

License Number

CBD1601056

Customer

Nexteer Automotive Corporation
Package: MSR_Ford_SLP1 - ECU product, "Steering Systems"

Maintenance Expiry Date

2018-03-01

SIP Version

18.00.15

SLP

MSR_Ford_SLP1

Delivery Number

D05

Report Creation Date

2018-07-31

Contact

In case of questions or the need for an update of the basic software delivery, please contact Support@vector.com or your Vector contact person.

Table of Contents

1. [Introduction](#)
 - [1.1 Resolving Issues](#)
 - [1.2 Issue Classification](#)
2. [New Issues](#)
 - [2.1 Safety Relevant Issues: 29](#)
 - [2.2 Runtime Issues without Workaround: 41](#)
 - [2.3 Runtime Issues with Workaround: 52](#)
 - [2.4 Not Released Functionality: 11](#)
 - [2.5 Apparent Issues: 211](#)
 - [2.6 Compiler Warnings: 54](#)
3. [New Issues for Information: 0](#)
4. [Report Legend](#)
5. [3rd Party Software Issues](#)
6. [Quality Management Contact](#)

1. Introduction

1.1 Resolving Issues

Reported issues are not automatically fixed with the next update delivery.

If a reported issue shall be fixed, please contact Vector agree on the issues that can be fixed with upcoming deliveries.

Please note that Vector may fix issues without explicit request.

1.2 Issue Classification

This Issue Report provides issues that have been detected since the last report. The issues have been classified to facilitate the assessment of their impact:

The chapter 'New Issues' lists issues that have been detected since the last report and which could not be excluded based on the use-case defined in the questionnaire. The issues are classified as follows:

- **Safety Related Issues:** Safety related issues have impact on the functional safety of the software module. If this issue interferes with the functional safety concept of the ECU, this module (or module configuration) must not be used for serial production in a safety-related project. The effect of the issue to the ECU functionality and functional safety has to be analyzed by the user as the software usage and its configuration is not known by Vector. The risk of change has also to be taken into account.
- **Runtime Issues without Workaround:** Runtime issues without a workaround require an update of the software delivery in case the issue affects the ECU overall functionality. The effect of an issue to the ECU functionality has to be analyzed by the customer as the software usage and its configuration is not known by Vector. The risk of change has also to be taken into account.
- **Runtime Issues with Workaround:** It is not recommended to update a delivery due to a runtime issue with a documented workaround. The effect of an issue to the ECU functionality has to be analyzed by the user as the software usage and its configuration is not known by Vector. The risk of change has also to be taken into account.
- **Not Released Functionality:** Not released functionalities (BETA) are either complete software modules or features in the software module that have not yet passed a complete development cycle (they are e.g. not or only partly tested). If a BETA issue ticket affects a complete software module, the software module must not be used for serial production. If a BETA issue ticket affects a feature in the software module, the user has to ensure that all BETA features are disabled as indicated for the serial production release of the ECU.
- **Apparent Issues:** Apparent issues are detected immediately when using the software module. If an issue does not show up while working with the software module, the ECU project is not affected by the issue. Apparent issues may or may not have workarounds.
- **Compiler Warnings:** As a service we also provide the known compiler warnings. The occurrence of a compiler warning may depend on the used software module configuration and compiler settings.

The chapter 'New Issues for Information' lists issues that are not relevant for the use-case that has been documented in the questionnaire provided to Vector. The issues may, however, be relevant for other use-cases. Additionally, issues that have been accepted or are tolerated by the OEM (as defined in the questionnaire) are reported here.

2. New Issues

2.1 Safety Relevant Issues

Safety related issues have impact on the functional safety of the software module. If this issue interferes with the functional safety concept of the ECU, this module (or module configuration) must not be used for serial production in a safety-related project.

The effect of the issue to the ECU functionality and functional safety has to be analyzed by the user as the software usage and its configuration is not known by Vector. The risk of change has also to be taken into account.

Index

ESCAN00095054	Compile error for arrayBased SignalGroups and UINT8_N ApplType Il_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00095541	Initialization of Rte_CS_WaitingTaskList triggers protection hook Rte_Core@Generator
ESCAN00095734	Path to the analyzed files is incomplete in the generation report Rte_Analyzer@Application
ESCAN00096055	Rte_IsUpdated reports wrong status Rte_Core@Implementation
ESCAN00096151	IR Analyzer misses data consistency problem GenTool_IRAnalyzer@Application
ESCAN00096202	CANTRCV_30_TJA1043_CONFIGURATION_VARIANT != CANTRCV_30_TJA1043_CONFIGURATION_VARIANT_POSTBUILD_LOADABLE is not ensured by ElisaPlugin DrvTrans_Tja1043CandioAsr@ElisaPlugin
ESCAN00096387	Undefined ECU behavior due to invalid index access if <MSN>WriteOutOfBoundsWriteProtectionStrategy is INDEX_SATURATION and Post Build Variance Support is true CommonAsr_ComStackLib@GenTool_GeneratorMsr
ESCAN00096411	Undefined ECU behavior due to invalid value access in post build selectable configuration with more than 2 variants CommonAsr_ComStackLib@GenTool_GeneratorMsr
ESCAN00096425	IR Analyzer misses data consistency problem GenTool_IRAnalyzer@Application
ESCAN00096797	EcuM_Shutdown throws a DET in some multicore projects in context of EcuM_Shutdown() and shutdown / reset is not performed SysService_Asr4EcuM@Doc_TechRef
ESCAN00096892	Undefined ECU behavior due to invalid index access if PduWriteOutOfBoundsWriteProtectionStrategy is INDEX_SATURATION and Post Build Variance Support is true Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00097193	Consider AUTOSAR package for ModeDeclarationGroups to allow ModeDeclarationGroups with same name in different packages Rte_Core@Implementation
ESCAN00097384	Service 0x22: Overwritten RAM behind a DCM buffer Diag_Asr4Dcm@Implementation
ESCAN00097518	CRC32 calculations deliver wrong results MemService_AsrNvM@Implementation
ESCAN00097644	RTE dereferences NULL_PTR after execution of a mapped server runnable Rte_Core@Implementation

Index

ESCAN00097801	Undefined ECU behavior due to invalid index access if <MSN>WriteOutOfBoundsWriteProtectionStrategy is INDEX_SATURATION and Post Build Variance Support is true IL_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00097829	Service 0x22: Overwritten call stack Diag_Asr4Dcm@Implementation
ESCAN00097901	Rx Deferred Event Cache leads to unexpected ECU behaviour under high load IL_AsrComCfg5@Implementation
ESCAN00098374	Memory corruption during module initialization Diag_Asr4Dem@Implementation
ESCAN00098443	SchM_Init starts triggering runnables before Rte_Start when schedule tables are configured Rte_Core@Implementation
ESCAN00098513	PanicHook is called or system enters endless loop on stack overflow Os_CoreGen7@Implementation
ESCAN00098598	Rte_Call returns invalid data Rte_Core@Implementation
ESCAN00098617	Shutdown / Reset of a multicore ECU is not performed as expected SysService_Asr4EcuM@Implementation
ESCAN00099129	Implicit write or read accesses do not work Rte_Core@Implementation
ESCAN00099150	External Rte_Read does not return initial value Rte_Core@Implementation
ESCAN00099276	Memory corruption in CompareKey SysService_AsrCryFord@Implementation
ESCAN00099646	Schedulable entity not triggered when entity is mapped to a basic task and cyclic trigger implementation is set to none Rte_Core@Implementation
ESCAN00099865	Error in Silent Analysis Nm_Asr4NmCan@Implementation
ESCAN00099885	Runnable with mode disabling not triggered when no mode switch point is configured Rte_Core@Implementation

ESCAN00095054 Compile error for arrayBased SignalGroups and UINT8_N ApplType

Component@Subcomponent: Il_AsrComCfg5@GenTool_GeneratorMsr

First affected version: 12.00.00

Fixed in versions: 13.01.00

Problem Description:

What happens (symptoms):

Compiler complains about an undefined Macro and will throw the following (Ansi Compiler) or similar messages:

1>..\..\..\external\BSW\Com\Com.c(5820): error C2059: syntax error : ')'

1>..\..\..\external\BSW\Com\Com.c(9377): error C2059: syntax error : ')'

1>..\..\..\external\BSW\Com\Com.c(9489): error C2061: syntax error : identifier

'Com_RxSignalProcessing'

1>..\..\..\external\BSW\Com\Com.c(9489): error C2059: syntax error : ';'.

1>..\..\..\external\BSW\Com\Com.c(9489): error C2059: syntax error : 'type'

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

/MICROSAR/Com/ComConfig/ComSignalGroup/ComSignalGroupArrayAccess is set to TRUE and UINT8_N applType is only defined for those groupSignals which are contained in an array based signalGroup.

Resolution Description:

Workaround:

Set /MICROSAR/Com/ComConfig/ComSignalGroup/ComSignalGroupArrayAccess to FALSE (only possible when no COM Based Transformer is used)

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095541 Initialization of Rte_CS_WaitingTaskList triggers protection hook	
Component@Subcomponent:	Rte_Core@Generator
First affected version:	1.13.00
Fixed in versions:	1.16.00
Problem Description:	
What happens (symptoms):	

The system calls the protection hook.	
When does this happen:	

During runtime when the system is initialized and Rte_Start is called.	
In which configuration does this happen:	

This happens when the configuration contains client-server calls to mapped server runnables and when one client operation can be called from runnables from different tasks.	
This only happens when the server runnable is mapped to a task from a different partition than the task that contains the BSW.	
Resolution Description:	
Workaround:	

Create separate server ports for every task that contains a client that calls the server runnable.	
Create separate client ports for every task that contains a client that calls the server runnable.	
Trigger the same server runnable by all server ports.	
Map all server triggers to the same task.	
Please note that every server port will use a separate queue, then.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095734 Path to the analyzed files is incomplete in the generation report	
Component@Subcomponent:	Rte_Analyzer@Application
First affected version:	0.07.00
Fixed in versions:	1.00.01
Problem Description: What happens (symptoms): ----- The path to the analyzed files in the analysis report is not complete The subfolder Source is missing for the stubs ".." is missing for the RTE sources This is only a display problem in the report. The files are still analyzed. When does this happen: ----- Always and immediately In which configuration does this happen: ----- All	
Resolution Description: Workaround: ----- During the review, it should be considered that the paths miss '..' or 'source' in analysis report. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096055 Rte_IsUpdated reports wrong status

Component@Subcomponent: Rte_Core@Implementation

First affected version: 1.08.00

Fixed in versions: 1.16.00

Problem Description:

What happens (symptoms):

Rte_IsUpdated reports that a data element was updated although it was not updated or it reports that a data element was not updated although it was updated.

When does this happen:

During runtime.

When a Rte_Read API that receives a signal group with enabled update status is interrupted by a callback or another task that also accesses the update flags.

In which configuration does this happen:

This happens when the all of the following conditions are true:

- data element is mapped to a signal group
- Rte_IsUpdated is enabled for the data element
- ComIPduSignalProcessing of the mapped signal group is set to DEFERRED
- the task that contains Com_MainFunctionRx cannot interrupt the task that executes the Rte_Read

When the RTE is optimized for RUNTIME, only the update status of the flag of this data element can be wrong.

When the RTE is optimized for MEMORY also the update status of other data elements with update flags can be wrong.

The problematic APIs can be found by searching for Rte_Read APIs with accesses to Rte_*RxUpdateFlags without interrupt locks.

Resolution Description:

Workaround:

Either

configure another port access in another runnable that runs in a task that can interrupt the first access.

Or

call Rte_Read APIs with accesses to Rte_*RxUpdateFlags that are not protected with interrupt locks with locked interrupts.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096151 IR Analyzer misses data consistency problem	
Component@Subcomponent:	GenTool_IRAnalyzer@Application
First affected version:	0.09.00
Fixed in versions:	1.00.01
Problem Description:	
What happens (symptoms):	

RTE Analyzer does not report a data consistency problem.	
When does this happen:	

During RTE analyzer execution.	
In which configuration does this happen:	

This happens when the analyzed source code contains multiple accesses to the same global variable in a task and	
when one access is protected with an interrupt lock and the other accesses are not protected	
although an interrupt lock would have been required.	
This error only leads to a problem in the ECU, when the RTE generator generates code that	
contains this problem.	
At the creation date of this ESCAN, no RTE generator problem that leads to the generation of such	
code is known.	
Resolution Description:	
Workaround:	

Manually check whether whether accesses to global variables that are not surrounded by interrupt	
locks can be interrupted	
by accesses in other tasks or callbacks.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096202	CANTRCV_30_TJA1043_CONFIGURATION_VARIANT != CANTRCV_30_TJA1043_CONFIGURATION_VARIANT_POSTBUILD_LOADABLE is not ensured by ElisaPlugin
Component@Subcomponent:	DrvTrans_Tja1043CandioAsr@ElisaPlugin
First affected version:	1.00.00
Fixed in versions:	1.00.01
Problem Description:	
What happens (symptoms):	

The condition:	
- CANTRCV_30_TJA1043_CONFIGURATION_VARIANT != CANTRCV_30_TJA1043_CONFIGURATION_VARIANT_POSTBUILD_LOADABLE in order to fulfill the COV_CANTRCV_HL_LL_TJA1043_VARCOV_SW_FEAT_NOT_SUPPORTED fully is not ensured by ElisaPlugin.	
When does this happen:	

(1) Always and immediately: In case of ASIL-D version of driver is applied	
In which configuration does this happen:	

- Independent of configuratopn: In case of ASIL-D version of driver is applied	
Resolution Description:	
Workaround:	

Please ensure by your own that the value of CANTRCV_30_TJA1043_CONFIGURATION_VARIANT != CANTRCV_30_TJA1043_CONFIGURATION_VARIANT_POSTBUILD_LOADABLE [-> See file CanTrcv_30_Tja1043_Cfg.h]	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096387		Undefined ECU behavior due to invalid index access if <MSN>WriteOutOfBoundsWriteProtectionStrategy is INDEX_SATURATION and Post Build Variance Support is true	
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr		
First affected version:	7.00.00		
Fixed in versions:	8.03.03, 8.07.03, 8.03.80, 8.06.01, 8.07.81, 8.05.02, 8.07.80, 8.01.01, 8.11.00, 8.00.01, 8.04.01, 8.03.81, 8.05.80		
Problem Description:			
What happens (symptoms):			

The issue results in an undefined behavior of the ECU due to invalid index RAM write access in the bounds of the array.			
When does this happen:			

The issue occurs always at runtime if VAR arrays are used by the component.			
In which configuration does this happen:			

Any configuration where the Post Build Variance Support is true in the module configuration.			
AND			
all component configuration data is different in variants.			
AND			
<MSN>WriteOutOfBoundsWriteProtectionStrategy is configured to INDEX_SATURATION.			
This feature is classified by the most components with the "WARNING" "The feature must never be used in productive builds!".			
Resolution Description:			
Workaround:			

IF			
the component generator offers the parameter <MSN>WriteOutOfBoundsWriteProtectionStrategy			
configure it to NONE			
ELSE			
no workaround available.			
Resolution:			

The described issue is corrected by modification of all affected work-products.			

