It is now possible to configure a mode manager connected to multiple mode user which are mapped to different partitions which in turn are scheduled on different micro-controller cores.
- The Rte now supports intra-ECU sender-receiver communication with SubElementMappings to primitive DataPrototypes.

  It is now possible to map a structure member on a sender side to a primitive data element on the receiver side.
- The Rte now supports the usage of LdCom without transformers.
- The Rte now supports mapping of timing events for category 1 executables to category 2 ISRs.

  It is now possible to map a timing event or a BSW timing event to an ISR.
- The Rte now supports transmission acknowledgment for implicit sender receiver communication.

  The information whether a value has been successfully passed to the communication infrastructure can be known by using the Rte_IFeedback API.
- The Rte now supports synchronous mode switching procedure over multiple partitions/cores.

  It is now possible to have the mode switching procedure steps synchronized over multiple partitions/cores.
- The Rte now Supports mapping of variable data prototypes to parameter data prototypes.
- The Rte now supports the access to a reference to the VariableDataPrototype referenced by VariableAccess in the dataReadAccess role.

  This can be done by using the Rte_IReadRef API.
- The Rte now supports retrieving partition context for Com on a per-PDU basis.
- The Rte now supports retrieving partition context for LdCom on a per-PDU basis.
- The Rte now supports graphical visualization for exclusive area optimization.
- The Rte now supports the improvement of start-up and initialization of NvBlockSwComponents. Restore default data if both ROM and an initialization callback are configured.
- The Rte now supports the access to a reference to the VariableDataPrototype defined with the implicitInterRunnableVariable role.

  This can be done by using the Rte_IrvIWriteRef API.
- The Rte now supports the transmission preparation callback (Rte_COMCbkTxPrep_mn) for inter-partition inter-ECU communication.
- The Rte now supports the generation of review instructions to easily identify memory writes access for freedom from interference analyses (released).

The configuration parameter `GenerateRteFreedomFromInterferenceReviewInstructions` can be used to generate review instructions and an html report to easily detect where a write access occurs.

- The Rte now supports the I-PDU based time-out according to SWS_Rte_08103.

  For all signals/signal groups without update bits the timeout occurs according to the greatest multiple of the I-PDU based time-out value lower than or equal to the aliveTimeout value in the NonqueuedReceiverComSpec.
- The Rte now supports the generation of implicit sender-receiver APIs as C Function when enableTakeAddress for the port is set to true.

2.3. Elektrobit-specific enhancements

This chapter lists the enhancements provided by the module.

- On demand, you are able to add the OS objects required by the Rte automatically to the OS configuration using the Service Needs Assistant.
- On demand, the Service Needs Assistant automatically configures the required Com callbacks in the Com configuration.
- The Rte provides an additional configuration option *OSEK OS compatibility mode*. This option enables the usage of the Rte in connection with an operating system which is not compliant to AUTOSAR, but only compliant to OSEK OS.
- When applying data consistency mechanisms (Os resources, interrupt locking etc.), the Rte Generator considers whether the hardware can make an atomic access. In this case, no data consistency mechanism is applied.
- You are able to configure the data consistency mechanism which the Rte Generator shall apply if a data consistency mechanism is required. You can choose between interrupt locking and the usage of Os resources.
- You are able to configure the interrupt blocking function which the Rte Generator shall use. The following options are provided: `SuspendResumeAllInterrupts`, `DisableEnableAllInterrupts`, fast interrupt locking (EB-specific) or you can use a user-specific interrupt locking function.
- You can configure whether the Com callbacks are interruptible or not. If they are not interruptible, the Rte Generator does not need to lock interrupts at all in the Com callbacks.
- When allocating buffers for data elements, the Rte Generator shares buffers if possible to reduce the RAM consumption.
- The Rte Generator supports function elidation. If function elidation is enabled, several API functions are realized as macros instead of functions.

- The Rte Generator provides an option to directly read data from the Com module if possible. If this option is switched on, the Rte Generator does not allocate an additional receive buffer. This reduces the RAM consumption. You may disable this option if your application reads the data element more frequently than the mapped signal is updated by Com.
- The Rte supports additional trace hooks for the implicit `IRead` and `IWrite` API.
- You can use the component-specific memory mapping if you want to map the code of your software components to other memory sections than the default memory section `RTE_APPL_CODE`.
- In addition to the specified return values, the blocking Rte API calls return `RTE_E_SHUTDOWN_NOTIFICATION` if the Rte received a shutdown notification:
  - `Rte_Receive`
  - `Rte_Feedback`
  - `Rte_Call` (synchronous invocation)
  - `Rte_Result`

  This happens while the Rte waits for the wait point or synchronous server call point to be resolved.
- You can configure whether the Rte shall handle inter-ECU signal invalidation or not. If the Rte shall not handle inter-ECU invalidation, it is handled by the Com as defined in the specification.
- The Rte Generator now supports the partitioning of software components. Software components and their runnable entities can be mapped to different partitions. The Rte realizes the communication between software components mapped to different partitions using the Inter OS Application Communicator (IOC) which is part of the Os or the Shared Memory Communicator (SMC) which is part of the Rte.

  The partitioning support enables the usage of the Rte in multi-core and memory-protected systems.
- It is possible to have up to 256 clients per operation in a client/server communication.

  The Rte Generator allows to connect 256 clients with the same server, i.e. operation.
- The Rte Generator used different Os events for timing events, that were mapped to the same task, had the same period, but different mode disabling dependencies.

  Now, the Rte Generator uses only one Os event for the timing events of a task, if they have the same period, but different mode disabling dependencies.
- The Rte supports the usage of partial records (signal degradation) for sender/receiver communication. This increases the flexibility of a sender-receiver interface by allowing new elements to be added to a record, while still being compatible. See Bugzilla issue http://www.autosar.org/bugzilla/show_bug.cgi?id=44863.
- It is possible to have multiple wait points that reference the same data send completed event or mode switched ack event. This implies that runnable entities mapped to different tasks can now call a blocking `Rte_SwitchAck` or `Rte_Feedback` at the same time.
- The Rte generator now supports `Rte_IsUpdated` for inter-partition and inter-partition/inter-ECU communication.

- The Rte provides an additional configuration parameter `SingleScheduleTablePartitionRef`. If this parameter is set and multiple partitions are used, only one schedule table will be generated for the referenced Os Application. Therefore a call to `Rte_Start`/`Rte_Stop()`/ `Rte_Restart()` in one of other partitions has no effect on the lifetime of the schedule table.
- The Rte provides an additional configuration parameter `DisablePartitionActiveChecks`. If set to true, the Rte will not fulfill requirements rte_sws_2538, rte_sws_2535, and rte_sws_2536 anymore. This means that the Rte does not check for each API function if the current partition is active. Ensure that no callbacks and no API functions are called before `Rte_Start()`/after `Rte_Stop()` has been executed. The effect of this optimization depends on the number of generated API functions and how often they are called by the application.
- The Rte generates further hook functions than specified by AUTOSAR 4.0.3:

  At the end of each task, the hook function `Rte_Task_EndHook(OsTask)` is called.

  Before each Bsw schedulable entity call, the hook function `Rte_Schedulable_<bsnp>[_<vi>_<ai>]_Start()` is generated.

  After each Bsw schedulable entity call, the hook function `Rte_Schedulable_<bsnp>[_<vi>_<ai>]_Return()` is generated.

  The header file `SchM_<bsnp>[_<vi>_<ai>]Hook.h` contains all hook functions accessible by that basic software module.
- The Rte supports combined provide and require ports as described by AUTOSAR 4.1.3:

  The Rte generator now supports combined provide and require ports (i.e. `PRPortPrototype`s) by means of AUTOSAR 4.1. Therefore the following additional requirements are fullfilled by the Rte verifier and generator:

Requirement / constraint	Origin	Document Version
SWS_Rte_06030	Specification of RTE	3.5.0
constr_1200	Software Component Template	4.5.0
constr_1202	Software Component Template	4.5.0
constr_1203	Software Component Template	4.5.0
constr_1204	Software Component Template	4.5.0
constr_1205	Software Component Template	4.5.0

Table 2.1. Introduced requirements for PRPortPrototypes

NOTE

The Rte considers an `AssemblySwConnector` between two `PRPortPrototype`s as a directed connection where the direction is specified by means of the provider and requester role. If a bi-directional connector is needed, two `AssemblySwConnector`s for each direction must be specified.

- The Rte supports the concept of Data Transformation via IF-API according to AUTOSAR version 4.2.-2. A list of supported scenarios is documented in section *Supported features* of the EB tresos AutoCore Generic 8 RTE documentation.
- The Rte supports fire-and-forget asynchronous client/server calls as specified by RfC 70295 (planned for AUTOSAR version 4.3.0). It is not required anymore to define an AsynchronousServerCallResultPoint for each AsynchronousServerCallPoint. The Rte generates no Rte_Result API if the AsynchronousServer-CallResultPoint is omitted. This enables the client to initiate a subsequent asynchronous server call without the need to first fetch the result as soon as the server processed the last request.
- The Rte supports cooperative tasks which is a list of OsTasks that cannot interrupt each other regardless of their configured priorities, schedule, application assignment, and core assignment. This list will be taken into account by the Rte generator to optimize locks and to reduce the number of implicit buffers.
- The Rte supports the dirtyFlag mechanism as described by AUTOSAR 4.3.0.

  The following additional requirements are fullfilled by the Rte generator: SWS_Rte_08080, SWS_-Rte_08081, SWS_Rte_08082, SWS_Rte_08083, SWS_Rte_08084, SWS_Rte_08085, SWS_Rte_08086, SWS_Rte_08087, SWS_Rte_08088, SWS_Rte_08089, SWS_Rte_08090.
- The Rte supports the InitEvents via OS task as described by AUTOSAR 4.3.1.