ESCAN00096411 Undefined ECU behavior due to invalid value access in post build selectable configuration with more than 2 variants	
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	8.07.80, 8.05.80, 8.01.01, 8.03.80, 8.05.02, 8.00.01, 8.07.03, 8.06.01, 8.03.03, 8.04.01, 8.11.01, 8.03.81, 8.07.81
Problem Description: What happens (symptoms): ----- Undefined ECU behavior due to invalid CONST value access in arrays or structs. The effect can be for example: - Det_ReportError is called if configured. - Callbacks can be called or even not if intended. - Memory can be overwritten. When does this happen: ----- Always at runtime if the condition below matches. In which configuration does this happen: ----- Any configuration where the Post Build Variance Support is true in the module configuration AND arrays and structures are of the Configuration Class PRECOMPILE or LINKTIME (Note: Even POST-BUILD configurations have data of the Configuration Class PRECOMPILE or LINKTIME) AND more than 2 configured variants AND the data is reduced. (The not optimized configuration data contains the same subsets of data in multiple variants. This triggers a variant dependent data reduction optimization in the generator. Note, that this is not visible in the generated code since the not optimized configuration data is not output by the generator.)	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096425 IR Analyzer misses data consistency problem	
Component@Subcomponent:	GenTool_IRAnalyzer@Application
First affected version:	0.09.00
Fixed in versions:	1.00.02
Problem Description:	
What happens (symptoms):	

RTE Analyzer does not report a data consistency problem.	
When does this happen:	

During RTE analyzer execution.	
In which configuration does this happen:	

This happens when the analyzed source code contains an unprotected access in a callback and when the accesses in the tasks are protected with interrupt locks.	
This only happens when the callback runs in the context of a preemptive task that can be interrupted by the accesses in the other tasks.	
This error only leads to a problem in the ECU, when the RTE generator generates code that contains this problem.	
At the creation date of this ESCAN, no RTE generator problem that leads to the generation of such code is known.	
Resolution Description:	
Workaround:	

Manually check whether whether accesses to global variables in the callbacks are surrounded by interrupt locks.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096797	EcuM_Shutdown throws a DET in some multicore projects in context of EcuM_Shutdown() and shutdown / reset is not performed
Component@Subcomponent:	SysService_Asr4EcuM@Doc_TechRef
First affected version:	2.00.00
Fixed in versions:	6.00.01
Problem Description:	
What happens (symptoms):	
----- The shutdown / reset of the ECU does not work and the API EcuM_Shutdown() causes a DET with error id ECUM_E_MODULE_NOT_IN_PREPSHUTDOWN.	
When does this happen:	
----- During shutdown of the ECU. Maybe this happens not all the time, this depends on special timing.	
In which configuration does this happen:	
----- In Multicore configurations where EcuM_Shutdown() is called from the OS shutdown hook for each core, and the shutdown hook is not core specific.	
Resolution Description:	
Workaround:	
----- Ensure that the OS shutdown hook calls EcuM_Shutdown only on the master core respectively on the core which is responsible for the ECU shutdown / reset.	
Resolution:	
----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096892	Undefined ECU behavior due to invalid index access if PduRWriteOutOfBoundsWriteProtectionStrategy is INDEX_SATURATION and Post Build Variance Support is true
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	10.01.01, 11.01.03, 13.00.00
Problem Description:	
What happens (symptoms):	
----- The issue results in an undefined behavior of the ECU due to invalid index RAM write access in the bounds of the array.	
When does this happen:	
----- The issue occurs always at runtime if VAR arrays are used by the component.	
In which configuration does this happen:	
----- Any configuration where the Post Build Variance Support is true in the module configuration. AND all component configuration data is different in variants. AND PduRWriteOutOfBoundsWriteProtectionStrategy is configured to INDEX_SATURATION.	
This feature is classified with the "WARNING" "The feature must never be used in productive builds!".	
Resolution Description:	
Workaround:	
----- Configure PduRWriteOutOfBoundsWriteProtectionStrategy to NONE.	
Resolution:	
----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097193 Consider AUTOSAR package for ModeDeclarationGroups to allow ModeDeclarationGroups with same name in different packages

Component@Subcomponent: Rte_Core@Implementation

First affected version: 1.00.00

Fixed in versions: 1.18.00

Problem Description:

What happens (symptoms):

Up to now, in case two mode declaration groups are located in different packages, one of them was picked at random. If these mode declaration groups differ, this results in a generation error or falsely code. Depending on the selected mode declaration group and its usage two scenarios are possible:

Case 1: the Perl code generates a RTE49999 error when accessing non existing modes.

Case 2: the RTE switch API might not work as expected. Runnables or tasks might not be activated or falsely activated. For this, the values for the mode declaration group in

Rte_<SWC>_Type.h differ from the values of

Rte_GetInternalModeIndex_<ModeDeclarationGroupName> function in Rte.c and <SWC> has a mode switch point using the mode declaration group.

When does this happen:

Whenever the definition of the mode declaration group does not match the mode declaration group used. This might toggle from one generation to the next one.

Case 1: During generation.

Case 2: During runtime.

In which configuration does this happen:

Mode declaration groups with the same name are located in different packages AND a mode switch point of a connected mode port of an SWC uses mode group values (function Rte_GetInternalModeIndex_<ModeDeclarationGroupName> in Rte.c) which differ from the expected mode group values (Rte_<SWC>_Type.h)

In addition to this, the following conditions have also to be fulfilled:

Case 1: a non-existing mode is accessed.

Case 2: the accessed mode values or transition values differ.

Resolution Description:

Workaround:

Ensure that mode group declaration names differ.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097384 Service 0x22: Overwritten RAM behind a DCM buffer	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.00.00
Fixed in versions:	9.02.00, 8.06.01
Problem Description:	

ESCAN00097384 Service 0x22: Overwritten RAM behind a DCM buffer

What happens (symptoms):

 The memory location behind a DCM message buffer will be corrupted.
 The possible amount of overwritten memory depends on the largest configured DID with variable length (up to 65533 bytes).
 The possible amount of overwritten memory of a configured paged-DID is the largest configured DCM buffer size minus one byte.

After ECU reset, the normal operation should be possible.

When does this happen:

 Depending on the configured DCM buffer size, maximum number of DIDs per single SID 0x22 (ReadDataByIdentifier) request, the size of the DIDs with variable length or usage of a paged-DID, the following sequences may cause the issue:

[Variant1]: A valid diagnostic request for SID 0x22 with single or multiple DIDs is sent, where:
 - At least one of the requested DIDs is with paged-data access.

Example:

Configuration:

- DCM buffer: 100Byte
- DID0: fixed length length: 50Byte
- DID1: paged DID with length: 500Byte

REQ: 0x22 DID0 DID1

DID0: ReadData() -> application writes 50 bytes

DID1: ReadData() -> application writes up to $100 - (1 + 2 + 50 + 2) = 45$ bytes, and for example we assume all 45 bytes are written. But RTE copies all specified in DID1 C/S interface 100 bytes to the passed by DCM buffer pointer. As a result: in total $55 + 100 = 155$ bytes has been written into the DCM buffer => 55 bytes overwritten behind it.

RES(start): 0x62 DID0 [50Byte] DID1 [45Byte]

DID1: ReadData() -> application writes subsequent data portion, leading to further RAM out-of-boundary access with size of overwritten memory, depending on the transport layer maximum frame size minus one byte (e.g. CanTp using normal addressing: up to 6 bytes).

RES(continued):DID1 [continued]...

[Variant2]: A valid diagnostic request for SID 0x22 with multiple DIDs is sent, where:

- The second or any subsequent DID in the request was configured with variable length, read via C/S PortInterface, mapped to a server runnable with OsTask mapping different than the one of the Dcm_MainFunction/-Worker();

AND

- For the above specified DID the application does not utilize the maximum DID size (i.e. reports to DCM in the corresponding C/S Xxx_ReadDataLength() operation the current DID length to be less than its configured maximum).

AND

- This request shall lead to NRC 0x14 if the requested DID(s) with dynamic length would utilize their maximum size (regardless of the server runnable entity OsTask mapping).

Example:

Configuration:

- DCM buffer: 100Byte
- DID0: fixed length: 50Byte
- DID1: dynamic length: up to 80Byte
- DID2: fixed length: 20Byte

SEQ1: DID1 current variable length < DID1 maximum length so that sum of all requested DID1

ESCAN00097384 Service 0x22: Overwritten RAM behind a DCM buffer

(current) lengths including the protocol response bytes (i.e. the SID and the DIDs) fit the DCM buffer size

REQ: 0x22 DID0, DID1

DID1: ReadDataLength() returns 20Byte

DID0: ReadData() -> application writes 50 Byte

DID1: ReadData() -> application writes only 20 Byte, but RTE copies all DID1 80 bytes to the passed by DCM buffer pointer. As a result: in total $1+2+50+2+80 = 135$ Byte has been written into the DCM buffer => 35 bytes overwritten behind it.

RES: 0x62 DID0 [50Byte], DID1 [20Byte]: Length = $1 + 2 + 50 + 2 + 20 = 75$ fits the DCM buffer

SEQ2: DID1 current variable length < DID1 maximum length so that sum of all requested DID1 (current) lengths including the protocol response bytes (i.e. the SID and the DIDs) fit the DCM buffer size

REQ: 0x22 DID0, DID1, DID2

DID1: ReadDataLength() returns 20Byte

DID0: ReadData() -> application writes 50 Byte

DID1: ReadData() -> application writes only 20 Byte, but RTE copies all DID1 80 bytes to the passed by DCM buffer pointer. As a result: in total $1+2+50+2+80 = 135$ Byte has been written into the DCM buffer => 35 bytes overwritten behind it.

DID2: ReadData() -> application writes 20 Byte, but this call does not overwrite memory, since the pointer passed to the application is calculated considering the current size of DID1 (i.e. 20bytes).

RES: 0x62 DID0 [50Byte], DID1 [20Byte], DID2 [20Byte] : Length = $1 + 2 + 50 + 2 + 20 + 2 + 20 = 97$ fits the DCM buffer

SEQ3: DID1 variable length <= DID1 maximum length so that sum of all requested DID1 (current) lengths including the protocol response bytes (i.e. the SID and the DIDs) does not fit the DCM buffer size

REQ: 0x22 DID0, DID1

DID1: ReadDataLength() returns 80Byte (maximum DID1 size)

RES: 0x7F 0x22 0x14 --> No issue, since DCM rejects the request with no data reading. Positive response length would be: $1 + 2 + 50 + 2 + 80 = 135$ does not fit the DCM buffer => NRC 0x14

In which configuration does this happen:

- There is an RTE used in the ECU project.

AND

- Service 0x22 is supported and handled by DCM (in Dcm_Cfg.h: #define DCM_SVC_22_SUPPORT_ENABLED == STD_ON)

AND

- There is at least one DID with a data element (DcmDspData) configured to:

- support paged read access (in Dcm_Cfg.h: #define

DCM_DIDMGR_OPCLS_READ_PAGED_ENABLED == STD_ON)

OR

- have variable size (in Dcm_Cfg.h: #define

DCM_DIDMGR_STATIC_DID_OPTYPE_READ_LENGTH_ENABLED == STD_ON)

AND

- Service 0x22 allows more than single DID per request

AND

- The access to those data elements is configured to be via a C/S interface (in DCM ECUC: /Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort == USE_DATA_ASYNCH_CLIENT_SERVER or USE_DATA_SYNCH_CLIENT_SERVER or

ESCAN00097384 Service 0x22: Overwritten RAM behind a DCM buffer

USE_PAGED_DATA_ASYNCH_CLIENT_SERVER)

AND

- The server runnable entity of those data elements has OsTask mapping other than the one the Dcm_MainFunction/-Worker() is mapped to.

Resolution Description:

Workaround:

- Avoid OsTask mapping of affected DID data element server runnable entities to let RTE optimize the C/S calls to simple C-function calls without data copies.

OR

- Do not use RTE C/S ports but simple callout functions (i.e. specify for DcmDspDataUsePort one of the applicable XXX_FNC values).

OR

- Omit usage of paged-DIDs, increasing the DCM buffer to fit all the paged-DID data at least in a single DID request of SID 0x22.

AND

- Reduce number of DIDs allowed per single SID 0x22 request to a single DID.

OR

- Omit configuration of DIDs with variable size where the DID size is smaller than the configured maximum and will never change (i.e. the ODX /CDD file specifies the DID from diagnostic client side, but the ECU will always report a fixed length).

OR

- Increase the DCM buffer size so that the DID with maximum variable size will always fulfill the equation:

DCM buffer size $\geq 1 + (N * (2 + \text{DID maximum size}))$

where N is the maximum number of DIDs allowed per single SID 0x22 request.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097518 CRC32 calculations deliver wrong results**Component@Subcomponent:** MemService_AsrNvM@Implementation**First affected version:** 5.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

CRC32s calculated internally by NVM are not as specified by AUTOSAR, i.e. the results may differ, depending on number of single CRC library calls done per NVM block. Calculated values are still CRCs, but they don't match the results from using corresponding standardized CRC32 calculations

Since CRC handling is done internally, this is usually not visible to users.

The issue becomes visible, if NVM's configuration changed between a write and a read request (see below): Data may become unreadable due to failed CRC check.

When does this happen:

It happens at run-time during CRC calculation. However this behavior is symmetric, i.e. calculated CRC during writes match the CRC calculated during reads. Data can be written and read back as expected.

In which configuration does this happen:

It happens for all blocks having CRC (NvMBlockUseCrc) enabled, and CRC type (NvMBlockCrcType) was set to CRC32 .
If (in a running project), the number of "Bytes per MainFunction" (NvMCrcNumOfBytes) was changed, existing data become unreadable, because same data result in different CRC.