  The following additional requirements are fullfilled by the Rte generator: SWS_Rte_06748, SWS_Rte_-06761, SWS_Rte_06762.
- The Rte supports the configuration parameters RteServerQueueLength and RteBswServerQueueLength as specified by RfC #79150. This enables projects to specify the server queue length for BSW module client-server communications and to overrule the queueLength attribute of the ServerComSpec for Rte client-server communications. The RteServerQueueLength resp. RteBswServerQueueLength can be configured in the event-to-task-mapping configuration for the particular OperationInvokedEvent which starts the server executable.
- The Rte supports the configuration parameters RtePeriod and RteBswPeriod as suggested by RfC #80049. This enables the time triggered activation of executable(s) assigned to an RteEvent/BswEvent and disables any event triggered activation. The RtePeriod resp. RteBswPeriod can be configured in the event-to-task-mapping configuration for the particular RteEvent/BswEvent which starts the executable. Rte currently supports the time triggered activation of the following types of event:
  - OperationInvokedEvent
  - DataReceivedEvent
  - DataSendCompletedEvent

- SwcModeSwitchEvent
- BswModeSwitchEvent
- InternalTriggerOccurredEvent
- BswInternalTriggerOccurredEvent
- AsynchronousServerCallReturnsEvent

- The Rte supports the configuration parameter RteMaxNrOfProcessedRequests as specified by RfC AR-107921. This enables projects to specify the maximum number of processed client requests per server executable entity within one task cycle. The RteMaxNrOfProcessedRequests can be configured in the event-to-task-mapping configuration for the particular OperationInvokedEvent resp. BswOperationInvokedEvent which starts the server executable, and it is only applicable if the parameter RtePeriod resp. RteBswPeriod is also configured.
- The Rte supports including additional type header files for the ImplementationDataTypes which provide an `ADMIN-DATA` container as shown below:

```
<ADMIN-DATA>
  <SDGS>
    <SDG GID="EB:PROVIDING-HEADER-FILE">
      <SD GID="PROVIDING-HEADER-FILE">FileName.h</SD>
    </SDG>
  </SDGS>
```

  If the TypeEmitter of an ImplementationDataType is available and set to a value different from RTE, the Rte will evaluate the AdminData of the ImplementationDataType and include the provided header file in the RTE Types Header File.
- The Rte supports value-based data filtering for primitive implementation data types on the sender side. The `DataFilter` algorithms `ALWAYS` and `MASKED-NEW-DIFFERS-MASKED-OLD` are only supported.

  Since the AUTOSAR metamodel doesn't support to describe `DataFilter` on sender side at the moment, the following `AdminData` structure has to be used in the context of the `PPortPrototype`.

```
<P-PORT-PROTOTYPE>
  <SHORT-NAME>pPort1</SHORT-NAME>
  <ADMIN-DATA>
    <SDGS>
      <SDG GID="EB:DATA-FILTER">
        <SD GID="VARIABLE-DATA-PROTOTYPE">/interfaces/if1/de1</SD>
        <SD GID="DATA-FILTER-TYPE">MASKED-NEW-DIFFERS-MASKED-OLD</SD>
        <SD GID="MASK">0b11111111</SD>
      </SDG>
      <SDG GID="EB:DATA-FILTER">
        <SD GID="VARIABLE-DATA-PROTOTYPE">/interfaces/if1/de2</SD>
        <SD GID="DATA-FILTER-TYPE">MASKED-NEW-DIFFERS-MASKED-OLD</SD>
        <SD GID="MASK">0xffff</SD>
      </SDG>
```

```
      </SDGS>
    </ADMIN-DATA>
    <PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/interfaces/if1</PROVIDED-INTERFACE-
  </P-PORT-PROTOTYPE>
```

2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- By pass is not supported

  Description:

  By pass is not supported

  Requirements:

  SWS_Rte_01252, SWS_Rte_03550, SWS_Rte_08500, SWS_Rte_01111, SWS_Rte_01354, SWS_Rte_03553, SWS_Rte_01118, SWS_Rte_02628, SWS_Rte_02725, SWS_Rte_07200, SWS_Rte_07122, SWS_Rte_01083, SWS_Rte_01120, SWS_Rte_03560, SWS_Rte_01003, SWS_Rte_01102, SWS_Rte_02631, SWS_Rte_07367, SWS_Rte_02599, SWS_Rte_03565, SWS_Rte_03741, SWS_Rte_01123, SWS_Rte_07203, SWS_Rte_01206, SWS_Rte_02614, SWS_Rte_05509, SWS_Rte_01126, SWS_Rte_03800, SWS_Rte_03744, SWS_Rte_02619, SWS_Rte_07390, SWS_Rte_03928, SWS_Rte_01091, SWS_Rte_01072, SWS_Rte_07194, SWS_Rte_01150, SWS_Rte_01092, SWS_Rte_07394, SWS_Rte_01071

- Std_TransformerForwardCode type is used in the RTE Send/Write/IWrite APIs instead of Std_TransformerForward

  Description:

  Std_TransformerForwardCode type is used in the RTE Send/Write/IWrite APIs instead of Std_TransformerForward

  Requirements:

  SWS_Rte_06721, SWS_Rte_07207, SWS_Rte_01102, SWS_Rte_02631, SWS_Rte_02599, SWS_Rte_01111, SWS_Rte_01166, SWS_Rte_03741, SWS_Rte_06720, SWS_Rte_07203, SWS_Rte_01206, SWS_Rte_05509, SWS_Rte_03744, SWS_Rte_03608, SWS_Rte_01091, SWS_Rte_07200, SWS_Rte_01072, SWS_Rte_01092, SWS_Rte_07394, SWS_Rte_01071

- Std_TransformerError is not supported

  Description:

  Rte_TransformerError is used as the argument of RTE APIs instead of Std_TransformerError

Requirements:

SWS_Rte_06721, SWS_Rte_07207, SWS_Rte_01102, SWS_Rte_02631, SWS_Rte_02599, SWS_Rte_01111, SWS_Rte_01166, SWS_Rte_03741, SWS_Rte_06720, SWS_Rte_07203, SWS_Rte_01206, SWS_Rte_05509, SWS_Rte_03744, SWS_Rte_03608, SWS_Rte_01091, SWS_Rte_07200, SWS_Rte_01072, SWS_Rte_01092, SWS_Rte_07394, SWS_Rte_01071, SWS_Rte_08791

- Metadata forwarding is not supported

Description:

Metadata forwarding of a data element to/from COM or LdCom to/from the application is not supported

Requirements:

SWS_Rte_06721, SWS_Rte_07207, SWS_Rte_01102, SWS_Rte_02631, SWS_Rte_02599, SWS_Rte_01111, SWS_Rte_01166, SWS_Rte_03741, SWS_Rte_06720, SWS_Rte_07203, SWS_Rte_01206, SWS_Rte_05509, SWS_Rte_03744, SWS_Rte_03608, SWS_Rte_01091, SWS_Rte_07200, SWS_Rte_01072, SWS_Rte_01092, SWS_Rte_07394, SWS_Rte_01071

- returnValueProvision of RTE API is not supported

Description:

This attribute controls the provision of return values for RTE APIs that correspond to the enclosing access point. The attribute is not evaluated.

Requirements:

SWS_Rte_06721, SWS_Rte_07207, SWS_Rte_01102, SWS_Rte_02631, SWS_Rte_02599, SWS_Rte_01111, SWS_Rte_01166, SWS_Rte_03741, SWS_Rte_06720, SWS_Rte_07203, SWS_Rte_01206, SWS_Rte_05509, SWS_Rte_03744, SWS_Rte_03608, SWS_Rte_01091, SWS_Rte_07200, SWS_Rte_01072, SWS_Rte_01092, SWS_Rte_07394, SWS_Rte_01071

- SwcExclusiveAreaPolicy.apiPrinciple is not evaluated

Description:

SwcExclusiveAreaPolicy.apiPrinciple is not evaluated

Requirements:

SWS_Rte_01120, SWS_Rte_01123, SWS_Rte_07253, SWS_Rte_03739, SWS_Rte_07250

- The symbol attribute of is not evaluated.

Description:

The symbol attribute of is not evaluated.

Requirements:

SWS_Rte_06713, SWS_Rte_01126, SWS_Rte_06714, SWS_Rte_07194, SWS_Rte_07163

- Client prefixes for VFB trace hook functions are not supported

  Description:

  Client prefixes for VFB trace hook functions are not supported

  Requirements:

  SWS_Rte_01250, SWS_Rte_01245, SWS_Rte_01246, SWS_Rte_01243, SWS_Rte_01244, SWS_Rte_-01247

- Hook functions for the task are not supported according to R20-11

  Description:

  Hook functions for the task do not contain the os application and task name.

  Requirements:

  SWS_Rte_01250, SWS_Rte_01245, SWS_Rte_01246, SWS_Rte_01243, SWS_Rte_01244, SWS_Rte_-01247

- Trusted function calls are not supported

  Description:

  Trusted function calls are not supported

  Requirements:

  SWS_Rte_07155, SWS_Rte_02527, SWS_Rte_02528

- Attributes HandleOutOfRange and HandleDataStatus is not evaluated

  Description:

  Attributes HandleOutOfRange and HandleDataStatus is not evaluated

  Requirements:

  SWS_Rte_02608, SWS_Rte_02600

- BindingTimeEnum is not evaluated

  Description:

  BindingTimeEnum is not evaluated

Requirements:

SWS_Rte_02613, SWS_Rte_03854, SWS_Rte_02615

- ShortName of the EcucPartition container is not used for determining the partition name.

  Description:

  ShortName of the EcucPartition container is not used for determining the partition name.

  Requirements:

  SWS_Rte_02712, SWS_Rte_02713, SWS_Rte_05088, SWS_Rte_07140

- Rte_Ports API do not return NULL_PTR if the port data structure for the specific port interface does not exist

  Description:

  Rte_Ports API do not return NULL_PTR if the port data structure for the specific port interface does not exist

  Requirements:

  SWS_Rte_03602

- Rte_NPorts API do not return 0 if the port data structure for the port interface does not exist

  Description:

  Rte_NPorts API do not return 0 if the port data structure for the port interface does not exist

  Requirements:

  SWS_Rte_03603

- BswModuleEntity.implementedEntry is not set/evaluated.

  Description:

  BswModuleEntity.implementedEntry is not set/evaluated.