Resolution Description:

Workaround:

In a running project's configuration don't change the number of "Bytes per MainFunction" (NvMCrcNumOfBytes).

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097644 RTE dereferences NULL_PTR after execution of a mapped server runnable

Component@Subcomponent: Rte_Core@Implementation

First affected version: 1.13.00

Fixed in versions: 1.18.00

Problem Description:

What happens (symptoms):

RTE dereferences a null pointer after a mapped server runnable has been executed.
This usually results in an os trap.

When does this happen:

During runtime when a client with lower task priority calls a mapped server runnable with higher priority.

In which configuration does this happen:

This happens when multiple clients are connected to the same server, and the server runnable is mapped to a task with higher priority than at least one of the clients. This happens only for synchronous client server communication.

Resolution Description:

Workaround:

- map the server runnable to a task with lower priority than the client tasks
or

- create a proxy component that forwards the client calls to the server.

To do so a server port + server runnable is created for each client.

Each client is connected to the corresponding proxy server port.

A single proxy client port is connected to the server.

The server runnables of the proxy component calls the server through the single client port.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097801	Undefined ECU behavior due to invalid index access if <MSN>WriteOutOfBoundsWriteProtectionStrategy is INDEX_SATURATION and Post Build Variance Support is true
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	13.03.03, 11.00.01, 14.01.00
Problem Description:	
What happens (symptoms):	

The issue results in an undefined behavior of the ECU due to invalid index RAM write access in the bounds of the array.	
When does this happen:	

The issue occurs always at runtime if VAR arrays are used by the component.	
In which configuration does this happen:	

Any configuration where the Post Build Variance Support is true in the module configuration.	
AND	
all component configuration data is different in variants.	
AND	
PduRWriteOutOfBoundsWriteProtectionStrategy is configured to INDEX_SATURATION.	
This feature is classified with the "WARNING" "The feature must never be used in productive builds!".	
Resolution Description:	
Workaround:	

Configure PduRWriteOutOfBoundsWriteProtectionStrategy to NONE.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097829 Service 0x22: Overwritten call stack

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 7.02.00

Fixed in versions: 8.06.02, 9.03.00

Problem Description:

What happens (symptoms):

 The call stack will be corrupted leading to indeterminable behavior.
 The possible amount of overwritten memory depends on the largest configured DID.
 After ECU reset, the normal operation should be possible.

When does this happen:

-
- When any asynchronous DID is requested via service 0x22.
 - AND
 - The access to the DID specific data elements is configured to be via a C/S interface (in DCM ECUC: /Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort == USE_DATA_ASYNC_CLIENT_SERVER or USE_PAGED_DATA_ASYNC_CLIENT_SERVER)
 - AND
 - The maximal length of that DID is larger than one byte.
 - AND
 - The read operation of that DID is cancelled due to
 - an interruption by a request of another tester with higher priority
 - OR
 - Reaching the RCR-RP limit.

In which configuration does this happen:

-
- Service 0x22 is supported and handled by DCM (in Dcm_Cfg.h: #define DCM_SVC_22_SUPPORT_ENABLED == STD_ON)
 - AND
 - There is at least one DID with a data element (DcmDspData) configured to support paged read access (in Dcm_Cfg.h: #define DCM_DIDMGR_OPCLS_READ_PAGED_ENABLED == STD_ON)
 - AND
 - The access to any data element is configured to be via a C/S interface (in DCM ECUC: /Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort == USE_DATA_ASYNC_CLIENT_SERVER or USE_PAGED_DATA_ASYNC_CLIENT_SERVER)
 - AND
 - DCM shall prioritize two or more diagnostic clients (e.g. OBD and UDS clients) (in Dcm_Cfg.h #define DCM_NET_PROTOCOL_PRIORITISATION_ENABLED == STD_ON)
 - OR
 - RCR-RP transmission limitation is supported (in Dcm_Cfg.h #define DCM_DIAG_RCRRP_LIMIT_ENABLED == STD_ON)
 - AND
 - There is an RTE used in the ECU project.
 - AND
 - The server runnable entity of those data elements has OsTask mapping other than the one the Dcm_MainFunction/-Worker() is mapped to.

Resolution Description:

ESCAN00097829 Service 0x22: Overwritten call stack

Workaround:

-
- Avoid OS-Task mapping of affected DID data element server runnable entities to let RTE optimize the C/S calls to simple C-function calls without data copies.
- OR
- Do not use RTE C/S ports but simple callout functions (i.e. specify for DcmDspDataUsePort one of the applicable XXX_FNC values).
- OR
- Omit usage of paged-DIDs, increasing the DCM buffer to fit all the paged-DID data at least in a single DID request of SID 0x22.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097901 Rx Deferred Event Cache leads to unexpected ECU behaviour under high load

Component@Subcomponent: Il_AsrComCfg5@Implementation

First affected version: 8.01.00

Fixed in versions: 12.00.02, 13.03.03, 14.00.01

Problem Description:

What happens (symptoms):

The usage of deferred event cache leads to unexpected ECU behaviour under high load.

When does this happen:

Error may occur during run time, whenever it's temporarily required to open the interrupt locks. It is most likely to occur whenever Com_RxIndication is called in high frequency.

In which configuration does this happen:

ComDeferredEventCacheSupport == TRUE

Resolution Description:

Workaround:

(Increase ComRxDeferredEventCacheSize
AND
increase ComRxDeferredProcessingISRLockThreshold (if configurable)
AND
increase ComRxDeferredNotificationCacheSize (if configurable))

OR

Deactivate ComDeferredEventCacheSupport

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098374 Memory corruption during module initialization

Component@Subcomponent: Diag_Asr4Dem@Implementation

First affected version: 7.00.00

Fixed in versions: 14.05.00, 12.01.05

Problem Description:

What happens (symptoms):

 Undefined behavior of the ECU due to out of bounds memory write access.
 Several bytes (number depends on the configuration, see below) of the memory following
 Dem_Cfg_DebounceData[<last valid>] is overwritten.

When does this happen:

 On call of Dem_NvM_InitDebounceData.
 The API is intended to be called by the NvM module in order to (re-)initialize the data in case the
 non-volatile memory has never been stored or was corrupted.
 The Dem calls the API itself in case of a changed implementation version or changed configuration
 Id.

In which configuration does this happen:

 (Any /Dem/DemConfigSet/DemEventParameter/DemEventClass/DemDebounceAlgorithmClass/
 DemDebounceCounterBased/DemDebounceCounterStorage == TRUE)
 AND
 (Dem/DemGeneral/DemUseMemcpyMacros == TRUE)

In such a configuration the number of overwritten bytes equals the number of events requiring de-
 bounce counter storage (Dem/DemConfigSet/DemEventParameter/DemEventClass/
 DemDebounceAlgorithmClass/DemDebounceCounterBased/DemDebounceCounterStorage ==
 TRUE).

Resolution Description:

Workaround:

 Use VStdLib implementation of Dem_MemSet (Dem/DemGeneral/DemUseMemcpyMacros ==
 FALSE)
 OR
 if using an own implementation of Dem_MemSet_Macro(destination_ptr, value_byte,
 length_in_byte), cast 'destination_ptr' to an uint8 pointer, before writing to it
 (for using an own implementation refer to the Dem Technical Reference).

Resolution:

 The described issue is corrected by modification of all affected work-products.

ESCAN00098443 SchM_Init starts triggering runnables before Rte_Start when schedule tables are configured	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.05.00
Fixed in versions:	1.18.00
Problem Description: What happens (symptoms): ----- Runnables are triggered directly after SchM_Init although Rte_Start was not called, yet. A development error RTE_E_DET_UNINIT is thrown when the uninit checks are enabled. When does this happen: ----- During runtime. In which configuration does this happen: ----- This happens when the configuration contains tasks that contain schedulable entities and runnable entities with periodic triggers and when the task type is configured to basic task.	
Resolution Description: Workaround: ----- Set the task type to extended. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098513 PanicHook is called or system enters endless loop on stack overflow**Component@Subcomponent:** Os_CoreGen7@Implementation**First affected version:** 1.01.07**Fixed in versions:****Problem Description:**

What happens (symptoms):

The PanicHook is called (if configured) while a call of the ProtectionHook would be expected.

When does this happen:

This happens with high probability if a task, ISR or hook function has caused a stack overflow which was detected by the MPU.

In which configuration does this happen:

This happens in scalability classes SC3 and SC4 if StackMonitoring is configured.**Resolution Description:**

Workaround:

On platforms where the MPU stays active in privileged mode, StackMonitoring may be switched off. The ProtectionHook is called as expected for a stack overflow then. This will have no influence on the safety if the MPU is set up as described in the technical reference, chapter "Stack Supervision by MPU".

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098598 Rte_Call returns invalid data	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.04.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

Rte_Call returns invalid return values and output parameters because it does not wait until the server is executed.	
When does this happen:	

During runtime.	
In which configuration does this happen:	

This happens when all of the following conditions evaluate to true:	
<ul style="list-style-type: none"> - the configuration contains client-server calls where client and server are mapped to different tasks - the server task has a smaller priority or equal priority than the client task, is mapped to another core or uses APIs with WaitEvent() or Schedule() calls - the server runnable has configured server call points - the client ports of the server runnable are not connected 	
Problematic APIs can be identified by searching for SetEvent calls without WaitEvent calls in the same Rte_Call API.	
Resolution Description:	
Workaround:	

Connect unconnected client-ports.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098617 Shutdown / Reset of a multicore ECU is not performed as expected	
Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	5.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

A requested shutdown / reset may not be finished completely by the EcuM.	
When does this happen:	

If a shutdown / reset is requested.	
In which configuration does this happen:	

This can only happen on multicore devices that execute the EcuM on multiple cores with different levels of diagnostic coverage (e.g. a system with a core with lock-step and a core without lock-step).	
Resolution Description:	
Workaround:	

No workaround is available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099129 Implicit write or read accesses do not work	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.07.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

Data passed to Rte_IWrite or Rte_IWriteRef APIs never reaches the receiver. Rte_IRead returns invalid data.	
When does this happen:	

During runtime.	
In which configuration does this happen:	

This happens when the configuration contains multiple implicit accesses to the same data element and when one access runs in a task that cannot be interrupted by other accesses to the data element and the other access runs in a task that can be interrupted by other accesses.	
Resolution Description:	
Workaround:	

Use explicit instead of implicit communication.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099150 External Rte_Read does not return initial value	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.04.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

Before first reception external Rte_Read returns uninitialized data instead of initial value.	
When does this happen:	

During runtime	
In which configuration does this happen:	

This happens when either	
a) FanIn is configured with signal specific transformation	
or	
b) if the data element is a primitive byte array (uint8[]) and LdCom is used without data transformation or E2E protection is configured for the data element.	
For a) problematic Rte_Read APIs can be found by searching for accesses to a variable Rte__callbackId	
The problem occurs when the API contains a call to ComXf_Inv and no call to Com_ReceiveSignalGroupArray and when	
no transformer length check is contained in the API: if (Rte__Length == 0)	
For b) problematic Rte_Read APIs can be found by searching for Rte_MemCpy and Rte_MemCpy32 operations that copy from a Rte_LdCom variable that is filled in a Rte_LdComCbRxIndication callback	
Rte_MemCpy(*(data), Rte_LdCom_, <len>);	
The problem occurs when no transformer length check is contained in the API: if (Rte__Length == 0)	
Resolution Description:	
Workaround:	

Enable data transformation.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099276 Memory corruption in CompareKey	
Component@Subcomponent:	SysService_AsrCryFord@Implementation
First affected version:	1.00.00
Fixed in versions:	2.00.01
Problem Description:	
What happens (symptoms):	

Out of bounds memory write access.	
This can be detected with:	
- unexpected ECU behavior.	
When does this happen:	

Whenever CompareKey of the Security Access is called.	
In which configuration does this happen:	

Whenever the static variable CryFord_MacSecAccessWorkSpace is not directly followed by the static variable CryFord_MacSecAccessVerify_Buffer in the binary. This can be seen in the map file.	
If SysService_AsrCryFord@Implementation version is < 2.00.00, then CryFord_MacSecAccessWorkSpace is located in CryFord_Cfg.c, otherwise it is located in CryFord.lib.	
Resolution Description:	
Workaround:	

Link CryFord_MacSecAccessVerify_Buffer directly behind CryFord_MacSecAccessWorkSpace.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099646		Schedulable entity not triggered when entity is mapped to a basic task and cyclic trigger implementation is set to none	
Component@Subcomponent:		Rte_Core@Implementation	
First affected version:		1.05.00	
Fixed in versions:		1.19.00	
Problem Description:			
What happens (symptoms):			

A schedulable entity is not triggered by the RTE.			
When does this happen:			

Always during runtime.			
In which configuration does this happen:			

This happens when all of the following conditions are true			
- the configuration contains a schedulable entity			
- the schedulable entity also is a runnable entity			
- the schedulable entity is mapped to a basic task			
- the cyclic trigger implementation is configured to none			
- the basic task also contains other entities			
Resolution Description:			
Workaround:			

Configure an extended task.			
Resolution:			

The described issue is corrected by modification of all affected work-products.			

ESCAN00099865 Error in Silent Analysis	
Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	5.00.00
Fixed in versions:	8.00.00
Problem Description:	
What happens (symptoms):	

In theory an invalid channel could be used for accessing the NmState variable array. This would lead to an out of bounds memory write access.	
NOTE: From technical point of view, this issue should not occur. Nevertheless it cannot be guaranteed, because there is no technical verification that the greatest index is smaller or equal to the array size.	
When does this happen:	

Each time the NmState is written after memory initialization.	
In which configuration does this happen:	

Each configuration.	
Resolution Description:	
Workaround:	

Check that the CanNm_GetSizeOfChannelConfig() is never greater than the CANNM_NUMBER_OF_CANNM_CHANNELS.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099885 Runnable with mode disabling not triggered when no mode switch point is configured	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.13.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

A runnable with mode disabling dependency is never triggered.	
When does this happen:	

Always during runtime.	
In which configuration does this happen:	

This happens when the mode machine cannot leave the initial mode because no mode switch points are configured and when the runnable is configured to be active in the initial mode.	
Resolution Description:	
Workaround:	

Connect the mode port and configured mode switch points.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

2.2 Runtime Issues without Workaround

Runtime issues without a workaround require an update of the software delivery in case the issue affects the ECU overall functionality. The effect of an issue to the ECU functionality has to be analyzed by the customer as the software usage and its configuration is not known by Vector. The risk of change has also to be taken into account.