  Requirements:

  SWS_Rte_05106, SWS_Rte_05177, SWS_Rte_05179, SWS_Rte_05180

- Det for some API are not supported

  Description:

  Det for some API are not supported

  Requirements:

SWS_Rte_06632

- ServerArgument of primitve type is set to void when the ServerArgumentPolicy is set to useVoid

  Description:

  If the ServerArgumentImplPolicy is set to useVoid the data type of the &lt

  Requirements:

  SWS_Rte_07027

- No support for self-referencing struct/union

  Description:

  No support for self-referencing struct/union

  Requirements:

  SWS_Rte_07114, SWS_Rte_07144

- PostBuild variant handling not supported

  Description:

  PostBuild variant handling not supported

  Requirements:

  SWS_Rte_07270

- It's not allowed to distribute RunnableEntities of a SWC to tasks or ISRs belonging to different partition

  Description:

  It's not allowed to distribute RunnableEntities of a SWC to tasks or ISRs belonging to different partition

  Requirements:

  SWS_Rte_07347

- SchM_StartTiming API is not supported

  Description:

  SchM_StartTiming API is not supported

  Requirements:

  SWS_Rte_07574

- Rte_NvMNotifyJobFinished is not implemented as per R20-11

  Description:

  Rte_NvMNotifyJobFinished do not accept the NvM_BlockRequestType as argument, instead the serviceId of uint8 type is used.

  Requirements:

  SWS_Rte_07623

- LiternalPrefix attribute defined by the IncludedDataTypeSet of the Referring AutosarDataType is not evaluated

  Description:

  LiternalPrefix attribute defined by the IncludedDataTypeSet of the Referring AutosarDataType is not evaluated

  Requirements:

  SWS_Rte_08403

- Compu method with category RAT_FUNC is not supported

  Description:

  Data conversion for compu method with category RAT_FUNC is not supported

  Requirements:

  SWS_Rte_07928

- Rte_LdComCbkTxConfirmation call back is not implemented according to R20-11

  Description:

  Rte_LdComCbkTxConfirmation function has a new argument of Std_ReturnType which informs RTE if the PDU transmission was successful. According to our current implementation, Rte_LdComCbkTxConfirmation function do not accept any arguments.

  Requirements:

  SWS_Rte_01411

- NvM_WritePRAMBlock API is not supported

  Description:

  NvM_WritePRAMBlock API is not supported

Requirements:

SWS_Rte_08083, SWS_Rte_08082, SWS_Rte_08085, SWS_Rte_08084

- SWS_Rte_08795 is supported for both blocking and non blocking Rte_Receive API

Description:

According to SWS_Rte_08795, RTE shall execute all transfomers of the chain even when the queue is empty if the executeDespiteDataUnavailability of DataTransformation is enabled and the Rte_Receive API has non-blocking characteristics. This behavior is also supported for blocking Rte_Receive API

Requirements:

SWS_Rte_08795

- Rte_Rips_DRead API is not supported

Description:

Rte_Rips_DRead API is not supported

Requirements:

SWS_Rte_80075, SWS_Rte_80065

- Rips for parameter interfaces is not supported

Description:

Rips for parameter interfaces is not supported

Requirements:

SWS_Rte_80079

- Cross cluster communication is not supported

Description:

Cross cluster communication is not supported

Requirements:

SWS_Rte_CONSTR_80002

- Changed functionalities

Description:

The listed requirement have been changed compared to previous AUTOSAR versions. They are currently not support and can be requested on demand.

Requirements:

SWS_Rte_06712

- Partly implemented functionalities

Description:

The listed changed requirements are partly not support and can be requested on demand.

Requirements:

SWS_Rte_06546, SWS_Rte_06547, SWS_Rte_06548, SWS_Rte_02741, SWS_Rte_03831, SWS_Rte_06541, SWS_Rte_02500, SWS_Rte_06542, SWS_Rte_07630, SWS_Rte_06543, SWS_Rte_02740, SWS_Rte_07632, SWS_Rte_02745, SWS_Rte_02746, SWS_Rte_02743, SWS_Rte_03833, SWS_Rte_02744, SWS_Rte_02749, SWS_Rte_03839, SWS_Rte_06549, SWS_Rte_07639, SWS_Rte_03837, SWS_Rte_06550, SWS_Rte_07640, SWS_Rte_07089, SWS_Rte_07083, SWS_Rte_07085, SWS_Rte_06534, SWS_Rte_06535, SWS_Rte_06536, SWS_Rte_06537, SWS_Rte_06530, SWS_Rte_08038, SWS_Rte_06531, SWS_Rte_08037, SWS_Rte_06532, SWS_Rte_06533, SWS_Rte_08039, SWS_Rte_02738, SWS_Rte_02739, SWS_Rte_06539, SWS_Rte_02736, SWS_Rte_02737, SWS_Rte_07190, SWS_Rte_08045, SWS_Rte_06540, SWS_Rte_08041, SWS_Rte_08040, SWS_Rte_07196, SWS_Rte_08042, SWS_Rte_06523, SWS_Rte_06765, SWS_Rte_06524, SWS_Rte_06525, SWS_Rte_06526, SWS_Rte_08027, SWS_Rte_06520, SWS_Rte_07179, SWS_Rte_08026, SWS_Rte_06521, SWS_Rte_08029, SWS_Rte_06522, SWS_Rte_07610, SWS_Rte_08028, SWS_Rte_02727, SWS_Rte_03817, SWS_Rte_06527, SWS_Rte_03816, SWS_Rte_06528, SWS_Rte_07616, SWS_Rte_03936, SWS_Rte_06529, SWS_Rte_07619, SWS_Rte_03935, SWS_Rte_02729, SWS_Rte_07181, SWS_Rte_07180, SWS_Rte_08034, SWS_Rte_07065, SWS_Rte_07186, SWS_Rte_08033, SWS_Rte_07068, SWS_Rte_08036, SWS_Rte_08035, SWS_Rte_02051, SWS_Rte_08030, SWS_Rte_07061, SWS_Rte_07064, SWS_Rte_07185, SWS_Rte_08032, SWS_Rte_08031, SWS_Rte_06512, SWS_Rte_06633, SWS_Rte_03001, SWS_Rte_06513, SWS_Rte_06634, SWS_Rte_06514, SWS_Rte_06515, SWS_Rte_08016, SWS_Rte_06510, SWS_Rte_04577, SWS_Rte_06511, SWS_Rte_08017, SWS_Rte_02710, SWS_Rte_02711, SWS_Rte_06516, SWS_Rte_07606, SWS_Rte_06517, SWS_Rte_06638, SWS_Rte_06518, SWS_Rte_06519, SWS_Rte_07055, SWS_Rte_08022, SWS_Rte_08025, SWS_Rte_07056, SWS_Rte_08024, SWS_Rte_07053, SWS_Rte_07173, SWS_Rte_08020, SWS_Rte_06501, SWS_Rte_06502, SWS_Rte_06503, SWS_Rte_06504, SWS_Rte_07399, SWS_Rte_08004, SWS_Rte_06620, SWS_Rte_06500, SWS_Rte_07038, SWS_Rte_06509, SWS_Rte_06505, SWS_Rte_06507, SWS_Rte_06508, SWS_Rte_07044, SWS_Rte_07043, SWS_Rte_07045, SWS_Rte_07042, SWS_Rte_07041, SWS_Rte_06611, SWS_Rte_06612, SWS_Rte_07029, SWS_Rte_06613, SWS_Rte_03582, SWS_Rte_07026, SWS_Rte_01169, SWS_Rte_06610, SWS_Rte_07148, SWS_Rte_03901, SWS_Rte_05090, SWS_Rte_05091, SWS_Rte_08001, SWS_Rte_07032, SWS_Rte_07035, SWS_Rte_07398, SWS_Rte_05099, SWS_Rte_07034, SWS_Rte_01292, SWS_Rte_05092, SWS_Rte_05093, SWS_Rte_07031, SWS_-

Rte_07030, SWS_Rte_07811, SWS_Rte_06601, SWS_Rte_06602, SWS_Rte_06603, SWS_Rte_06724, SWS_Rte_07812, SWS_Rte_07137, SWS_Rte_06604, SWS_Rte_06725, SWS_Rte_06605, SWS_Rte_06726, SWS_Rte_06606, SWS_Rte_07022, SWS_Rte_07021, SWS_Rte_07020, SWS_Rte_07261, SWS_Rte_05500, SWS_Rte_07800, SWS_Rte_07921, SWS_Rte_07920, SWS_Rte_07802, SWS_Rte_07923, SWS_Rte_02593, SWS_Rte_07801, SWS_Rte_07922, SWS_Rte_05188, SWS_Rte_05189, SWS_Rte_06718, SWS_Rte_07808, SWS_Rte_06719, SWS_Rte_07807, SWS_Rte_07809, SWS_Rte_07804, SWS_Rte_06715, SWS_Rte_07803, SWS_Rte_07924, SWS_Rte_06716, SWS_Rte_07806, SWS_Rte_06717, SWS_Rte_07805, SWS_Rte_05191, SWS_Rte_05192, SWS_Rte_03793, SWS_Rte_06700, SWS_Rte_06701, SWS_Rte_06702, SWS_Rte_CONSTR_09092, SWS_Rte_07116, SWS_Rte_CONSTR_09094, SWS_Rte_06707, SWS_Rte_06708, SWS_Rte_04526, SWS_Rte_06705, SWS_Rte_06706, SWS_Rte_05184, SWS_Rte_05064, SWS_Rte_05185, SWS_Rte_05065, SWS_Rte_05182, SWS_Rte_05183, SWS_Rte_07349, SWS_Rte_01242, SWS_Rte_02573, SWS_Rte_07109, SWS_Rte_02697, SWS_Rte_07344, SWS_Rte_05168, SWS_Rte_05169, SWS_Rte_07104, SWS_Rte_07346, SWS_Rte_01249, SWS_Rte_01248, SWS_Rte_07110, SWS_Rte_05053, SWS_Rte_05174, SWS_Rte_05175, SWS_Rte_05176, SWS_Rte_07111, SWS_Rte_05170, SWS_Rte_05172, SWS_Rte_05159, SWS_Rte_03771, SWS_Rte_08305, SWS_Rte_05155, SWS_Rte_08302, SWS_Rte_05156, SWS_Rte_08301, SWS_Rte_05157, SWS_Rte_07336, SWS_Rte_08304, SWS_Rte_05158, SWS_Rte_01238, SWS_Rte_01239, SWS_Rte_05162, SWS_Rte_07583, SWS_Rte_07582, SWS_Rte_07585, SWS_Rte_05160, SWS_Rte_05161, SWS_Rte_05148, SWS_Rte_05144, SWS_Rte_03764, SWS_Rte_07564, SWS_Rte_07685, SWS_Rte_05146, SWS_Rte_02679, SWS_Rte_03526, SWS_Rte_01346, SWS_Rte_01105, SWS_Rte_01347, SWS_Rte_02557, SWS_Rte_05152, SWS_Rte_07692, SWS_Rte_05153, SWS_Rte_05154, SWS_Rte_07694, SWS_Rte_05150, SWS_Rte_05137, SWS_Rte_01330, SWS_Rte_05138, SWS_Rte_05139, SWS_Rte_08400, SWS_Rte_02662, SWS_Rte_01332, SWS_Rte_02663, SWS_Rte_05136, SWS_Rte_05140, SWS_Rte_07682, SWS_Rte_05141, SWS_Rte_07681, SWS_Rte_05142, SWS_Rte_07684, SWS_Rte_05143, SWS_Rte_07680, SWS_Rte_03860, SWS_Rte_03980, SWS_Rte_07300, SWS_Rte_07663, SWS_Rte_08510, SWS_Rte_03620, SWS_Rte_03861, SWS_Rte_03868, SWS_Rte_02658, SWS_Rte_07670, SWS_Rte_08088, SWS_Rte_07673, SWS_Rte_05115, SWS_Rte_02760, SWS_Rte_07535, SWS_Rte_08742, SWS_Rte_03731, SWS_Rte_02640, SWS_Rte_02761, SWS_Rte_05113, SWS_Rte_07534, SWS_Rte_05114, SWS_Rte_07533, SWS_Rte_03978, SWS_Rte_03977, SWS_Rte_03976, SWS_Rte_03975, SWS_Rte_08507, SWS_Rte_03979, SWS_Rte_08509, SWS_Rte_07662, SWS_Rte_05121, SWS_Rte_07525, SWS_Rte_07645, SWS_Rte_02752, SWS_Rte_03842, SWS_Rte_07642, SWS_Rte_02753, SWS_Rte_03841, SWS_Rte_05101, SWS_Rte_03840, SWS_Rte_05102, SWS_Rte_02756, SWS_Rte_03846, SWS_Rte_02757, SWS_Rte_03845, SWS_Rte_02754, SWS_Rte_02755, SWS_Rte_03843, SWS_Rte_02639, SWS_Rte_03849, SWS_Rte_03848, SWS_Rte_02759, SWS_Rte_03847, SWS_Rte_07091, SWS_Rte_07093, SWS_Rte_07092, SWS_Rte_07099, SWS_Rte_07095, SWS_Rte_07094, SWS_Rte_07097, SWS_Rte_07096

- Unsupported functionalities

  Description:

  The listed requirements are currently not support and can be requested on demand.

  Requirements:

SWS_Rte_08603, SWS_Rte_08605, SWS_Rte_08604, SWS_Rte_08601, SWS_Rte_08600, ECUC_Rte_09129, ECUC_Rte_09136, ECUC_Rte_09137, ECUC_Rte_09134, ECUC_Rte_09135, ECUC_Rte_09132, ECUC_Rte_09133, ECUC_Rte_09130, ECUC_Rte_09131, ECUC_Rte_09149, ECUC_Rte_09147, ECUC_Rte_09148, ECUC_Rte_09145, ECUC_Rte_09146, ECUC_Rte_09143, ECUC_Rte_09144, ECUC_Rte_09142, ECUC_Rte_09116, ECUC_Rte_09117, ECUC_Rte_09114, ECUC_Rte_09115, ECUC_Rte_09112, ECUC_Rte_09113, SWS_Rte_05300, ECUC_Rte_09118, ECUC_Rte_09119, ECUC_Rte_09127, ECUC_Rte_09128, ECUC_Rte_09125, ECUC_Rte_79004, ECUC_Rte_09126, ECUC_Rte_79003, ECUC_Rte_09123, ECUC_Rte_79006, ECUC_Rte_09124, ECUC_Rte_79005, ECUC_Rte_09121, ECUC_Rte_79008, ECUC_Rte_09122, ECUC_Rte_79007, ECUC_Rte_09120, SWS_Rte_08801, SWS_Rte_08800, SWS_Rte_08803, SWS_Rte_08802, SWS_Rte_01164, ECUC_Rte_09105, ECUC_Rte_09106, ECUC_Rte_09103, ECUC_Rte_09104, ECUC_Rte_09101, ECUC_Rte_09102, ECUC_Rte_09100, SWS_Rte_01393, SWS_Rte_01394, SWS_Rte_06840, SWS_Rte_06832, SWS_Rte_06833, SWS_Rte_06834, SWS_Rte_01389, SWS_Rte_06830, SWS_Rte_06839, SWS_Rte_07925, SWS_Rte_06835, SWS_Rte_06836, SWS_Rte_07927, SWS_Rte_06837, SWS_Rte_07926, SWS_Rte_06838, SWS_Rte_01391, SWS_Rte_01392, SWS_Rte_01390, SWS_Rte_CONSTR_09091, SWS_Rte_CONSTR_09090, SWS_Rte_CONSTR_09093, SWS_Rte_CONSTR_09095, SWS_Rte_06031, SWS_Rte_CONSTR_09086, SWS_Rte_CONSTR_09085, SWS_Rte_CONSTR_09088, SWS_Rte_06033, SWS_Rte_CONSTR_09087, SWS_Rte_06034, SWS_Rte_CONSTR_09089, SWS_Rte_08423, SWS_Rte_08424, SWS_Rte_70133, SWS_Rte_70134, SWS_Rte_70135, SWS_Rte_70136, SWS_Rte_70130, SWS_Rte_70131, SWS_Rte_70132, SWS_Rte_70137, SWS_Rte_70138, SWS_Rte_70139, SWS_Rte_08416, SWS_Rte_08417, SWS_Rte_08413, SWS_Rte_70140, SWS_Rte_08419, SWS_Rte_70141, SWS_Rte_70142, SWS_Rte_08405, SWS_Rte_08407, SWS_Rte_08406, SWS_Rte_06204, SWS_Rte_06205, SWS_Rte_06206, SWS_Rte_06200, SWS_Rte_06201, SWS_Rte_06203, SWS_Rte_06210, SWS_Rte_80116, SWS_Rte_80117, SWS_Rte_80114, SWS_Rte_80115, SWS_Rte_80118, SWS_Rte_80119, SWS_Rte_CONSTR_09006, SWS_Rte_CONSTR_09005, SWS_Rte_CONSTR_09008, SWS_Rte_CONSTR_09007, SWS_Rte_CONSTR_09000, SWS_Rte_CONSTR_09002, SWS_Rte_80112, SWS_Rte_CONSTR_09001, SWS_Rte_80113, SWS_Rte_80110, SWS_Rte_80111, SWS_Rte_80127, SWS_Rte_80128, SWS_Rte_80125, SWS_Rte_80126, SWS_Rte_80129, SWS_Rte_91000, SWS_Rte_CONSTR_09017, SWS_Rte_CONSTR_09016, SWS_Rte_CONSTR_09019, SWS_Rte_CONSTR_09018, SWS_Rte_80120, SWS_Rte_CONSTR_09011, SWS_Rte_CONSTR_09010, SWS_Rte_CONSTR_09013, SWS_Rte_80123, SWS_Rte_CONSTR_09012, SWS_Rte_80124, SWS_Rte_CONSTR_09015, SWS_Rte_80121, SWS_Rte_CONSTR_09014, SWS_Rte_80122, SWS_Rte_CONSTR_09028, SWS_Rte_CONSTR_09027, SWS_Rte_CONSTR_09029, SWS_Rte_CONSTR_09020, SWS_Rte_CONSTR_09022, SWS_Rte_CONSTR_09021, SWS_Rte_CONSTR_09024, SWS_Rte_CONSTR_09023, SWS_Rte_CONSTR_09026, SWS_Rte_CONSTR_09025, SWS_Rte_80105, SWS_Rte_80106, SWS_Rte_80103, SWS_Rte_80104, SWS_Rte_80109, SWS_Rte_CONSTR_09040, SWS_Rte_80107, SWS_Rte_80108, SWS_Rte_CONSTR_09039, SWS_Rte_CONSTR_09038, SWS_Rte_CONSTR_09031, SWS_Rte_CONSTR_09030, SWS_Rte_CONSTR_09033, SWS_Rte_CONSTR_09032, SWS_Rte_CONSTR_09035, SWS_Rte_80101, SWS_Rte_CONSTR_09034, SWS_Rte_80102, SWS_Rte_CONSTR_09037, SWS_Rte_CONSTR_09036, SWS_Rte_80100, SWS_Rte_CONSTR_09051, SWS_Rte_CONSTR_09050, SWS_Rte_70111, SWS_Rte_70112, SWS_Rte_70113, SWS_Rte_70114, SWS_Rte_70110, SWS_Rte_