Index

ESCAN00092995	CAN-FD message without BRS will not be received DrvCan_Mpc5700McanLI@Implementation
ESCAN00095297	Service 0x27: Wrong prioritisation between NRC 0x24 and 0x13 Diag_Asr4Dcm@Implementation
ESCAN00095298	No transmit cancellation when calling Can_CancelTx() / CanIf_CancelTransmit() DrvCan__coreAsr@Implementation
ESCAN00095778	Wrong filterInitState Value calculation IL_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00096100	Main function tick time resolution smaller than 1 ms is not supported Nm_Asr4NmCan@GenTool_GeneratorMsr
ESCAN00096102	Main function tick time resolution smaller than 1 ms is not supported Ccl_Asr4SmCan@GenTool_GeneratorMsr
ESCAN00096203	Validator: Service 0x31: Routine info byte not considered in calculation of positive response length for the buffer size estimation Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00096207	Service 0x27: The attempt counters only the first eight security levels are read at power on/reset Diag_Asr4Dcm@Implementation
ESCAN00096247	Service 0x22/0x2A/0x2F: ECU returns invalid data Diag_Asr4Dcm@Implementation
ESCAN00096295	Debounce data not persisted after manual NV synchronization Diag_Asr4Dem@Implementation
ESCAN00096492	Aging counter is incremented incorrectly Diag_Asr4Dem@Implementation
ESCAN00096547	Pending Status Bit and WIR Status Bit are reset too early / unexpectedly Diag_Asr4Dem@Implementation
ESCAN00096776	Execution of daq lists may fail. Cp_Asr4Xcp@Implementation
ESCAN00096960	CycleState is not stored in NvRam Diag_Asr4Dem@Implementation
ESCAN00097111	Service 0x2C: Reading DDDID contains invalid response data Diag_Asr4Dcm@Implementation
ESCAN00097324	Missing TxAcknowledge in configurations with multiple variants Rte_Core@Implementation
ESCAN00097367	DAQ overrun indication not deactivateable - max PID limited to 0x7B Cp_Asr4Xcp@Implementation
ESCAN00097699	Dem_SetWIRStatus is processed and returns E_OK for unavailable events Diag_Asr4Dem@Implementation
ESCAN00097735	Paged-Buffer: Positive response with corrupted data Diag_Asr4Dcm@Implementation
ESCAN00097748	Memory corruption of management information leads to dead lock of corresponding block If_AsrIfFeeSmallSector@Implementation
ESCAN00097848	Dem_GetWIRStatus returns E_OK for unavailable events Diag_Asr4Dem@Implementation
ESCAN00097849	Dem_GetEventEnableCondition returns E_OK for unavailable events Diag_Asr4Dem@Implementation

Index

ESCAN00097850	Dem_GetEventAvailable returns AvailableStatus 'True' for events unavailable in variant Diag_Asr4Dem@Implementation
ESCAN00097911	Deferred Rx PDUs are not received if ComDeferredEventCacheSupport is enabled IL_AsrComCfg5@Implementation
ESCAN00097951	Wrong marshalling/ unmarshalling of <XInt64> Signals IL_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00098006	Declined Immediate Nm Transmissions is retried later than expected Nm_Asr4NmCan@Implementation
ESCAN00098203	Service 0x3E: S3 timer reloaded for different protocol Diag_Asr4Dcm@Implementation
ESCAN00098248	Immediate priority block requests during WriteAll may lead to unsuccessful WriteAll MemService_AsrNvM@Implementation
ESCAN00098392	S3 timer reloaded by any request from different tester Diag_Asr4Dcm@Implementation
ESCAN00098604	Service 0x3E: Keep-Alive timer reloaded for different protocol Diag_Asr4Dcm@Implementation
ESCAN00098606	Keep-Alive timer reloaded by any request from different tester Diag_Asr4Dcm@Implementation
ESCAN00098938	Unconfirmed DTCs need healing before aging Diag_Asr4Dem@Implementation
ESCAN00099092	CONNECT will stop a running DAQ measurement from resume mode Cp_Asr4Xcp@Implementation
ESCAN00099171	Value of 'Cycles Since First Failed' is 0 Diag_Asr4Dem@Implementation
ESCAN00099197	Wrong DLC calculation for zero bit signals IL_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00099383	Failure to collect data via Client/Server or Sender/Receiver R-Port results in random data Diag_Asr4Dem@Implementation
ESCAN00099495	Measurement failure when more than 255 measurement values are configured and used Cp_Asr4Xcp@Implementation
ESCAN00099586	ErrorFrame after startup with wakeup validation Ccl_Asr4SmCan@Implementation
ESCAN00099742	Status block is not immediately written to NvRAM after a clear request Diag_Asr4Dem@Implementation
ESCAN00099810	Event heals too early after displacement Diag_Asr4Dem@Implementation
ESCAN00099905	XCP internal memory not initialized correctly in huge configurations. Cp_Asr4Xcp@Implementation

ESCAN00092995 CAN-FD message without BRS will not be received	
Component@Subcomponent:	DrvCan_Mpc5700McanLI@Implementation
First affected version:	2.09.00
Fixed in versions:	2.12.00
Problem Description: What happens (symptoms): ----- A CAN-FD message without Bitrate Switching will: - not be received by the upper layers. - produce a DET Error (CAN_E_PARAM_DLC) for messages with a DLC greater than 8 bytes. When does this happen: ----- During runtime, always and immediately, when a CAN-FD message is received without Bitrate Switching (CAN-FD BRS bit) set. In which configuration does this happen: ----- Only in configurations using CAN-FD Rx messages without Bitrate switching.	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00095297 Service 0x27: Wrong prioritisation between NRC 0x24 and 0x13	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.00.00
Fixed in versions:	8.04.00
Problem Description: What happens (symptoms): ----- NRC 0x13 is responded instead of NRC 0x24. When does this happen: ----- If a seed request was expected but a too long key request was sent. Note: An NRC 0x13 caused by the minimum length check according to ISO 14229-1:2013 Figure 6 is not affected by this issue. In which configuration does this happen: ----- - Service 0x27 is supported (in Dcm_Cfg.h: DCM_SVC_27_SUPPORT_ENABLED == STD_ON)	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

**ESCAN00095298 No transmit cancellation when calling
Can_CancelTx() / CanIf_CancelTransmit()****Component@Subcomponent:** DrvCan__coreAsr@Implementation**First affected version:** 2.00.00**Fixed in versions:** 5.03.04, 5.05.06, 5.04.04, 4.06.06, 5.06.00**Problem Description:**

What happens (symptoms):

Calling Can_CancelTx() / CanIf_CancelTransmit() will not lead to a cancellation.
Confirmation will be unexpectedly called

When does this happen:

Only under specific circumstances:
same send request (same PDU) is stored multiple in the basic CAN hardware object, and API
Can_CancelTx() is called.

In which configuration does this happen:

MicroSar4:

CanMultiplexedTransmission == true / (CAN_MULTIPLEXED_TRANSMISSION == STD_ON)
and
CanCancelSupportApi == true / (CAN_CANCEL_SUPPORT_API == STD_ON)

MicroSar3:

CanMultiplexedTransmission == true / (CAN_MULTIPLEXED_TRANSMISSION == STD_ON)
and
CanCancelSupportApi == true / (CAN_CANCEL_SUPPORT_API == STD_ON)
and
CanTP Version >= 1.11.00 && CanTP Version < 1.27.04
and
CANTP_CANCEL_TRANSMIT == STD_ON or CANTP_TC == STD_ON or CANTP_CANIF_ENABLE_TC
== STD_ON

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095778 Wrong filterInitState Value calculation	
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	11.00.00
Fixed in versions:	13.03.00
Problem Description: What happens (symptoms): ----- The filterInitState value is not calculated correctly and set to a wrong value. Therefore a wrong initial TxMode (TxModeTrue / TxModeFalse) is set during initialization of COM When does this happen: ----- During calculation of filterInitStates for each rx or tx signal/groupSignal. In which configuration does this happen: ----- In all configs with tx or rx filter configured and filterAlgorithm is equals MASKED_NEW_EQUALS_X/ MASKED_NEW_DIFFERS_X.	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096100 Main function tick time resolution smaller than 1 ms is not supported	
Component@Subcomponent:	Nm_Asr4NmCan@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	8.00.00
Problem Description: What happens (symptoms): ----- AUTOSAR defines a float data type to configure the main function tick time. The generator truncates the configured tick time to a millisecond. e.g. 1.5 ms tick time is configured, but the counter values generated in the code are based on a 1 ms tick time. As a result, wrong timer values are used by the module When does this happen: ----- always In which configuration does this happen: ----- If a main function tick time with a resolution smaller than 1 ms is configured	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096102 Main function tick time resolution smaller than 1 ms is not supported	
Component@Subcomponent:	Ccl_Asr4SmCan@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- AUTOSAR defines a float data type to configure the main function tick time. The generator truncates the configured tick time to a millisecond. e.g. 1.5 ms tick time is configured, but the counter values generated in the code are based on a 1 ms tick time. As a result, wrong timer values are used by the module When does this happen: ----- always In which configuration does this happen: ----- If a main function tick time with a resolution smaller than 1 ms is configured	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096203 Validator: Service 0x31: Routine info byte not considered in calculation of positive response length for the buffer size estimation

Component@Subcomponent: Diag_Asr4Dcm@GenTool_GeneratorMsr

First affected version: 7.01.00

Fixed in versions: 8.05.00, 7.02.02

Problem Description:

What happens (symptoms):

During validation of Dcm module no validation error for insufficient buffer size is displayed. As a result of this wrong estimation for SID 0x31 (Routine Control), a Routine C/S port may overwrite a single byte memory location behind the Dcm diagnostic buffer.

When does this happen:

Configurator 5: During validation of Dcm module.

DCM code: At runtime each and every time the affected Routine is started via valid diagnostic request (i.e. correct format, session, security).

In which configuration does this happen:

- Service 0x31 (RoutineControl) is handled by DCM (in Dcm_Cfg.h #define DCM_SVC_31_SUPPORT_ENABLED == STD_ON);
AND

- At least 1 DcmDspRoutine exists with an explicitly configured routine info byte (RIB) parameter (Parameter /MICROSAR/Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineInfoByte exists (for DCM versions 8.03.XX and older: in Dcm_Cfg.h #define DCM_RIDMGR_SUPPORT_ROUTINEINFOBYTE_ENABLED == STD_ON);
AND

- Routine sub-function specific length of the response data (ResDataLen) fulfills the equation: $\text{ResDataLen} > \text{DcmDslBufferSize} - 5$ (i.e. response data exceeds the remaining buffer after exclusion of response protocol header: SID, SF, RID and the RIB);

Note: The explicit modelling of the routine info byte is typical for OBD projects where a UDS Routine overlaps with an OBD one.

But since MSR4 DCM allows the routine info byte usage also for any Routine, the issue is not limited only to OBD projects

Resolution Description:

Workaround:

Increase the size of all DCM buffers (DcmDslBuffer) by one byte.

Optimization hint:

You may adapt only those buffers that are referred by a diagnostic protocol (DcmDslProtocolRow) which support SID 0x31 (DcmDslProtocolRow/DcmDslProtocolSIDTable refers to a DcmDsdServiceTable containing diagnostic service with ID 0x31).

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096207	Service 0x27: The attempt counters only the first eight security levels are read at power on/reset
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	4.01.00
Fixed in versions:	7.02.01, 8.05.00
Problem Description:	
What happens (symptoms):	

The following scenarios are possible after ECU power on/reset for the affected security levels:	
1) The DCM will accept a new "GetSeed" request instead of sending NRC 0x37 (delay time not expired). OR 2) The DCM will allow more attempts prior sending the NRC 0x36 (number of attempts exceeded) to an invalid key.	
When does this happen:	

1) If during the last ECU operation cycle the DCM has responded with NRC 0x36 and the ECU has been reset/powered down while the delay time was still active (i.e. during an attempt to shorten the penalty delay via reboot). 2) If during the last ECU operation cycle the DCM has registered some failed unlock attempts.	
In which configuration does this happen:	

- Service 0x27 (SecurityAccess) is handled by DCM (in Dcm_Cfg.h #define DCM_SVC_27_SUPPORT_ENABLED is set to STD_ON); AND - The NvM storage of an security access attempt counter is required (in Dcm_Cfg.h #define DCM_STATE_SEC_ATT_CNTR_EXT_STORAGE_ENABLED is set to STD_ON); AND - There are more than eight security levels "DcmDspSecurityRow"s configured; AND - At least one security level (DcmDspSecurityRow), starting with the ninth one, needs NvM storage of the attempt counter (/Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDspSecurityAttemptCounterEnabled == TRUE);	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096247 Service 0x22/0x2A/0x2F: ECU returns invalid data

Component@Subcomponent: Diag_Asr4Dcm@Implementation
First affected version: 7.02.00
Fixed in versions: 8.05.00

Problem Description:

What happens (symptoms):

 On valid request of services:

0x22 (ReadDataByIdentifier),
 0x2A (ReadDataByPeriodicIdentifier),
 0x2F (InputOutputControlByIdentifier)

the ECU responds with expected message length, but invalid DID data (i.e. the value of the last DID data element is located at the beginning of the DID response record, overwriting the first N data bytes, where N is the last data element length).

When does this happen:

 At runtime each time a valid request of one of the above services is processed on a DID that:
 - has more than one data element;
 AND
 - one or more data elements after the first one have asynchronous read access;

In which configuration does this happen:

 - There is any asynchronous DID configured (in Dcm_Cfg.h:
 DCM_DIDMGR_OPCLS_READ_ASYNC_ENABLED == STD_ON)
 AND
 - At least one DID with more than one data element (in Dcm_Cfg.h:
 DCM_DIDMGR_MSIG_OPTYPE_READ_ENABLED == STD_ON)
 AND
 - An affected diagnostic service (i.e. 0x22/0x2A) is internally handled in DCM (in Dcm_Cfg.h:
 DCM_SVC_22_SUPPORT_ENABLED == STD_ON resp. DCM_SVC_2A_SUPPORT_ENABLED ==
 STD_ON)
 OR
 - Service 0x2F, is internally handled in DCM (in Dcm_Cfg.h: DCM_SVC_2F_SUPPORT_ENABLED
 == STD_ON)
 AND
 - Service 0x2F IODID has more than one data element (in Dcm.c:
 DCM_DIDMGR_MSIG_OPTYPE_IO_ANY_ENABLED == STD_ON)
 AND
 - Service 0x2F shall return the IODID data (in Dcm_Cfg.h:
 DCM_DIDMGR_IODID_READ_SUPPORT_ENABLED == STD_ON)
 AND
 - Service 0x2F refers an asynchronous IODID. (in Dcm_Cfg.h:
 DCM_DIDMGR_ASYNC_IODID_SUPPORT_ENABLED == STD_ON)

Resolution Description:

ESCAN00096247 Service 0x22/0x2A/0x2F: ECU returns invalid data

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096295 Debounce data not persisted after manual NV synchronization

Component@Subcomponent: Diag_Asr4Dem@Implementation

First affected version: 7.00.00

Fixed in versions: 12.01.03, 13.06.00, 10.00.03

Problem Description:

What happens (symptoms):

After Dem's API Dem_RequestNvSynchronization the Nvm is not informed (call of NvM_WriteBlock does not occur for the debounce data block) to immediately persist the counter based Debounce data .

Nevertheless, in a full shutdown sequence of the ECU, debounce data are correctly persisted.

When does this happen:

Data loss of debounce data always occurs when ECU is restarted without a full shut down sequence (e.g. interrupted power supply) although API Dem_RequestNvSynchronization has been called before.

In which configuration does this happen:

Events with counterbased debouncing have Dem/DemConfigSet/DemEventParameter/
DemEventClass/DemDebounceAlgorithmClass/DemDebounceCounterBased/
DemDebounceCounterStorage == TRUE
AND
Dem/DemGeneral/DemNvSynchronizeSupport == TRUE

Resolution Description:

Workaround:

No workaround available.

Nevertheless, under normal operation conditions (full shutdown sequence of ECU), debounce data are consistently persisted.

Resolution:

The described issue is corrected by modification of the code.

ESCAN00096492 Aging counter is incremented incorrectly	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	4.00.00
Fixed in versions:	13.06.00, 12.01.03
Problem Description: What happens (symptoms): ----- Aging counter of the event is incremented earlier than expected. This can result in the event getting aged one aging cycle earlier. When does this happen: ----- When aging of the event is progressed by manually restarting the operation cycle of the event (reported operation cycle state as stop and start). It does not happen when operation cycle is implicitly restarted (reporting as an already started operation cycle as start again). In which configuration does this happen: ----- /MICROSAR/Dem/DemGeneral/DemAgingBehavior == AGING_TYPE_2 OR /MICROSAR/Dem/DemGeneral/DemAgingBehavior == AGING_TYPE_3 OR /MICROSAR/Dem/DemGeneral/DemAgingBehavior == AGING_TYPE_5 AND /MICROSAR/Dem/DemConfigSet/DemEventParameter/DemEventClass/DemAgingCycleRef == /MICROSAR/Dem/DemConfigSet/DemEventParameter/DemEventClass/DemOperationCycleRef	
Resolution Description: Workaround: ----- No workaround available. Resolution: -----	

ESCAN00096547 Pending Status Bit and WIR Status Bit are reset too early / unexpectedly

Component@Subcomponent: Diag_Asr4Dem@Implementation

First affected version: 5.00.00

Fixed in versions: 13.06.00, 12.01.03

Problem Description:

What happens (symptoms):

When application reads Pending / WIR status bit for a DTC it can read an incorrect value after restart as:

- the Pending status bit is reset for a DTC, although the DTC has not been tested passed for a whole operation cycle.
- as a result on the next operation cycle end also the WIR status bit will be reset, independent if the event has already healed.

In best case event is healed and status is just reset too early.

Otherwise if event is not healed status bit is incorrect (reset instead of set) for a short time until event is recognized as failed again.

When does this happen:

When the DTC has aged, but is still stored in the event memory, and the ECU is restarted.

In which configuration does this happen:

DemGeneral/DemAgingRetainEnvironmentalData == TRUE

AND

(
DemGeneral/DemAgingBehavior == AGING_TYPE_1 (DEM_AGING_AT_PASSED)

OR

DemGeneral/DemAgingBehavior == AGING_TYPE_4
(DEM_AGING_AT_PASSED_CONT_NOT_FAILED)

)

AND

DemConfigSet/DemEventParameter/DemEventClass/DemAgingCycleCounterThreshold > 0 (for events with aging target 0, the Pending status bit is intentionally reset when the event is aged with the first passed result)

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096776 Execution of daq lists may fail.	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	3.00.00
Problem Description: What happens (symptoms): ----- On platforms where uint8_least is smaller than uint16, execution of daq lists may fail due to a counter variable that is too small. As a result daq measurement might be performed only partially. When does this happen: ----- When more than 255 daq lists are used. In which configuration does this happen: ----- When daq measurement is used	
Resolution Description: Workaround: ----- The basic data types are defined in the file Platform_Types.h Please make sure that the type uint8_least is defined as unsigned short at least (uint16). Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096960 CycleState is not stored in NvRam

Component@Subcomponent: Diag_Asr4Dem@Implementation
First affected version: 7.00.00
Fixed in versions: 13.06.02, 12.01.03, 10.00.03, 14.01.00

Problem Description:

What happens (symptoms):

Events can not be reported due to stopped operation cycle.

When does this happen:

The operation cycle of the event was started and synchronized with NvM using API Dem_RequestNvSynchronization before ECU reset
AND
The ECU did not perform a normal shutdown (e.g. hard reset or Clamp 15 ECU)
AND
the Event is reported after ECU start, and its operation cycle has not been restarted.

Note:

This scenario presumes that the integration correctly synchronizes the Dem state with the NV-Ram using API Dem_RequestNvSynchronization after the operation cycle was started. Otherwise, data loss is expected due to the missing shutdown sequence.

In which configuration does this happen:

DemGeneral/DemOperationCycleStatusStorage == TRUE
AND
Dem/DemGeneral/DemNvSynchronizeSupport == TRUE
AND
Dem/DemGeneral/DemOperationCycle != Dem/DemGeneral/DemRestartCycleOnInitRef
AND
Dem/DemGeneral/DemOperationCycle/DemOperationCycleType != DEM_OPCYC_OBD_DCY

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097111 Service 0x2C: Reading DDDID contains invalid response data

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 1.02.00

Fixed in versions: 8.06.01, 9.01.00

Problem Description:

What happens (symptoms):

The positive response to the diagnostic service 0x22 (ReadDataByIdentifier) of valid and defined DDDID (DynamicallyDefinedDID) has the expected length, but contains unexpected/implausible data.

When does this happen:

Each and every time the affected DDDID is read.

In which configuration does this happen:

- DCM supports and implements diagnostic service 0x2C (DynamicallyDefineDataIdentifier)

AND

- The number of DDDIDs and total number of their source items (i.e. source DIDs or memory blocks) does not fulfill the logical expression:

$(\text{SUM}(\text{DDDID}(X).\text{numSourceItems}, X = 1..N) \leq 256) \text{ OR } (N \geq 256)$

where: N is the total number of DDDIDs in the DCM configuration, DDDID(X).numSourceItems is the number of source items a particular DDDID can hold (i.e. /Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidAccess/DcmDspDidDefine/DcmDspDDDIDMaxElements)

Resolution Description:

Workaround:

Reduce total number of source items per DDDID to fulfill the equation from the affected configuration page.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097324 Missing TxAcknowledge in configurations with multiple variants	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.11.00
Fixed in versions:	1.18.00
Problem Description: What happens (symptoms): ----- Rte_Feedback reports RTE_E_NO_DATA or RTE_E_TIMEOUT although COM notified the transmission. ----- When does this happen: ----- During runtime. ----- In which configuration does this happen: ----- This happens when tx acknowledge is configured for a data element that is mapped to multiple signals (fan-out). Moreover the project needs to use multiple postbuild variants.	
Resolution Description: Workaround: ----- No workaround available. ----- Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097367 DAQ overrun indication not deactivateable - max PID limited to 0x7B	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	3.00.00
Problem Description: What happens (symptoms): ----- The configuration parameter /MICROSAR/Xcp/XcpCmdConfig/XcpDaqAndStim/XcpOverrunIndication has no influence. The feature "overrun indication" always stays enabled. This leads to the fact that the max PID is limited to 0x7B as the MSB is used for overrun indication. When does this happen: ----- Always and immediately In which configuration does this happen: ----- When DAQ measurement is used.	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097699 Dem_SetWIRStatus is processed and returns E_OK for unavailable events	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	6.02.00
Fixed in versions:	12.01.04, 14.04.00
Problem Description: What happens (symptoms): ----- API Dem_SetWIRStatus is processed and returns E_OK when called for an unavailable event. When does this happen: ----- Always and immediately In which configuration does this happen: ----- (Dem/DemGeneral/DemUserControlledWirSupport == TRUE) AND ((Dem/DemConfigSet/DemEventParameter/DemEventAvailableInVariant == FALSE for the concerned event) OR (Dem/DemGeneral/DemAvailabilitySupport == DEM_EVENT_AVAILABILITY and concerned event is set unavailable by API Dem_SetEventAvailable)) Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097735 Paged-Buffer: Positive response with corrupted data

Component@Subcomponent: Diag_Asr4Dcm@Implementation
First affected version: 7.02.00
Fixed in versions: 9.02.00, 8.06.01

Problem Description:

What happens (symptoms):

 ECU sends a positive response with corrupted data.

Note:

The first bytes of the response are always valid so that the response can be recognized as a positive one.

The exact amount of valid bytes depends on the chunk size of the transmission layer and other factors.

When does this happen:

-
- When service 0x19 or 0x22 is requested
 - AND
 - Paged-Buffer is used for transmission of the positive response
 - AND
 - DCM is waiting for next paged data portion because DEM or the application had returned E_PENDING
 - AND
 - DCM wants to defragment the Paged-Buffer, because some data was already transmitted
 - AND
 - The amount of data available for transmission is greater than the amount of data already transmitted since last defragmentation

In which configuration does this happen:

-
- If any service using the paged-buffer is supported (in Dcm_Cfg.h: #define DCM_PAGED_BUFFER_ENABLED == STD_ON)

Resolution Description:

Workaround:

For Service 0x19:

- Deactivate the Paged-Buffer for this service (/Dcm/DcmConfigSet/DcmPageBufferCfg/DcmPagedBufferEnabled)

For Service 0x22:

- Do not use the return value DCM_E_PENDING for ReadData() (paged-data-reading) callouts
- OR
- Do not use paged DIDs at all (/Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort != USE_PAGED_DATA_ASYNCH_CLIENT_SERVER and USE_PAGED_DATA_ASYNCH_FNC)

Resolution:

 The described issue is corrected by modification of all affected work-products.

ESCAN00097748 Memory corruption of management information leads to dead lock of corresponding block	
Component@Subcomponent:	If_AsrIfFeeSmallSector@Implementation
First affected version:	1.00.00
Fixed in versions:	2.00.01, 1.00.03
Problem Description: What happens (symptoms): ----- Memory corruption (e.g. double bit error caused by reset) leads to not repairable NVM/FEE block. Application tries to write a block and FEE always stops processing the job because FLS issues an error while accessing FEE's management information. FEE block can no longer be accessed by read or write jobs. When does this happen: ----- A reset can lead to an error in the flash hardware (e.g. double bit error) which issues a FLS error during read accesses. If FEE's management information is somehow affected by a such an error, FEE will no longer be able to read or write this block. In which configuration does this happen: ----- Every configuration.	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097848 Dem_GetWIRStatus returns E_OK for unavailable events

Component@Subcomponent: Diag_Asr4Dem@Implementation

First affected version: 6.02.00

Fixed in versions: 12.01.04, 14.04.00

Problem Description:

What happens (symptoms):

API Dem_GetWIRStatus returns E_OK when called for an unavailable event.

When does this happen:

Always and immediately

In which configuration does this happen:

From Dem version 7.00.00 on:

(Dem/DemGeneral/DemUserControlledWirSupport == TRUE)

AND

(
(Dem/DemConfigSet/DemEventParameter/DemEventAvailableInVariant == FALSE for the
concerned event)

OR

(Dem/DemGeneral/DemAvailabilitySupport == DEM_EVENT_AVAILABILITY and concerned event is
set unavailable by API Dem_SetEventAvailable)

)

In earlier Dem versions:

(Dem/DemGeneral/DemUserControlledWirSupport == TRUE)

AND

(Dem/DemGeneral/DemAvailabilitySupport == DEM_EVENT_AVAILABILITY and concerned event is
set unavailable by API Dem_SetEventAvailable)

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097849 Dem_GetEventEnableCondition returns E_OK for unavailable events	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	6.02.00
Fixed in versions:	14.04.00, 12.01.04
Problem Description: What happens (symptoms): ----- API Dem_GetEventEnableCondition returns E_OK when called for an unavailable event. When does this happen: ----- Always and immediately In which configuration does this happen: ----- (Dem/DemGeneral/DemEnableConditionSupport) AND ((Dem/DemConfigSet/DemEventParameter/DemEventAvailableInVariant == FALSE for the concerned event) OR (Dem/DemGeneral/DemAvailabilitySupport == DEM_EVENT_AVAILABILITY and concerned event is set unavailable by API Dem_SetEventAvailable)) Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097850 Dem_GetEventAvailable returns AvailableStatus 'True' for events unavailable in variant	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	10.00.00
Fixed in versions:	14.04.00, 12.01.04
Problem Description:	
What happens (symptoms):	

API Dem_GetEventAvailable returns E_OK and AvailableStatus 'True' when called for an event unavailable in variant.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