70119, SWS_Rte_CONSTR_09049, SWS_Rte_70115, SWS_Rte_70116, SWS_Rte_70117, SWS_Rte_70118, SWS_Rte_CONSTR_09042, SWS_Rte_80152, SWS_Rte_CONSTR_09041, SWS_Rte_80153, SWS_Rte_80150, SWS_Rte_CONSTR_09043, SWS_Rte_80151, SWS_Rte_CONSTR_09046, SWS_Rte_CONSTR_09045, SWS_Rte_CONSTR_09048, SWS_Rte_80154, SWS_Rte_CONSTR_09047, SWS_Rte_CONSTR_09060, SWS_Rte_08598, SWS_Rte_CONSTR_09061, SWS_Rte_70122, SWS_Rte_70123, SWS_Rte_70124, SWS_Rte_70125, SWS_Rte_70120, SWS_Rte_70121, SWS_Rte_70126, SWS_Rte_70127, SWS_Rte_70128, SWS_Rte_70129, SWS_Rte_CONSTR_09053, SWS_Rte_CONSTR_09052, SWS_Rte_CONSTR_09055, SWS_Rte_CONSTR_09054, SWS_Rte_CONSTR_09057, SWS_Rte_CONSTR_09056, SWS_Rte_CONSTR_09059, SWS_Rte_CONSTR_09058, SWS_Rte_80138, SWS_Rte_80139, SWS_Rte_80136, SWS_Rte_80137, SWS_Rte_CONSTR_09064, SWS_Rte_08111, SWS_Rte_80130, SWS_Rte_CONSTR_09063, SWS_Rte_08594, SWS_Rte_80131, SWS_Rte_CONSTR_09066, SWS_Rte_CONSTR_09065, SWS_Rte_08596, SWS_Rte_80134, SWS_Rte_80135, SWS_Rte_08593, SWS_Rte_80132, SWS_Rte_08592, SWS_Rte_80133, SWS_Rte_80149, SWS_Rte_CONSTR_09080, SWS_Rte_80147, SWS_Rte_80148, SWS_Rte_08577, SWS_Rte_CONSTR_09081, SWS_Rte_08576, SWS_Rte_CONSTR_09084, SWS_Rte_CONSTR_09083, SWS_Rte_70100, SWS_Rte_70101, SWS_Rte_70102, SWS_Rte_70103, SWS_Rte_70108, SWS_Rte_70109, SWS_Rte_70104, SWS_Rte_70105, SWS_Rte_70106, SWS_Rte_70107, SWS_Rte_80141, SWS_Rte_08583, SWS_Rte_CONSTR_09077, SWS_Rte_CONSTR_09076, SWS_Rte_08101, SWS_Rte_80140, SWS_Rte_CONSTR_09079, SWS_Rte_80145, SWS_Rte_CONSTR_09078, SWS_Rte_80146, SWS_Rte_08582, SWS_Rte_01373, SWS_Rte_01374, SWS_Rte_01371, SWS_Rte_01372, SWS_Rte_01375, SWS_Rte_06828, SWS_Rte_06810, SWS_Rte_06811, SWS_Rte_06812, SWS_Rte_06813, SWS_Rte_06814, SWS_Rte_06815, SWS_Rte_06816, SWS_Rte_06800, SWS_Rte_06801, SWS_Rte_06806, SWS_Rte_06807, SWS_Rte_06808, SWS_Rte_06809, SWS_Rte_06802, SWS_Rte_06803, SWS_Rte_06804, SWS_Rte_06805, ECUC_Rte_09198, ECUC_Rte_09199, ECUC_Rte_09196, ECUC_Rte_09197, ECUC_Rte_09194, ECUC_Rte_09195, ECUC_Rte_09192, ECUC_Rte_09193, ECUC_Rte_09190, ECUC_Rte_09191, SWS_Rte_04612, SWS_Rte_04611, SWS_Rte_04610, SWS_Rte_04614, SWS_Rte_04613, SWS_Rte_03992, SWS_Rte_03991, SWS_Rte_03990, SWS_Rte_04601, SWS_Rte_04600, SWS_Rte_03994, SWS_Rte_03993, SWS_Rte_04605, SWS_Rte_03999, SWS_Rte_04604, SWS_Rte_03998, SWS_Rte_04603, SWS_Rte_04602, SWS_Rte_04609, SWS_Rte_04608, SWS_Rte_04607, SWS_Rte_04606, ECUC_Rte_09179, ECUC_Rte_09176, ECUC_Rte_09177, ECUC_Rte_09174, ECUC_Rte_09175, ECUC_Rte_09173, ECUC_Rte_09170, ECUC_Rte_09171, SWS_Rte_03985, SWS_Rte_03988, SWS_Rte_03987, SWS_Rte_03986, ECUC_Rte_09189, ECUC_Rte_09187, ECUC_Rte_09188, ECUC_Rte_09185, ECUC_Rte_09186, ECUC_Rte_09183, ECUC_Rte_09184, ECUC_Rte_09181, ECUC_Rte_09182, ECUC_Rte_09180, ECUC_Rte_09158, ECUC_Rte_09159, ECUC_Rte_09156, ECUC_Rte_09157, ECUC_Rte_09154, ECUC_Rte_09155, ECUC_Rte_09152, ECUC_Rte_09153, ECUC_Rte_09150, ECUC_Rte_09151, ECUC_Rte_89014, ECUC_Rte_89013, ECUC_Rte_89011, ECUC_Rte_89010, ECUC_Rte_09169, ECUC_Rte_09167, ECUC_Rte_09168, ECUC_Rte_09165, ECUC_Rte_89009, ECUC_Rte_09166, ECUC_Rte_89008, ECUC_Rte_09163, ECUC_Rte_89007, ECUC_Rte_09164, ECUC_Rte_89006, ECUC_Rte_09161, ECUC_Rte_89005, ECUC_Rte_09162, ECUC_Rte_89004, ECUC_Rte_89003, ECUC_Rte_09160, ECUC_Rte_89002, ECUC_Rte_89001, ECUC_Rte_89000, SWS_Rte_06787, SWS_Rte_06788, SWS_Rte_06789, SWS_Rte_06783, SWS_Rte_06784, SWS_Rte_06785, SWS_Rte_06786, ECUC_Rte_09008, ECUC_Rte_09009, ECUC_-

Rte_09017, ECUC_Rte_09018, ECUC_Rte_09015, ECUC_Rte_09013, ECUC_Rte_09014, ECUC_Rte_09011, ECUC_Rte_09012, SWS_Rte_06790, SWS_Rte_06791, ECUC_Rte_09010, SWS_Rte_06792, SWS_Rte_06551, SWS_Rte_06793, SWS_Rte_06776, SWS_Rte_04597, SWS_Rte_06777, SWS_Rte_04596, SWS_Rte_06778, SWS_Rte_04595, SWS_Rte_06779, SWS_Rte_04594, SWS_Rte_06772, SWS_Rte_06773, SWS_Rte_06774, SWS_Rte_04599, SWS_Rte_06775, SWS_Rte_04598, ECUC_Rte_09019, ECUC_Rte_09028, ECUC_Rte_09029, ECUC_Rte_09026, ECUC_Rte_09027, ECUC_Rte_09024, ECUC_Rte_09025, ECUC_Rte_09022, ECUC_Rte_09023, ECUC_Rte_09020, SWS_Rte_06780, ECUC_Rte_09021, SWS_Rte_06781, SWS_Rte_06782, SWS_Rte_70092, SWS_Rte_04586, SWS_Rte_70093, SWS_Rte_04585, SWS_Rte_70094, SWS_Rte_04584, SWS_Rte_70095, SWS_Rte_04583, SWS_Rte_07851, SWS_Rte_07850, SWS_Rte_70090, SWS_Rte_04588, SWS_Rte_70091, SWS_Rte_04587, SWS_Rte_70096, SWS_Rte_70097, SWS_Rte_70098, SWS_Rte_70099, SWS_Rte_06771, SWS_Rte_04593, SWS_Rte_04592, SWS_Rte_04591, SWS_Rte_04575, SWS_Rte_07843, SWS_Rte_04574, SWS_Rte_07846, SWS_Rte_04573, SWS_Rte_06757, SWS_Rte_04572, SWS_Rte_07840, SWS_Rte_04579, SWS_Rte_07842, SWS_Rte_07841, SWS_Rte_04576, SWS_Rte_07848, SWS_Rte_07847, SWS_Rte_07849, ECUC_Rte_09006, ECUC_Rte_09007, ECUC_Rte_09004, ECUC_Rte_09005, ECUC_Rte_09002, ECUC_Rte_09003, ECUC_Rte_09000, ECUC_Rte_09001, SWS_Rte_04581, SWS_Rte_04580, SWS_Rte_06743, SWS_Rte_07833, SWS_Rte_04564, SWS_Rte_06744, SWS_Rte_07832, SWS_Rte_04563, SWS_Rte_06745, SWS_Rte_07835, SWS_Rte_04562, SWS_Rte_06746, SWS_Rte_07834, SWS_Rte_04561, SWS_Rte_04568, SWS_Rte_06740, SWS_Rte_04567, SWS_Rte_06741, SWS_Rte_07831, SWS_Rte_04566, SWS_Rte_06742, SWS_Rte_07830, SWS_Rte_04565, SWS_Rte_04569, SWS_Rte_06747, SWS_Rte_07837, SWS_Rte_07836, SWS_Rte_07839, ECUC_Rte_09206, SWS_Rte_07838, SWS_Rte_04571, SWS_Rte_04570, SWS_Rte_06732, SWS_Rte_04553, SWS_Rte_08911, SWS_Rte_04552, SWS_Rte_08910, SWS_Rte_04551, SWS_Rte_04550, SWS_Rte_08912, SWS_Rte_04557, SWS_Rte_04556, SWS_Rte_06730, SWS_Rte_04554, SWS_Rte_04559, SWS_Rte_04558, SWS_Rte_06739, SWS_Rte_04560, SWS_Rte_04542, SWS_Rte_08900, SWS_Rte_04541, SWS_Rte_04540, SWS_Rte_08902, SWS_Rte_08901, SWS_Rte_04546, SWS_Rte_04545, SWS_Rte_04543, SWS_Rte_06729, SWS_Rte_08908, SWS_Rte_04549, SWS_Rte_04548, SWS_Rte_04547, SWS_Rte_08909, SWS_Rte_08904, SWS_Rte_08903, SWS_Rte_06727, SWS_Rte_08906, SWS_Rte_06728, SWS_Rte_08905, SWS_Rte_04531, SWS_Rte_04530, SWS_Rte_04532, SWS_Rte_04538, SWS_Rte_04536, ECUC_Rte_09204, ECUC_Rte_09205, ECUC_Rte_09202, ECUC_Rte_09203, ECUC_Rte_09200, ECUC_Rte_09201, SWS_Rte_82018, SWS_Rte_82017, SWS_Rte_08565, SWS_Rte_08568, SWS_Rte_70034, SWS_Rte_70035, SWS_Rte_70036, SWS_Rte_70037, SWS_Rte_70030, SWS_Rte_70031, SWS_Rte_70032, SWS_Rte_70033, SWS_Rte_70038, SWS_Rte_80071, SWS_Rte_70039, SWS_Rte_80072, SWS_Rte_80070, SWS_Rte_82012, SWS_Rte_82011, SWS_Rte_82010, SWS_Rte_80074, SWS_Rte_82016, SWS_Rte_82015, SWS_Rte_82014, SWS_Rte_80078, SWS_Rte_82013, SWS_Rte_08317, SWS_Rte_08316, SWS_Rte_08319, SWS_Rte_08318, SWS_Rte_70040, SWS_Rte_08313, SWS_Rte_08312, SWS_Rte_08315, SWS_Rte_08314, SWS_Rte_70045, SWS_Rte_70046, SWS_Rte_70047, SWS_Rte_70048, SWS_Rte_70042, SWS_Rte_70043, SWS_Rte_70044, SWS_Rte_70049, SWS_Rte_80082, SWS_Rte_80083, SWS_Rte_80080, SWS_Rte_80081, SWS_Rte_08320, SWS_Rte_08322, SWS_Rte_08321, SWS_Rte_80085, SWS_Rte_08789, SWS_Rte_80059, SWS_Rte_08786, SWS_Rte_08785, SWS_Rte_08788, SWS_Rte_08787, SWS_Rte_70012, SWS_Rte_70013, SWS_Rte_70015, SWS_Rte_08309, SWS_Rte_70010, SWS_Rte_70011, SWS_Rte_70016,

SWS_Rte_70017, SWS_Rte_70018, SWS_Rte_70019, SWS_Rte_08793, SWS_Rte_08311, SWS_Rte_80057, SWS_Rte_80058, SWS_Rte_08790, SWS_Rte_80056, SWS_Rte_82009, SWS_Rte_08778, SWS_Rte_82008, SWS_Rte_82007, SWS_Rte_82006, SWS_Rte_08775, SWS_Rte_06112, SWS_Rte_08774, SWS_Rte_06113, SWS_Rte_06114, SWS_Rte_08776, SWS_Rte_06115, SWS_Rte_70023, SWS_Rte_70024, SWS_Rte_70025, SWS_Rte_70026, SWS_Rte_70020, SWS_Rte_70021, SWS_Rte_70022, SWS_Rte_70027, SWS_Rte_80060, SWS_Rte_70028, SWS_Rte_80061, SWS_Rte_70029, SWS_Rte_08782, SWS_Rte_80064, SWS_Rte_82001, SWS_Rte_06120, SWS_Rte_82000, SWS_Rte_08784, SWS_Rte_80063, SWS_Rte_80068, SWS_Rte_82005, SWS_Rte_80069, SWS_Rte_82004, SWS_Rte_08780, SWS_Rte_82003, SWS_Rte_80067, SWS_Rte_82002, SWS_Rte_08768, SWS_Rte_06105, SWS_Rte_70070, SWS_Rte_08767, SWS_Rte_06106, SWS_Rte_70071, SWS_Rte_06107, SWS_Rte_70072, SWS_Rte_08769, SWS_Rte_06108, SWS_Rte_70073, SWS_Rte_08764, SWS_Rte_06101, SWS_Rte_08763, SWS_Rte_06102, SWS_Rte_08766, SWS_Rte_06103, SWS_Rte_CONSTR_04559, SWS_Rte_06104, SWS_Rte_CONSTR_04558, SWS_Rte_70078, SWS_Rte_72015, SWS_Rte_70079, SWS_Rte_72014, SWS_Rte_72017, SWS_Rte_72016, SWS_Rte_06109, SWS_Rte_70074, SWS_Rte_72011, SWS_Rte_70075, SWS_Rte_72010, SWS_Rte_70076, SWS_Rte_72013, SWS_Rte_70077, SWS_Rte_72012, SWS_Rte_72019, SWS_Rte_CONSTR_80031, SWS_Rte_72018, SWS_Rte_CONSTR_80030, SWS_Rte_08771, SWS_Rte_08770, SWS_Rte_08773, SWS_Rte_06110, SWS_Rte_08772, SWS_Rte_06111, SWS_Rte_07426, SWS_Rte_70081, SWS_Rte_08514, SWS_Rte_07425, SWS_Rte_70082, SWS_Rte_70083, SWS_Rte_72020, SWS_Rte_07427, SWS_Rte_70084, SWS_Rte_08511, SWS_Rte_07422, SWS_Rte_07421, SWS_Rte_08513, SWS_Rte_07424, SWS_Rte_08512, SWS_Rte_07423, SWS_Rte_70080, SWS_Rte_70089, SWS_Rte_70085, SWS_Rte_72022, SWS_Rte_70086, SWS_Rte_72021, SWS_Rte_70087, SWS_Rte_70088, SWS_Rte_72023, SWS_Rte_06100, SWS_Rte_07415, SWS_Rte_07414, SWS_Rte_70050, SWS_Rte_70051, SWS_Rte_70056, SWS_Rte_70057, SWS_Rte_70058, SWS_Rte_70059, SWS_Rte_70052, SWS_Rte_70053, SWS_Rte_70054, SWS_Rte_70055, SWS_Rte_06799, SWS_Rte_70060, SWS_Rte_70061, SWS_Rte_70062, SWS_Rte_06552, SWS_Rte_06794, SWS_Rte_06553, SWS_Rte_06795, SWS_Rte_06796, SWS_Rte_06797, SWS_Rte_70067, SWS_Rte_72004, SWS_Rte_70068, SWS_Rte_72003, SWS_Rte_70069, SWS_Rte_72006, SWS_Rte_72005, SWS_Rte_08739, SWS_Rte_70063, SWS_Rte_72000, SWS_Rte_70064, SWS_Rte_70065, SWS_Rte_72002, SWS_Rte_70066, SWS_Rte_72001, SWS_Rte_72008, SWS_Rte_72007, SWS_Rte_72009, SWS_Rte_CONSTR_04584, SWS_Rte_08049, SWS_Rte_01413, SWS_Rte_08050, SWS_Rte_CONSTR_09121, SWS_Rte_CONSTR_09120, SWS_Rte_CONSTR_09123, SWS_Rte_CONSTR_09122, SWS_Rte_CONSTR_09124, SWS_Rte_91122, SWS_Rte_91121, SWS_Rte_CONSTR_80003, SWS_Rte_CONSTR_80000, SWS_Rte_CONSTR_80009, SWS_Rte_CONSTR_80006, SWS_Rte_CONSTR_80004, SWS_Rte_CONSTR_80005, SWS_Rte_91111, SWS_Rte_91110, SWS_Rte_91113, SWS_Rte_91112, SWS_Rte_CONSTR_80013, SWS_Rte_91115, SWS_Rte_06090, SWS_Rte_CONSTR_80014, SWS_Rte_91114, SWS_Rte_06091, SWS_Rte_CONSTR_80011, SWS_Rte_89023, SWS_Rte_91117, SWS_Rte_06092, SWS_Rte_CONSTR_80012, SWS_Rte_91116, SWS_Rte_89021, SWS_Rte_91119, SWS_Rte_CONSTR_80010, SWS_Rte_89022, SWS_Rte_89020, SWS_Rte_06097, SWS_Rte_06098, SWS_Rte_06099, SWS_Rte_CONSTR_80019, SWS_Rte_06093, SWS_Rte_CONSTR_80017, SWS_Rte_06094, SWS_Rte_06095, SWS_Rte_CONSTR_80015, SWS_Rte_06096, SWS_Rte_CONSTR_80016, SWS_Rte_06079, SWS_Rte_91100, SWS_Rte_91102, SWS_Rte_91101, SWS_Rte_91104, SWS_Rte_CONSTR_80024, SWS_Rte_91103, SWS_Rte_CONSTR_80025,