Dem/DemGeneral/DemAvailabilitySupport == DEM_EVENT_AVAILABILITY AND Dem/DemConfigSet/DemEventParameter/DemEventAvailableInVariant == FALSE for the event the API Dem_SetEventAvailable is called for.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097911 Deferred Rx PDUs are not received if ComDeferredEventCacheSupport is enabled	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	8.01.00
Fixed in versions:	14.00.01, 13.03.03, 12.00.02
Problem Description:	
What happens (symptoms): ----- Deferred Rx PDUs might not be processed if deferred event cache is enabled. This situation will sustain until the deferred event cache overflows. When does this happen: ----- This issue may occur at runtime, whenever more deferred PDUs are received than the cache can store. In this case the fallback strategy will be applied. If during processing of the deferred PDU's new deferred PDUs are received, they might not be cached properly. In which configuration does this happen: ----- ComDeferredEventCacheSupport == TRUE AND type of processed Rx PDU is set to DEFERRED	
Resolution Description:	
Workaround: ----- If possible, set ComDeferredEventCacheSupport == FALSE. Otherwise no workaround is available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097951 Wrong marshallng/ unmarshalling of <XInt64> Signals	
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	11.00.00
Fixed in versions:	12.00.03, 13.03.05, 14.01.02
Problem Description: What happens (symptoms): ----- Signals/ GroupSignals with ComSignalType <XInt64> are read / written incorrectly from/ to the bus. A Rx Signal for example could be truncated when it's read by Com_ReceiveSignal. When does this happen: ----- Issue occurs at runtime In which configuration does this happen: ----- ComSignalType == UINT64 SINT64 AND ComBitPosition starts at byte boundary AND 32 Bits < ComBitSize < 64 Bits AND Signal does not end at byte boundary	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098006 Declined Immediate Nm Transmissions is retried later than expected	
Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	1.00.00
Fixed in versions:	7.00.00
Problem Description: What happens (symptoms): ----- An Nm message is transmitted later than expected. The transmission is retried after CanNmImmediateNmCycleTime but should be retried after CanNmMainFunctionPeriod. /MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmImmediateNmCycleTime /MICROSAR/CanNm/CanNmGlobalConfig/CanNmMainFunctionPeriod When does this happen: ----- After active startup of the ECU and if one of the 'Immediate Nm Transmissions' is declined. In which configuration does this happen: ----- - /MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/ CanNmImmediateNmTransmissions > 0 on the affected channel AND - /MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmImmediateNmCycleTime > /MICROSAR/CanNm/CanNmGlobalConfig/CanNmMainFunctionPeriod	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098203 Service 0x3E: S3 timer reloaded for different protocol	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.05.00
Fixed in versions:	9.05.00, 8.06.03
Problem Description: What happens (symptoms): ----- Although a different protocol is active a functional addressed 0x3E 0x80 request (with set SPRMIB) reloads the S3 timer during any non-default session. When does this happen: ----- Any time a functional addressed 0x3E 0x80 request is sent with a protocol priority higher than the active one. In which configuration does this happen: ----- - DCM shall prioritize two or more diagnostic clients (e.g. OBD and UDS clients) (in Dcm_Cfg.h #define DCM_NET_PROTOCOL_PRIORITISATION_ENABLED == STD_ON) AND - The client specific session protection is activated (in Dcm_Cfg.h: #define DCM_NET_PROTECT_SESSION_OF_CLIENT_ENABLED == STD_ON)	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098248 Immediate priority block requests during WriteAll may lead to unsuccessful WriteAll**Component@Subcomponent:** MemService_AsrNvM@Implementation**First affected version:** 5.00.00**Fixed in versions:** 5.07.01**Problem Description:**

What happens (symptoms):

NvM WriteAll does not finish successfully through NvM accepts WriteBlock for immediate priority blocks (return E_OK). Some blocks may be not ok after the WriteAll. During the following ReadAll integrity failed error may occur.

When does this happen:

If some immediate priority WriteBlock jobs are requested during an ongoing WriteAll.

In which configuration does this happen:

Any configuration with immediate priority NvM blocks (NvMBlockJobPriority == 0) and disabled parameter checks NvMDevErrorDetect == false && (NvMSafeBswChecks == false || EcuCSafeBswChecks == false) (depends on NvM version) and NvMKillWriteAllApi == true

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098392 S3 timer reloaded by any request from different tester	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.00.00
Fixed in versions:	9.05.00, 8.06.03
Problem Description:	
What happens (symptoms):	

During non-default session a request from a different tester reloads the S3 timer.	
When does this happen:	

Any time when during non-default session a request from a different tester with lower or equal priority is send.	
In which configuration does this happen:	

- Pseudo parallel request handling is activated (in Dcm_Cfg.h: #define DCM_NET_MULTI_CLIENT_ENABLED == STD_ON)	
AND	
- The client specific session protection is activated (in Dcm_Cfg.h: #define DCM_NET_PROTECT_SESSION_OF_CLIENT_ENABLED == STD_ON)	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098604 Service 0x3E: Keep-Alive timer reloaded for different protocol	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	4.01.00
Fixed in versions:	9.05.00, 8.06.03
Problem Description: What happens (symptoms): ----- Although a different protocol is active a functional addressed 0x3E 0x80 request (with set SPRMIB) reloads the Keep-Alive timer during any non-default session, which prevents ECU from going asleep. When does this happen: ----- Any time a functional addressed 0x3E 0x80 request is sent with a protocol priority higher than the active one. In which configuration does this happen: ----- - Keep-Alive timer is supported (in Dcm_Cfg.h #define DCM_NET_KEEP_ALIVE_TIME_ENABLED == STD_ON) AND - DCM shall prioritize two or more diagnostic clients (e.g. OBD and UDS clients) (in Dcm_Cfg.h #define DCM_NET_PROTOCOL_PRIORITISATION_ENABLED == STD_ON) AND - The client specific session protection is activated (in Dcm_Cfg.h: #define DCM_NET_PROTECT_SESSION_OF_CLIENT_ENABLED == STD_ON)	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098606 Keep-Alive timer reloaded by any request from different tester	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	4.01.00
Fixed in versions:	9.05.00, 8.06.03
Problem Description: What happens (symptoms): ----- During non-default session a request from a different tester reloads the Keep-Alive timer, which prevents ECU from going asleep. When does this happen: ----- Any time when during non-default session a request from a different tester with lower or equal priority is send. In which configuration does this happen: ----- - Keep-Alive timer is supported (in Dcm_Cfg.h #define DCM_NET_KEEP_ALIVE_TIME_ENABLED == STD_ON) AND - Pseudo parallel request handling is activated (in Dcm_Cfg.h: #define DCM_NET_MULTI_CLIENT_ENABLED == STD_ON) AND - The client specific session protection is activated (in Dcm_Cfg.h: #define DCM_NET_PROTECT_SESSION_OF_CLIENT_ENABLED == STD_ON)	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098938 Unconfirmed DTCs need healing before aging	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	9.00.00
Fixed in versions:	14.01.00, 12.01.05
Problem Description: What happens (symptoms): ----- A stored DTC needs healing before aging, although neither its ConfirmedDTC nor WIR status bit is set. When does this happen: ----- Always. In which configuration does this happen: ----- Dem/DemGeneral/DemAgingAfterHealing == DEM_AGING_AFTER_HEALING_ALL_DTC AND Dem/DemConfigSet/DemEventParameter/DemEventClass/DemIndicatorAttribute/ DemIndicatorHealingCycleCounterThreshold > 0	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00099092 CONNECT will stop a running DAQ measurement from resume mode	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	3.00.01
Problem Description: What happens (symptoms): ----- When the XCP Slave receives a CONNECT command, a running DAQ measurement (e.g. from resume mode) is stopped and has to be started again, manually. The ASAM spec states that a DAQ measurement started by resume mode shall not be interrupted by the CONNECT command. The implementation deviates from this ASAM requirement. When does this happen: ----- Always and immediately In which configuration does this happen: ----- When resume mode is used. (/MICROSAR/Xcp/XcpCmdConfig/XcpDaqAndStim/XcpResumeMode enabled and resume mode activated by CANape)	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00099171 Value of 'Cycles Since First Failed' is 0

Component@Subcomponent: Diag_Asr4Dem@Implementation
First affected version: 5.00.00
Fixed in versions: 15.01.00, 12.01.05

Problem Description:

What happens (symptoms):

The Dem returns 0 for 'Cycles Since First Failed' for a DTC although the operation cycle was restarted at least once since the DTC was tested 'failed' for the first time.

When does this happen:

1. A DTC is tested 'failed' (TestFailed status bit is set), but is not stored in event memory (for example because the event memory is already full).
2. Later on the DTC is stored in the event memory (for example because another DTC is cleared from event memory).
3. The DTC is tested 'failed' again (TestFailed status bit is still set).
4. The operation cycle is at least restarted once.
5. The internal data element 'Cycles Since First Failed' is requested via extended data or snapshot records (by DCM service 19x04 or 19x06 or application APIs).

As soon as the TestFailed status bit is reset (by testing DTC 'passed') and the DTC is tested 'failed' again, the 'Cycles Since First Failed' will be calculated correctly from this failed result on (assuming that the DTC is still stored in the event memory).

In which configuration does this happen:

The event/DTC has an extended data record (Dem/DemConfigSet/DemEventParameter/DemExtendedDataClassRef) or snapshot record (Dem/DemConfigSet/DemEventParameter/DemFreezeFrameClassRef) configured including data element 'CyclesSinceFirstFailed' (Dem/DemGeneral/DemDataClass/DemDataElementInternalData == CYCLES_SINCE_FIRST_FAILED) AND
Dem/DemGeneral/DemEventMemoryEntryStorageTrigger ==
DEM_STORAGE_ON_FDC_THRESHOLD

From Dem version 8.00.00 to 12.xx.xx the ESCAN only holds for events/DTCs

- a) with Dem/DemConfigSet/DemEventParameter/DemEventKind == DEM_EVENT_KIND_BSW
OR
- b) which are part of a combined group (at least one other event references the same DTC by Dem/DemConfigSet/DemEventParameter/DemDTCClassRef)
OR
- c) using time based de-bouncing (Dem/DemConfigSet/DemEventParameter/DemEventClass/DemDebounceAlgorithmClass/DemDebounceTimeBase)

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099197 Wrong DLC calculation for zero bit signals

Component@Subcomponent: Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version: 3.00.00
Fixed in versions: 15.00.00

Problem Description:

What happens (symptoms):

On reception-side, a zero bit signal might get rejected on partially received PDU.
 A configured signal notification might not be called.

When does this happen:

Issue occurs at runtime.

In which configuration does this happen:

Configuration A:

- ComSignalEndianness == LITTLE_ENDIAN
 AND
 - ComBitSize is 0
 AND
 (
 - ComBitPosition < 8
 OR
 - (ComBitPosition % 8) != 0, where ComBitPosition > 8
)

Configuration B:

- ComSignalEndianness == BIG_ENDIAN
 AND
 - ComBitSize is 0
 AND
 - (ComBitPosition % 8) != 7

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099383 Failure to collect data via Client/Server or Sender/Receiver R-Port results in random data	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	10.00.00
Fixed in versions:	12.01.05, 15.03.00
Problem Description: What happens (symptoms): ----- The data content of DataElement(s) in ExtendedDataRecords, PIDs, DIDs and thereof in OBDII/WWHOBD - FreezeFrames, SnapshotRecords deviates from the expected value, if the DataElement content is collected via RTE over a Client/Server or Sender/Receiver port interface. When does this happen: ----- When an OBDII - WWHOBD Freeze Frame, Snapshot Record, Extended Data Record is stored/updated which triggers collecting data corresponding to configured Data Elements through the RTE from the Application. Issue happens always when the RTE call to collect the data is not successful, whereupon the return code is not E_OK (0). According AUTOSAR, this RTE call MUST ALWAYS be successful - so in theory this issue shall NEVER occur, if the configuration is correct. In practice, a wrong RTE configuration may violate this constraint - check RTE specification for problematic configurations. In which configuration does this happen: ----- The DataElement in consideration here is configured to read data from application through a Client/Server port: /Dem/DemGeneral/DemDataClass/DemDataElementUsePort == USE_DATA_CLIENT_SERVER_PORT or /Dem/DemGeneral/DemDataClass/DemDataElementUsePort == USE_DATA_CLIENT_SERVER_PORT_WITH_EVENTID OR The DataElement in consideration here is configured to read data from application through a Sender/Receiver port: /Dem/DemGeneral/DemDataClass/DemDataElementUsePort == USE_DATA_SENDER_RECEIVER AND The incorrect configuration of the RTE may result in un-successful data collection, and the RTE call returns an error value different from "E_OK" (0) and "E_NOT_OK" (1). Examples for incorrect RTE configurations are : - DEM R-Port is not connected in the RTE - connection (e.g. to another ECU) can have timeouts, failures... - failures with buffer transformations - buffer overflow at the server side - inaccessible server - limits in simultaneous server access - ... (this list is incomplete)	
Resolution Description:	

ESCAN00099383 Failure to collect data via Client/Server or Sender/Receiver R-Port results in random data

Workaround:

As required by AUTOSAR, use only Client/Server- or Sender/Receiver-port calls, that can never fail and will ALWAYS return the requested data and return value E_OK (0).
Particularly, always connect DEM's R-Ports! The MICROSAR RTE generates template code for unconnected ports - which doesn't comply above requirement.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099495 Measurement failure when more than 255 measurement values are configured and used

Component@Subcomponent: Cp_Asr4Xcp@Implementation

First affected version: 1.00.00

Fixed in versions: 3.00.01

Problem Description:

What happens (symptoms):

When the Master Tool configures and tries to measure more than 255 measurement values, the values with an index above 255 are not measured. The XCP frames are not assembled correctly and ODT frames are incomplete.

When does this happen:

Always and immediately

In which configuration does this happen:

When DAQ measurement is used
(Presence of: /MICROSAR/Xcp/XcpCmdConfig/XcpDaqAndStim)

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099586 ErrorFrame after startup with wakeup validation

Component@Subcomponent: Ccl_Asr4SmCan@Implementation
First affected version: 2.06.00
Fixed in versions: 3.00.00

Problem Description:

What happens (symptoms):

 ErrorFrame after startup with wakeup validation.

When does this happen:

 After a successful wakeup validation and if only two ECUs are at the bus.
 In most cases this is a test setup, in which only the ECU and a tester, e.g. CANoe, are connected to the CAN bus.

1. WakeupValidation is started
 2. A valid CAN Msg is received ==> wakeup is validated
 3. The CAN controller is restarted.
- If right now the other ECU sends the next CAN Msg, the ECU doesn't get a acknowledgement

In which configuration does this happen:

 If wakeup validation (supported by CanSM) is used (i.e. CanSM_StartWakeupSources() is used in EcuM_StartWakeupSources() resp. the StopWakeupSources).

Resolution Description:

Workaround:

 No workaround available.

Resolution:

 The described issue is corrected by modification of all affected work-products.

ESCAN00099742 Status block is not immediately written to NvRAM after a clear request

Component@Subcomponent: Diag_Asr4Dem@Implementation
First affected version: 7.00.00
Fixed in versions: 13.05.00, 12.01.05

Problem Description:

What happens (symptoms):

1. The Status block is not immediately written to NvRAM after a clear request.
2. Depending on the configuration ClearDTC will return a final result although the Status block was not persisted in NvRAM.