SWS_Rte_06080, SWS_Rte_91106, SWS_Rte_CONSTR_80022, SWS_Rte_06081, SWS_Rte_91105, SWS_Rte_CONSTR_80023, SWS_Rte_91108, SWS_Rte_CONSTR_80020, SWS_Rte_91107, SWS_Rte_CONSTR_80021, SWS_Rte_91109, SWS_Rte_06086, SWS_Rte_06087, SWS_Rte_06088, SWS_Rte_06089, SWS_Rte_06082, SWS_Rte_CONSTR_80028, SWS_Rte_06083, SWS_Rte_CONSTR_80029, SWS_Rte_06084, SWS_Rte_CONSTR_80026, SWS_Rte_06085, SWS_Rte_CONSTR_80027, SWS_Rte_80039, SWS_Rte_80037, SWS_Rte_80038, SWS_Rte_06068, SWS_Rte_06069, SWS_Rte_06075, SWS_Rte_06076, SWS_Rte_06077, SWS_Rte_80030, SWS_Rte_80035, SWS_Rte_80036, SWS_Rte_06073, SWS_Rte_80033, SWS_Rte_06074, SWS_Rte_80034, SWS_Rte_06057, SWS_Rte_06058, SWS_Rte_06059, SWS_Rte_70001, SWS_Rte_70002, SWS_Rte_70003, SWS_Rte_70004, SWS_Rte_70000, SWS_Rte_70009, SWS_Rte_CONSTR_80200, SWS_Rte_CONSTR_80201, SWS_Rte_70005, SWS_Rte_70006, SWS_Rte_70007, SWS_Rte_70008, SWS_Rte_06064, SWS_Rte_06065, SWS_Rte_CONSTR_03874, SWS_Rte_06066, SWS_Rte_CONSTR_03873, SWS_Rte_CONSTR_80206, SWS_Rte_06067, SWS_Rte_80041, SWS_Rte_06060, SWS_Rte_CONSTR_80204, SWS_Rte_06061, SWS_Rte_CONSTR_03870, SWS_Rte_CONSTR_80205, SWS_Rte_CONSTR_80202, SWS_Rte_80045, SWS_Rte_CONSTR_80203, SWS_Rte_06046, SWS_Rte_06047, SWS_Rte_06048, SWS_Rte_06049, SWS_Rte_06053, SWS_Rte_06054, SWS_Rte_06055, SWS_Rte_06056, SWS_Rte_06050, SWS_Rte_06051, SWS_Rte_06052, SWS_Rte_06039, SWS_Rte_80029, SWS_Rte_06035, SWS_Rte_06036, SWS_Rte_06037, SWS_Rte_06038, SWS_Rte_06042, SWS_Rte_06043, SWS_Rte_06044, SWS_Rte_06045, SWS_Rte_06040, SWS_Rte_06041, ECUC_Rte_09090, SWS_Rte_04528, SWS_Rte_04529, ECUC_Rte_09099, ECUC_Rte_09097, ECUC_Rte_09098, ECUC_Rte_09095, ECUC_Rte_09096, ECUC_Rte_09093, ECUC_Rte_09091, ECUC_Rte_09092, SWS_Rte_02320, SWS_Rte_03891, SWS_Rte_03890, SWS_Rte_02321, SWS_Rte_02322, ECUC_Rte_09079, ECUC_Rte_09077, ECUC_Rte_09078, ECUC_Rte_09075, ECUC_Rte_09076, ECUC_Rte_09073, ECUC_Rte_09074, ECUC_Rte_09071, ECUC_Rte_09072, ECUC_Rte_09070, SWS_Rte_03882, SWS_Rte_03881, SWS_Rte_03880, SWS_Rte_03886, SWS_Rte_03885, SWS_Rte_02310, SWS_Rte_03884, SWS_Rte_02311, SWS_Rte_03883, SWS_Rte_02316, SWS_Rte_02317, SWS_Rte_03889, SWS_Rte_03888, SWS_Rte_03887, SWS_Rte_02318, SWS_Rte_02319, ECUC_Rte_09088, ECUC_Rte_09089, ECUC_Rte_09086, ECUC_Rte_09087, ECUC_Rte_09084, ECUC_Rte_09085, ECUC_Rte_09082, ECUC_Rte_09083, ECUC_Rte_09080, ECUC_Rte_09081, SWS_Rte_03871, SWS_Rte_03875, SWS_Rte_03632, SWS_Rte_03631, SWS_Rte_03630, SWS_Rte_03879, SWS_Rte_03878, SWS_Rte_03877, SWS_Rte_03876, SWS_Rte_02309, SWS_Rte_02307, SWS_Rte_02308, SWS_Rte_89003, SWS_Rte_08091, SWS_Rte_89004, ECUC_Rte_09059, SWS_Rte_89001, SWS_Rte_89002, ECUC_Rte_09057, ECUC_Rte_09058, SWS_Rte_89000, ECUC_Rte_09055, ECUC_Rte_09056, ECUC_Rte_09053, ECUC_Rte_09054, ECUC_Rte_09051, SWS_Rte_89009, ECUC_Rte_09052, SWS_Rte_89007, ECUC_Rte_09050, SWS_Rte_89008, SWS_Rte_89005, SWS_Rte_89006, SWS_Rte_03864, SWS_Rte_03622, SWS_Rte_03863, SWS_Rte_03621, SWS_Rte_03626, SWS_Rte_03867, SWS_Rte_03625, SWS_Rte_03866, SWS_Rte_03624, SWS_Rte_03865, SWS_Rte_03623, SWS_Rte_03629, SWS_Rte_03628, SWS_Rte_03869, SWS_Rte_03627, SWS_Rte_89014, SWS_Rte_89015, SWS_Rte_89012, SWS_Rte_89013, ECUC_Rte_09068, SWS_Rte_89010, ECUC_Rte_09069, SWS_Rte_89011, ECUC_Rte_09066, ECUC_Rte_09067, ECUC_Rte_09064, ECUC_Rte_09065, ECUC_Rte_09062, ECUC_Rte_09063, ECUC_Rte_09060, SWS_Rte_89018, ECUC_Rte_09061, SWS_Rte_89019, SWS_Rte_89016, SWS_Rte_89017, SWS_Rte_03610, SWS_Rte_02523, SWS_Rte_03619, SWS_Rte_08070, ECUC_-

Rte_09039, ECUC_Rte_09037, ECUC_Rte_09038, ECUC_Rte_09035, ECUC_Rte_09036, ECUC_Rte_09033, ECUC_Rte_09034, ECUC_Rte_09031, SWS_Rte_08077, ECUC_Rte_09032, SWS_Rte_08079, ECUC_Rte_09030, SWS_Rte_CONSTR_09101, SWS_Rte_08073, SWS_Rte_CONSTR_09100, SWS_Rte_08076, SWS_Rte_CONSTR_09102, SWS_Rte_08061, ECUC_Rte_09048, ECUC_Rte_09049, SWS_Rte_03609, ECUC_Rte_09046, ECUC_Rte_09047, ECUC_Rte_09044, ECUC_Rte_09045, SWS_Rte_08067, ECUC_Rte_09042, SWS_Rte_08066, ECUC_Rte_09043, SWS_Rte_08069, ECUC_Rte_09040, SWS_Rte_08068, ECUC_Rte_09041, SWS_Rte_08065

2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- The value RTE_E_COM_STOPPED is not returned for inter-partition/inter-ECU communication

  Description:

  If the Com module is not able to successfully send a ComSignal or ComSignalGroup and the calling software component is located in another partition as the Com module, an Rte Api function will not return the value RTE_E_COM_STOPPED to the calling software component.

  Rationale:

  When the Com module is located in a different partition as the SWC, the ComSignal must be sent asynchronously via the configured BswTask. The Rte can therefore not access the result of the Com function execution before returning control to the software component.
- The Rte expects an IOC configuration and IOC API which is not fully compliant to AUTOSAR 4.0

  Description:

  The Rte expects an EB-specific IOC configuration. Moreover, the Rte expects the IOC to provide an `Ioc_ReInit` function. Also the Rte expects that the IOC API is reentrant.

  Rationale:

  The AUTOSAR 4.0 specification of the IOC still has some gaps. The EB-specific solution allows more optimizations.
- When converting an AUTOSAR 3.1 system description to AUTOSAR 4.0, not all initial value types for parameters are supported

  Description:

When converting an AUTOSAR 3.1 system description to AUTOSAR 4.0, initial values for parameters, that represent a nested record or array type, are not supported. Only primitive types, simple record, and array types are supported.

Rationale:

The converter cannot perform a deep copy on the value specification of the parameter's initial value. Thus, it is not supported that the initial value is a nested record or array, etc.

- When converting an AUTOSAR 3.1 system description to AUTOSAR 4.0, an opaque type will lose the byte size information

  Description:

  When converting an AUTOSAR 3.1 system description to AUTOSAR 4.0, an opaque type from AUTOSAR 3.1 will lose its byte size information.

  Rationale:

  The opaque type is converted to a primitive type. Thus the byte size information gets lost.

- If inter-partition mode management is used, it cannot be guaranteed that the initialization of the inter-partition mode users' partition is complete once its initialization routine returns

  Description:

  Because partitions execute independently, it may be possible that the initialization routine of the mode users' partition returns before that of the mode provider's partition. In this situation, the `OnEntry` runnables of the inter-partition mode users will not have been executed after the initialization of the partition has been completed. Also, the mode disabling dependencies of the initial mode would not be enabled.

  Reliance on inter-partition modes for initialization should be avoided. If inter- partition modes are to be used for initialization, a check of the mode provider's partition state should be made before accessing any resources which require initialization.

  Rationale:

  Partitions execute, terminate, and start independently.

- Rte Enhanced Mode Api for the mode manager in case of inter-partition mode management

  Description:

  Rte Mode Api for the enhanced mode provider inter-partition mode management, will return the current mode for the previous and next mode in case all mode users have the same current mode.

- The data filter types `NEWISEQUAL`, `NEWISDIFFERENT`, `MASKEDNEWEQUALSMASKEDOLD`, `NEWISGREATER`, `NEWISLESSOREQUAL`, `NEWISLESS`, and `NEWISGREATEROREQUAL` for a require port typed by sender receiver interfaces are not supported

Description:

The EB tresos Studio supports AUTOSAR 4.0.3 where the mentioned data filter types are not supported any longer.

Rationale:

In AUTOSAR 4.0.3, the mentioned data filter types have been removed.

- Partial record support (signal degradation) is not available for inter-partition communication

Description:

Sender/receiver communication with the IOC or Shared memory communicator (Smc) will not function if the destination data element is a partial record.

- Segmentation fault checks are not supported

Description:

The segmentation fault check performed by the Rte in case of an inter-partition communication is not supported.

- Minimal offset of allocated Os schedule table expiry points is one microsecond

Description:

When allocating an Os schedule table, the offset of an expiry point must be divisible by one microsecond.

Rationale:

The expiry point names consist of the prefix `EP_` and the offset value in microseconds.

- Implementation data types of category function reference are not supported

Description:

Implementation data types or implementation data type elements of category `FUNCTION_REFERENCE` are not supported.

- Limited calibration support on S12X with Metroworks compiler

Description:

When a mixed memory model is used on the S12X, the following applies:

- `RTE_VAR` is mapped to far and `RTE_CONST` is mapped to near.
- `RTE_VAR` is mapped to near and `RTE_CONST` is mapped to far.

The following limitations apply when calibration is used:

1. `Rte_CData` and `Rte_CalPrm` API do not work with record and array types.

2. Online calibration using single pointered or double pointered methods is not possible.

Rationale:

1. For complex types, the APIs return pointers. This requires nested compiler abstraction. The Metroworks compiler cannot handle this correctly. Although nested compiler abstraction is forbidden by the AUTOSAR Compiler Abstraction specification (COMPILER058), nested compiler abstraction must be used because the RTE SWS specifies that the APIs shall return pointers. Here, the RTE SWS has to be changed. See Bugzilla #42952.
2. The pointers in the calibration reference table point to constants in ROM. For double pointered method, the calibration base pointer points to a calibration reference table in ROM. When `RTE_CONST` is mapped to near, the pointers cannot be redirected to variables located in RAM which would have to be addressed far in this case.

- The activation offset of a timing event cannot be greater or equal to the period

Description:

The Rte does not support timing events with an activation offset greater or equal to the period. If you configure an activation offset greater or equal to the period anyway, then the offset is automatically shortened to be within the period by a modulo division (e.g. period is 100ms, original activation offset is 150ms and the resulting offset will be 50ms).

- If pointer to array type is used, linker type checking cannot be enabled

Description:

The function signature of API functions that are implemented by the Rte always uses pointer to array base type for array arguments. If pointer to array type is used, the array type in the function declaration of the software component header does not match with the array type of the Rte implementation. On object code level, both variants are equivalent. However a linker where type checking is enabled, may report an error.