When does this happen:

This situation can happen in different scenarios wherein the status block is updated after a DTC Clear Operation within the same DEM main function call.

An example situation would be the following:

1. A clear request is triggered (->Status block is reset and marked to be persisted immediately on the following main function).
2. The status block content changes (e.g. due to a reported monitor result), before the Dem can trigger NvRAM storage.

In this case the Dem will request the writing of the blocks to NvRAM only again after a new clear request, a call of Dem_RequestNvSynchronization or at the latest on Dem_Shutdown.

In which configuration does this happen:

```
Dem/DemGeneral/DemUseNvm == TRUE
AND
(
If NVM immediate updates are supported, i.e.
Dem/DemGeneral/DemClearDTCBehavior == DEM_CLRRESP_NONVOLATILE_FINISH
OR
Dem/DemGeneral/DemImmediateNvStorageSupport == TRUE OR any Dem/DemConfigSet/
DemDTCClass/DemImmediateNvStorage == TRUE
OR
Dem/DemGeneral/DemNvSynchronizeSupport == TRUE
)
```

Symptom 2 is only relevant in configurations with Dem/DemGeneral/DemClearDTCBehavior == DEM_CLRRESP_NONVOLATILE_FINISH.

Resolution Description:

Workaround:

A workaround is to manually invoke the Dem_RequestNvSynchronization(), if available, which would write back all the modified NV Blocks (including for example de-bounce or available NvRAM block which wouldn't be written immediately after a clear request) to the backing storage. Else the Status block would be written to NvM either at the next ClearDTC request or at shutdown.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099810 Event heals too early after displacement	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	4.00.00
Fixed in versions:	14.03.00, 12.01.05
Problem Description:	
What happens (symptoms):	

An event may 'heal' too early and UDS Wir-Bit is reset too early.	
When does this happen:	

After an event is displaced, the event is reported failed but the event's UDS pending status bit can not be set (e.g. event's storage conditions not fulfilled or the event fails to acquire a memory slot and configuration is /MICROSAR/Dem/DemGeneral/DemPendingDtcProcessing == DEM_PROCESS_PDTC_STOREDONLY).	
In which configuration does this happen:	

/DemGeneral/DemEventDisplacementStrategy != DEM_DISPLACEMENT_NONE	
AND	
Event has an indicator attached (/DemConfigSet/DemEventParameter/DemEventClass/DemIndicatorAttribute/DemIndicatorRef)	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099905 XCP internal memory not initialized correctly in huge configurations.	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- In configurations where many Odt Entries are configured >~10k, the resulting memory structure of the XCP is >64k. In this case the Xcp_Hlp_MemSet which has only an uint16 as length does not initialize the whole memory area. As a result the XCP component might have an undefined behaviour if such a configuration is used (e.g. an AllocDaq might not work and measurement is not possible). When does this happen: ----- Always and immediately In which configuration does this happen: ----- When a huge amount of Odt Entries is used (>~10k)	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

2.3 Runtime Issues with Workaround

It is not recommended to update a delivery due to a runtime issue with a documented workaround. The effect of an issue to the ECU functionality has to be analyzed by the user as the software usage and its configuration is not known by Vector. The risk of change has also to be taken into account.

Index

ESCAN00061207	DaVinci Configurator 5: Issue Reporting Procedure GenTool_ConfiguratorCfg5@Application
ESCAN00076256	BswM_CanSM_Indication called with locked interrupts - OS ErrorHook on Os API Invocation Ccl_Asr4SmCan@Implementation
ESCAN00089164	The EcuM stays in RUN state even if EcuM_KillAllRunRequests has been called SysService_Asr4EcuM@Implementation
ESCAN00091305	EcuM with fixed state machine causes a Det error in Dem_Init because this module has been initialized two times SysService_Asr4EcuM@Implementation
ESCAN00091550	Service 0x27: Dcm allows seed/key attempt earlier than the configured security delay time Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00093906	ECU returns NRC 0x13 instead of 0x33 or other execution pre-condition related NRC for services with a sub-function parameter Diag_Asr4Dcm@Implementation
ESCAN00094013	Missing callback configuration is not detected If_AsrIfFeeSmallSector@Implementation
ESCAN00094333	Timeout Action Replace doesn't work for Rx SignalGroups with Array Access enabled Il_AsrComCfg5@Implementation
ESCAN00094451	Supervision of the STmin by the application fails Tp_Asr4TpCan@Implementation
ESCAN00094464	NvM data is not stored after calling ComM_DeInit() Ccl_Asr4ComMCfg5@Implementation
ESCAN00095016	Service 0x23 responds with NRC 0x14 instead of 0x31 Diag_Asr4Dcm@Implementation
ESCAN00095254	Missing DET error PDUR_E_PDU_INSTANCES_LOST description in case of N:1 communication interface routings with upper layer Gw_AsrPduRCfg5@Doc_TechRef
ESCAN00095441	[Only in case of multiple CAN driver are used] FIFO behavior is not fulfilled for transmission of CAN messages If_AsrIfCan@GenTool_GeneratorMsr
ESCAN00095651	Wrong software check for correct AuthID DrvCry_Rh850Icum@Implementation
ESCAN00095653	Primitive returns with error with valid input parameters (using keyID on 2nd keypage) DrvCry_Rh850Icum@Implementation
ESCAN00095919	Services 0x23/0x3D/0x2C: Unexpected response behavior on invalid memory Id request Diag_Asr4Dcm@Implementation
ESCAN00095963	Service 0x31: Erroneous implementation of request minimum length check Diag_Asr4Dcm@Implementation
ESCAN00096099	Main function tick time resolution smaller than 1 ms is not supported Tp_Asr4TpCan@GenTool_GeneratorMsr
ESCAN00096195	Rte_LdComCbkTriqgerTransmit returns invalid error code Rte_Core@Implementation

Index

ESCAN00096248	Validator PDUR12501 is not shown as error for PduRDestPduRef Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00096369	CAN driver stops sending message. DrvCan__coreAsr@Implementation
ESCAN00096716	Queued sender/receiver N:1 connections not detected Rte_Analyzer@Application
ESCAN00096901	Possible incorrect interpretation of last page status If_AsrIfFeeSmallSector@Implementation
ESCAN00096926	a2I: Calculation of communication parameter does not consider PropSeg parameter Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00097045	Dem_SetEventAvailable returns E_OK for events unavailable in variant Diag_Asr4Dem@Implementation
ESCAN00097052	Data send completed trigger fired to early when Rte_ComSendSignalProxyPeriodic is used Rte_Core@Implementation
ESCAN00097141	The Xcp Module ID is not according to AR 4 Cp_Asr4Xcp@Implementation
ESCAN00097176	When the XCP Master requests Timestamps but none are configured no negative response is returned Cp_Asr4Xcp@Implementation
ESCAN00097264	Service 0x2C: Valid DDDID definition requests always responded with NRC 0x31 Diag_Asr4Dcm@Description
ESCAN00097288	When fixed timestamps are configured but the XCP Master does not request them, no negative response is returned Cp_Asr4Xcp@Implementation
ESCAN00097333	ComM_BusSM_ModeIndication called with locked interrupts - OS ErrorHook on Os API Invocation Ccl_Asr4SmCan@Implementation
ESCAN00097361	Service 0x2A: Wrong prioritisation between NRC 0x13 and 0x31 Diag_Asr4Dcm@Implementation
ESCAN00097377	Communication not possible in Multi Channel use case on channels > 0 Cp_Asr4Xcp@Implementation
ESCAN00097381	Service 0x27: PowerOn delay time not started on single false access attempt Diag_Asr4Dcm@Doc_TechRef
ESCAN00097520	Callout Dcm_FilterDidLookUpResult() is called with unexpected OpStatus Diag_Asr4Dcm@Implementation
ESCAN00097602	CAN interrupt lost DrvCan_Rh850McanAsr@Implementation
ESCAN00097637	EcuM calls the Mcu_SetMode API for setting the normal mcu mode with an invalid mcu mode SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00097731	Service 0x10: Jump to FBL performed although forced RCR-RP could not be sent Diag_Asr4Dcm@Implementation
ESCAN00097925	Memory read access exception due to incorrect address conversion Cp_Asr4Xcp@Implementation
ESCAN00098007	Declined Immediate Nm Transmissions is retried later than expected Nm_Asr4NmCan@Doc_TechRef
ESCAN00098036	Cyclic PDUs with a ComGwRoutingTimeout are triggered when started if ComTxDI MonTimeBase is configured Il_AsrComCfg5@Implementation

Index

ESCAN00098095	PduInfoPtr of API Appl_GenericConfirmation() contain DLC instead of message length DrvCan__coreAsr@Implementation
ESCAN00098361	Service 0x2F: Wrong ReturnControlToECU operations are called on session/ security state change Diag_Asr4Dcm@Implementation
ESCAN00099078	RTE generator does not trigger NvM_MainFunction when a NvBlock SWC has no NvBlockDescriptors Rte_Core@Implementation
ESCAN00099239	Enabled event memory that has no events assigned leads to wrong code generation Diag_Asr4Dem@GenTool_GeneratorMsr
ESCAN00099440	RTE sends unexpected data when activation reasons are configured for a runnable with implicit write accesses Rte_Core@Implementation
ESCAN00099458	Channel restarts when all Partial Networks are released and channel is supposed to shut down Nm_Asr4NmCan@GenTool_GeneratorMsr
ESCAN00099480	Dirty flags not handled when multiple implicit accesses for the same NvBlock descriptor are configured and implicit write is skipped Rte_Core@Implementation
ESCAN00099536	Safety Manual does not fit to current description file / generator output If_Asr4IfWd@root
ESCAN00099665	Xcp_Event throws DET check if called before Xcp_Init Cp_Asr4Xcp@Implementation
ESCAN00100047	Consecutive failed cycle counter not persisted to NvRAM Diag_Asr4Dem@Implementation
ESCAN00100243	incorrect ODT messages assembled if ODT Entries are initialized incompletely Cp_Asr4Xcp@Implementation

ESCAN00061207 DaVinci Configurator 5: Issue Reporting Procedure	
Component@Subcomponent:	GenTool_ConfiguratorCfg5@Application
First affected version:	5.00.01
Fixed in versions:	
Problem Description: This ticket describes the reporting of DaVinci Configurator Pro issues. This ticket is a general information and not an issue. ----- Issues of the DaVinci Configurator Pro tool are not part of the active issue reporting (i.e. this report). The DaVinci Configurator Pro issue list can be downloaded from our home page: DaVinci Developer OpenIssue Lists https://portal.vector.com/web/davinci/shared-folder?t=c2b431ff-5dae-4a72-83ec-b9c8ca17561c DaVinci Configurator Pro OpenIssue Lists https://portal.vector.com/web/davinci/shared-folder?t=15d156f3-65d3-4b6e-8107-ec44051aebff	
Resolution Description: Workaround: ----- This is not an issue but only a reference to the tool specific issue reporting. No changes to the delivery required.	

ESCAN00076256 BswM_CanSM_Indication called with locked interrupts - OS ErrorHook on Os API Invocation

Component@Subcomponent: Ccl_Asr4SmCan@Implementation

First affected version: 2.00.00

Fixed in versions: 2.13.01

Problem Description:

What happens (symptoms):

Symptom 1)

OS API Invocation ends up with a call to ErrorHook.

Error Cause of this ErrorHook is that OS API is called whereas Interrupts are locked.

Errorcodes would then be:

osdErrATIntAPIDisabled 0x1104U,
osdErrHTMultipleActivation 0x1305U,
osdErrSEIntAPIDisabled 0x4105U,
osdErrSECallContext 0x4106U or
osdErrWEInterruptsDisabled 0x4403U

Symptom 2)

Rte Runnable that is triggered on TxError will not be called when Communication is shut down and Sending Request is outstanding

When does this happen:

every time function "BswM_CanSM_Indication" ends up in an Os API Invocation (for example BswM Rule to stop Ipdu Groups AND a transmission is still ongoing AND TxError is configured to trigger a runnable in the Rte)

AND

the function "BswM_CanSM_Indication" is called with locked interrupts (Transition to NO_COMMUNICATION)

In which configuration does this happen:

If any BswM rule (which belongs to a BswM_CanSM_Indication), is configured as immediate

AND

an OS function is called within the call context of the BswM rule

AND

this OS function must not be called with locked interrupts.

e.g.

Std Rules in BswM are configured

AND

A Rte Runnable is triggered on Tx Error

AND

Osek OS which implements the requirement "If ActivateTask / SetEvent is called with locked Interrupts, reject the action and call an ErrorHook" is used (for example Vector Os)

Resolution Description:

ESCAN00076256 BswM_CanSM_Indication called with locked interrupts - OS ErrorHook on Os API Invocation

Workaround:

Configure the BswM port of the CanSM indication as deferred.

If the startup is not affected and a fast startup is desired it is possible to "copy" the port and set one as deferred (e.g. used for the shut down rules) and the other port as immediate (e.g. used for the startup rules).

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00089164 The EcuM stays in RUN state even if EcuM_KillAllRunRequests has been called	
Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	3.00.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- The ECU stays in RUN state, even if anyone has called the API EcuM_KillAllRunRequests. When does this happen: ----- Always after EcuM_KillAllRunRequests() has been called and at least one channel is in a state unequal COMM_NO_COM_NO_PENDING_REQUEST. In which configuration does this happen: ----- Only in configurations with ECUM_FIXED_BEHAVIOR is active (EcuM/EcuMGeneral/EcuMEnableFixBehavior).	
Resolution Description:	

ESCAN00089164 The EcuM stays in RUN state even if EcuM_KillAllRunRequests has been called

Workaround:

The following shows a possible workaround to ignore all ComM channel states in case of an active KillAllRunRequests.

Hint: EcuM_SetWakeupEvent considers wakeup events even if EcuM_KillAllRunRequests() was called. This might cause that the EcuM transits from PostRun to Run again, because of a new occurred wakeup event.