- Limited support for NvBlockSwComponentType data consistency

Description:

The NvBlockSwComponentType currently only supports a simple interrupt blocking mechanism to assure data consitency. Ohter mechanisms, e.g. "Usage of OS resources", are not supported.

- Calibration access to NvBlockDescriptor romBlock is not supported

Description:

The NvBlockSwComponentType does not support calibration access for the NvBlockDescriptor romBlock.

- Limited support for NvBlockSwComponentType NvBlockDescriptor parameters

Description:

The NvBlockSwComponentType does not support the NvBlockDescriptor parameters, "constantValueMapping", "dataTypeMapping" and "instantiationDataDefProps".

- Limited support for PortInterfaceMapping

  Description:

  NvDataInterfaces currently support only Bitfield Textable PortInterfaceMapping.

- Limited support for SubElementMapping of VariableAndParameterInterfaceMapping without transformers

  Description:

  Currently the following limitations apply when using SubElementMappings without transformers:

  Only mapping a composite element to primitive DataPrototype is supported

  Only the mapping of struct elements is supported (no array elements)

  Data conversion is not supported

  Data invalidation is not supported

  Mapping of variable data prototypes to parameter data prototypes is not supported

- Limited support for SubElementMapping of VariableAndParameterInterfaceMapping with transformers

  Description:

  Currently the following limitations apply when using SubElementMappings with transformers:

  If optional structure elements are used, only SubElementMappings from one ImplementationDataType to another ImplementationDataType are supported

  Mapping of variable data prototypes to parameter data prototypes is not supported

- Limited support for inter-ECU trigger communication

  Description:

  Currently the following limitations apply when using inter-ECU trigger communication:

  Only supported with data transformation

  Inter-partition communication is not supported (the mapped signal must be on the same partition as the SWC)

- The blocking Rte_Receive API is not supported for inter-partition communication

  Description:

The Rte generator logs an error if all of the following conditions apply:

1. Sender and receiver are located in different partitions.
2. A blocking `Rte_Receive` API is configured.

- The Com_InvalidateSignalGroup API is not supported

  Description:

  The Rte does not use the `Com_InvalidateSignalGroup` API to invalidate a signal group. Instead, it always sends the signal group with the invalid value.

- The Com_InvalidateSignal API is not used in case of inter-partition inter-ECU signal invalidation

  Description:

  Regardless of the value of parameter `InterECUInvalidationHandledByRte` the Rte always sends the invalid value in case of inter-partition inter-ECU sender/receiver communication instead of using the `Com_InvalidateSignal` API.

- The operation invoked event of the server which is called by the `Rte_NvMNotifyJobFinished` callback shall not be mapped to a task

  Description:

  The `Rte_NvMNotifyJobFinished` callback supports only direct calls. Please note that the behavior of the Rte is undefined if this runnable calls any Rte API, because it's running in an interrupt context (see also SWS_Rte_03600).

- Mode disabling dependencies for direct call dirtyFlag runnables of a NvBlockSwComponent are not supported.

  Description:

  Mode disabling dependencies for dirtyFlag runnables of a NvBlockSwComponent are only supported if the DataReceivedEvent is mapped to a task.

- Data consistency is not guaranteed for sender-receiver invalidation external replacement

  Description:

  If a sender-receiver data element has been received invalidated and the attribute handleInvalid is set to externalReplacement the Rte doesn't guarantee data consistency for the external replacement value.

- Sender-receiver invalidation external replacement with ParameterDataPrototype not fully supported

  Description:

Data element invalidation with the attribute handleInvalid is set to externalReplacement and replaceWith references a ParameterDataPrototype is not fully supported in case of inter-partition sender-receiver communication.

- Debounced activation of executables is not supported for certain events

  Description:

  The minimumStartInterval is ignored for executables started by ModeSwitchEvents, ModeSwitchedAckEvents, OperationInvokedEvents or only TimingEvents.

- Event order cannot be guaranteed for debounced activation of executables

  Description:

  The Rte cannot guarantee that a delayed event (i.e. the first event occurrence within the minimumStartInterval period) will finally lead to the start of the executable if another event for the same executable occurred in close succession to the expiration of the debounced alarm of the delayed event.

  Rationale:

  It depends on the internal scheduling of the Os which task of the executable will enter running state first when multiple events are set at once.

- Distribution of the Com onto multiple partitions

  Description:

  The following parameters cannot be configured separately for each Com partition:

  - SendSignalQueueStrategy
  - BswOsTaskPeriod

  For client/server communication, the call and result signal must be mapped onto the same partition.

  For client/server communication, an operation fan-in/out (multiple call/result signals mapped to the same operation) is not supported.

- Transport protocol for serialization

  Description:

  The non-AUTOSAR API LdCom_TpTransmit is used by the Rte.

- XfrmBufferLengthType UINT32 not supported for inter-partition communication via IOC

  Description:

  For inter-partition communication, IOC does not support an XfrmBufferLengthType value other than UINT16.

Rationale:

IOC does not provide an API for UINT32 variable length data.

- The following Signal fan-in/out scenario is not supported for inter-ECU sender/receiver communication

  Description:

  ISignal(1) to (Ld)Com (n)

- The following Signal fan-in/out scenarios are not supported for inter-ECU client/server communication

  Description:

  ISignal(1) to (Ld)Com (n)

  SystemSignal(1) to ISignal (n)

  Operation(n) to SystemSignal (1)

  Server port(n) to SystemSignal (1)

- Asynchronous client/server communication with two clients mapped to Os Interrupt and OsTask is not supported

  Description:

  Asynchronous client/server communication with two clients where one client is mapped to an Os Interrupt and the other to an OsTask is not supported.

- Limited support for writing data to different NvRamBlocks

  Description:

  Writing data to different NvRamBlocks using the same RPortPrototype of a NvBlockSwComponentType is not supported.

- Limited support for Coherency Groups

  Description:

  Currently the following limitations apply when using Coherency Groups:

  DirectReadFromCom is not supported

  NvBlockSwComponentTypes are not supported

  Grouping of accesses across execution conditions (e.g. different data received events or mode disabling dependencies) is not supported

For read/write accesses not in a Coherency Group, which share the same receive buffer with accesses in a Coherency Group, the same locking mechanisms are applied

Grouping of the status members for senders with transmition acknowledgement enabled is not supported

- Limited support for mapping events to ISRs

  Description:

  Currently the following limitations apply when mapping events to ISRs:

  Only (BSW)TimingEvent and (BSW)ExternalTriggerOccurredEvent are supported.

  It is allowed to map either timing or external trigger occurred events to an ISR but not the both event types.

- Limited support for mode switched acknowledgment event

  Description:

  Currently the following limitations apply when using mode switched acknowledgment event:

  one mode switch may lead to more than one mode switched acknowledgment event in case of mode communication with multiple mode user partitions on several cores.

- Limited support for Det error reporting when multiple partitions, and Bsw Modules multiple instances with shared exclusive area are used

  Description:

  Currently the following limitations apply when multiple partitions are used. In this case the Det error reporting is not supported.

- Limited support for rule based value specification

  Description:

  Currently the following limitations apply when using RuleBasedValueSpecifications:

  Compound Primitive Data Types are not supported.

  RULE-ARGUMENTS of type VTF are not supported.

- Limited support for variable size array inter-ECU communication without transformers

  Description:

  Currently the following limitations apply when variable size array inter-ECU communication without transformers is used:

  Data invalidation is not supported.

Multiple variable size arrays with different data types can not be received by the same TP LdCom PDU.

- Limited support for RIPS with parameter interface

  Description:

  Currently the following limitations apply when using RIPS with parameter interfaces:

  Rte Online Calibration must be deactivated (RteCalibrationSupport = NONE)

  RteRipsGlobalCopyInstantiationPolicy must be set to RTE_RIPS_INSTANTIATION_BY_PLUGIN

  RTE_PTR2ARRAYTYPE_PASSING is not supported for `Rte_Prm` API

  Only 1:n communication is supported

- Limited support for synchronous mode switching procedure over multi partitions/cores

  Description:

  Currently the following limitations apply when using synchronous mode switching procedure over multi partitions/cores:

  Partition restarts are not considered. That means if a partition is not yet started or stopped then a mode switch during this time could lead to infinite waiting in transition of the active partitions. So we assume that all mode user partitions are either active or about to start so that the application will not get stuck in a mode transition.

- Limited support for mapping of variable data prototypes to parameter data prototypes

  Description:

  Currently the following limitations apply when mapping of variable data prototypes to parameter data prototypes:

  PR port is not supported

  Inter ECU is not supported

  Only 1:n communication is supported

  Transformers, SubElementMapping and DataConversion are not supported

  Timeout monitoring, handleNeverReceived, enable update, handleOutOfRange, and invalidation are not applicable for this case

  DataReceivedEvents/DataReceiveErrorEvents are not supported

  RIPS is not supported

- Limited support for the initialization of the Rte implicit write buffers according to SWS_Rte_08418

  Description:

  Currently the following limitations apply when the Rte implicit write buffers are initialized according to SWC_Rte_08418:

  For a pure Inter-ECU communication, the Rte implicit write buffers might be unitialized before the first invocation of the runnable with Rte_IWrite API.
- Limited support for transformer code forwarding according to SWS_Rte_04572

  Description:

  Currently the following limitations apply when transformer code forwarding is used:

  Only the Rte_Send and Rte_Write APIs are supported.

2.6. Open-source software

The software that is delivered with EB tresos AutoCore Generic can be classified into the following two categories:

- Software that is executed on the electronic control unit (ECU).
- Software that is used for the development infrastructure (configuration, generation, building) and thus executed on the development platform.

2.6.1. Open-source software in software executed on the ECU

No open-source software that runs on the ECU is delivered with Rte.

2.6.2. Open-source software in software used for the development infrastructure

The following list of open-source software that is used in development is delivered with Rte:

- Commons Math: The Apache Commons Mathematics Library
  3.2.
  https://commons.apache.org/proper/commons-math/

List of licenses:

- Apache License Version 2.0
  commons-math.txt

List of copyrights:

- Copyright (C) The Apache Software Foundation.