The call of the API ComM_GetState() has to be mapped to an application function in case that it is called in context of EcuM.c. This can be done by configure the following header file as a User Configuration file in the EcuM configuration (EcuM/EcuMGeneral/EcuMUserConfigurationFile):

- Example Appl_ComM_EcuM.h:

```
#if defined (ECUM_SOURCE)
extern Std_ReturnType Appl_ComM_GetState(const NetworkHandleType Channel,
ComM_StateType* State);
```

```
# define ComM_GetState Appl_ComM_GetState
#endif
```

- Example Appl_ComM_EcuM.c:

```
#include "Std_Types.h"
#include "ComM.h"
```

```
#define ECUM_PRIVATE_CFG_INCLUDE
#include "EcuM_PrivateCfg.h"
#undef ECUM_PRIVATE_CFG_INCLUDE
```

```
Std_ReturnType Appl_ComM_GetState(const NetworkHandleType Channel, ComM_StateType*
State)
{
    Std_ReturnType retVal = E_OK;
    /* Verify that EcuM_KillAllRunRequests() was not called */
    if ((EcuM_GetKillAllInProgress() & (0x01u)) == 0u)
    {
        retVal = ComM_GetState(Channel, State);
    }
    else
    {
        /* In case of an active KillAllRunRequest, set the virtual ComM State to no communication and no
        request. */
        *State = COMM_NO_COM_NO_PENDING_REQUEST;
    }

    return retVal;
}
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00091305 EcuM with fixed state machine causes a Det error in Dem_Init because this module has been initialized two times

Component@Subcomponent: SysService_Asr4EcuM@Implementation

First affected version: 3.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

In some situations the EcuM with fixed state machine calls Dem_Init() two times, this lead to a Det error thrown by the Dem with the message DEM_E_WRONG_CONDITION,

When does this happen:

During runtime of the EcuM API EcuM_StartupTwo().

In which configuration does this happen:

All of the following conditions have to be fulfilled to be affected by this issue:

- The Ecum with fixed state machine has to be active (EcuM/EcuMGeneral/EcuMEnableFixBehavior).
- The include Dem has to be active (EcuM/EcuMFixedGeneral/EcuMIncludeDem).
- At least one wakeup source has to be configured for wakeup validation (EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMWakeupSource/EcuMValidationTimeout).
- At startup the standard wakeup source (ECUM_WKSOURCE_RESET) has to be cleared via the API EcuM_ClearWakeupEvent() to force a wakeup validation after startup and to prevent a transition to RUN state until this wakeup source is validated.

Resolution Description:

Workaround:

In case that the valid wakeup event during initialization (ECUM_WKSOURCE_RESET) is cleared in context of driver init list two or three and a wakeup event for validation is set the Dem_Init call has to be avoided.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00091550 Service 0x27: Dcm allows seed/key attempt earlier than the configured security delay time

Component@Subcomponent: Diag_Asr4Dcm@GenTool_GeneratorMsr

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

A security delay timer expires too early. Dcm accepts new seed requests before the configured delay time is expired.

When does this happen:

If after last unsuccessful attempt responded with Nrc 0x36 (exceededNumberOfAttempts) a new seed request is sent before the expected delay time.

In which configuration does this happen:

- Service 0x27 is supported

AND

- There is more than one security level configured

AND

- Any delay time is configured for any security level (in Dcm_Cfg.h:

DCM_STATE_SEC_RETRY_ENABLED == STD_ON or

DCM_STATE_SEC_DELAY_ON_BOOT_ENABLED == STD_ON)

AND

- The dividend of a configured security delay time (in milliseconds) and the task cycle (also in milliseconds) is greater than 65535

Resolution Description:

Workaround:

The equation shall become true: ($\text{<TimeParameter>} / \text{DcmTaskTime}$) < 65535.

Therefore, the following workarounds are possible:

1) Increase the DcmTaskTime parameter value.

OR

2) Reduce the timeout value in the corresponding timing parameter.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00093906	ECU returns NRC 0x13 instead of 0x33 or other execution pre-condition related NRC for services with a sub-function parameter
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.00.00
Fixed in versions:	8.04.00
Problem Description:	
What happens (symptoms):	
----- ECU returns NRC 0x13 instead of 0x33 or other execution pre-condition related NRC for services with a sub-function parameter.	
When does this happen:	

Each time a diagnostic client requests a service that:	
- supports sub-function parameter (e.g. SID 0x10 (DiagnosticSessionControl))	
AND	
- The requested sub-function is a valid one, but:	
- not supported in the currently active security level	
OR	
- not allowed under currently active ECU project specific sub-function related pre-conditions (i.e. modeled via DcmModeRules)	
AND	
- the request length is not valid for the sub-function	
In which configuration does this happen:	

For any diagnostic service that has sub-functions supported in different security levels resp. different DcmModeRule pre-conditions.	
Example:	
- ECU is configured to support the following security levels: locked, level X.	
- Service 0x10 has following sub-functions (SF): 0x01 and 0x02.	
This issue will occur if:	
- SF 0x01 is allowed in all security levels (locked one inclusively) i.e. has no security restrictions	
- SF 0x02 is allowed only once security access level X is enabled	
Any request for service 0x10 0x02 with any wrong length more than two bytes, while the ECU is locked, will result in NRC 0x13 (ICMLOF) instead of 0x33 (SAD).	
Please note:	
This issue will <u>not</u> occur if:	
- All SID 0x10 sub-functions have the same diagnostic security access execution precondition dependency e.g. allowed only in the security level X. In this case not only the sub-functions but the SID 0x10 itself will not be supported in the security access locked level. This will result in NRC 0x33 (SAD), which is no deviation of the ISO specification.	
Resolution Description:	

ESCAN00093906 ECU returns NRC 0x13 instead of 0x33 or other execution pre-condition related NRC for services with a sub-function parameter

Workaround:

 Avoid sub-function related security level resp. DcmModeRule execution precondition dependencies. Let the complete service (at SID level) be restricted via security level resp. DcmModeRule execution pre-condition.

Resolution:

 The described issue is corrected by modification of all affected work-products.

ESCAN00094013 Missing callback configuration is not detected

Component@Subcomponent: If_AsrIfFeeSmallSector@Implementation

First affected version: 1.00.00

Fixed in versions: 2.00.00

Problem Description:

What happens (symptoms):

 Fee in callback mode contains a global parameter with the last job result of FLS. This value is only updated upon invocation of FEE's JobEnd-/JobErrorNotification at the end of any FLS job. If FEE's callback functions aren't configured in FLS this parameter will never be updated. Consequently FEE always uses the initial value of this parameter (MEMIF_JOB_OK), which will lead to incorrect job processing.

FEE's callback functions shall be configured in FLS in following parameters:

Fls/FlsConfigSet/FlsJobEndNotification

Fls/FlsConfigSet/FlsJobErrorNotification

When does this happen:

 During job processing.

In which configuration does this happen:

 FEE is used with callback notifications _and_ callback functions aren't configured in FLS (Fls/FlsConfigSet/FlsJobEndNotification and Fls/FlsConfigSet/FlsJobErrorNotification).

Resolution Description:

Workaround:

 Configure FEE's callback functions:

Fls/FlsConfigSet/FlsJobEndNotification = Fee_30_SmallSector_JobEndNotification

Fls/FlsConfigSet/FlsJobErrorNotification = Fee_30_SmallSector_JobErrorNotification

Resolution:

 The described issue is corrected by modification of all affected work-products.

ESCAN00094333 Timeout Action Replace doesn't work for Rx SignalGroups with Array Access enabled**Component@Subcomponent:** Il_AsrComCfg5@Implementation**First affected version:** 7.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

The timeout Replace action for rx SignalGroups with array access enabled does not replace the buffer value either with the initial value or the configured Rx Data Timeout Substitution Value.

When does this happen:

During call of Com_MainFunctionRx()

In which configuration does this happen:

In all configurations with rx SignalGroups which have a configured timeout > 0, timeout action set to Replace and array access is enabled.

Resolution Description:

Workaround:

Disable Array Access for signalGroups if applicable (no use of com-based transformer)

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094451 Supervision of the STmin by the application fails	
Component@Subcomponent:	Tp_Asr4TpCan@Implementation
First affected version:	2.01.01
Fixed in versions:	3.03.02
Problem Description:	
What happens (symptoms):	

The separation time (STmin) between consecutive frames is not correct, when the time supervision is done by the application instead of the CanTp (the application calls the function CanTp_StopSeparationTime to trigger the transmission of the next consecutive frame).	
When does this happen:	

Everytime the application is in charge of the STmin supervision	
In which configuration does this happen:	

1. The parameter CanTp/CanTpGeneral/CanTpApplSTminStartFunction is not empty. AND 2. The macro CANTP_CONFIGURATION_VARIANT == CANTP_CONFIGURATION_VARIANT_POSTBUILD_LOADABLE (in CanTp_Cfg.h) OR The macro CANTP_POSTBUILD_VARIANT_SUPPORT == STD_ON (in CanTp_Cfg.h)	
Resolution Description:	

ESCAN00094451 Supervision of the STmin by the application fails

Workaround:

The CanTp calls the function configured in CanTp/CanTpGeneral/CanTpApplSTminStartFunction to indicate that the separation time for the next CF has to be started.

In the configurations already mentioned and with the current implementation, the Tx NSDU id used by the CanTp when calling this function corresponds to an internal id, which could be different to the one in CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxNSduId, and therefore also different to the value assigned to the symbolic name of the Tx NSDU.

The mapping between the symbolic name of the Tx NSDU and the internal id is constant and can be found in the array CanTp_TxSduSrv2Hdl. The array can be found in CanTp_Lcfg.c or CanTp_PBcfg.c.

For example, if the CanTp_TxSduSrv2Hdl array looks like this:

```
static CONST(CanTp_TxSduSrv2HdlType, CANTP_PBCFG) CanTp_TxSduSrv2Hdl[7] = {
/* Index TxSduCfgIdx Comment */
{ /* 0 */ CANTP_NO_TXSDUCFGIDXOFTXSDUSRV2HDL }, /* [Unused] */
{ /* 1 */ 0U }, /* [TxNSdu_PE2] */ /* 1 != 0 */
{ /* 2 */ 1U }, /* [TxNSdu_PM1] */
{ /* 3 */ 2U }, /* [TxNSdu_PS1] */
{ /* 4 */ 3U }, /* [TxNSdu_PE3] */
{ /* 5 */ CANTP_NO_TXSDUCFGIDXOFTXSDUSRV2HDL }, /* [Unused] */
{ /* 6 */ 4U } /* [TxNSdu_FS1] */
};
```

the CanTp will use the internal id 2 when calling the function to indicate the start of the separation time for the Tx NSDU with CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxNSduId = 3. That means the application has to react to id 2 instead of id 3 when the function configured in CanTp/CanTpGeneral/CanTpApplSTminStartFunction is called.

On the other hand, the function CanTp_StopSeparationTime can be called using the id in CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxNSduId or using the symbolic name of the Tx NSDU. Following the previous example, the application would have to call the function CanTp_StopSeparationTime using id 3.

Resolution:

Not resolved yet.

ESCAN00094464 NvM data is not stored after calling ComM_DeInit()	
Component@Subcomponent:	Ccl_Asr4ComMCfg5@Implementation
First affected version:	1.00.00
Fixed in versions:	8.01.01
Problem Description:	
What happens (symptoms):	

The status values of Mode Limitation and Wake-up Inhibition will be re-initialized on each ComM_Init() call.	
The following values are affected:	
<ul style="list-style-type: none"> - the state of Wake-up Inhibition if changed at run-time via ComM_PreventWakeUp() and - the number of inhibited requests of FULL_COM by ComM users assigned to channels which have Mode Limitation or Wake-up Inhibition activated and - the ECU Group Classification if changed at run-time via ComM_SetECUGroupClassification(). 	
When does this happen:	

It happens if ComM_DeInit() is called and at least one ComM channel is not in COMM_NO_COM_NO_PENDING_REQUEST state.	
Note, the behavior is as expected if ComM_DeInit() is called and all ComM channels are in COMM_NO_COM_NO_PENDING_REQUEST state.	
In which configuration does this happen:	

1) Mode Limitation Enabled is activated or Wake-up Inhibition Enabled is activated	
#define COMM_MODE_LIMITATION STD_ON or #define COMM_WAKEUP_INHIBITION STD_ON	
has been generated to ComM_Cfg.h	
AND	
2) Global NvM Block Descriptor is configured	
#define COMM_NVM_SUPPORT STD_ON has been generated to ComM_Cfg.h	
Resolution Description:	

ESCAN00094464 NvM data is not stored after calling ComM_DeInit()

Workaround:

1) Create a user configuration file with the following content and assign it to ComMUserConfigurationFile parameter.

```
#if defined CCL_ASR_COMM_SOURCE
# define ComM_DeInit ComM_DeInit_Original
#endif
```

2) Define "new" ComM_DeInit() function in a source file

```
#include <NvM.h>
extern FUNC(void, COMM_CODE) ComM_DeInit_Original(void);

FUNC(void, COMM_CODE) ComM_DeInit(void)
{
    /* Clear the No Com Mode Limitation bit, it shall not be stored to NvM */
    (void)ComM_LimitECUToNoComMode(FALSE);

    /* Set block flag to be written in NvM */
    (void)NvM_SetRamBlockStatus((NvM_BlockIdType)COMM_NVM_BLOCK_ID, TRUE);

    /* Execute the original ComM_DeInit() */
    ComM_DeInit_Original();
}
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095016 Service 0x23 responds with NRC 0x14 instead of 0x31	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	4.01.00
Fixed in versions:	8.05.00
Problem Description:	
What happens (symptoms):	

Service 0x23 (ReadMemoryByAddress) responds with NRC 0x14 instead of 0x31.	
When does this happen:	

Each time a valid (supported by the ECU configuration) memory block is requested with SID 0x23 will not fit the configured in DCM diagnostic buffer.	
In which configuration does this happen:	

- Service 0x23 supported by ECU and handled by DCM;	
AND	
- There are readable memory blocks configured in DCM (ECUC container DcmDspReadMemoryRangeInfo) that do not fit the diagnostic buffer assigned with the DcmDslProtocolRow supporting SID 0x23;	
OR	
- There are two or more readable memory blocks defined such	
- they are located side by side	
AND	
- have a common precondition that will allow reading all these blocks with a single request	
AND	
- their accumulated total length do no fit the configured buffer size as mentioned above.	
Resolution Description:	
Workaround:	

Implement service 0x23 processor in the application.	
Note: If SID 0x2C 0x02 shall be supported, DCM still handles it internally and requires the implementation of the callout Dcm_ReadMemory().	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095254 Missing DET error PDUR_E_PDU_INSTANCES_LOST description in case of N:1 communication interface routings with upper layer**Component@Subcomponent:** Gw_AsrPduRCfg5@Doc_TechRef**First affected version:** 2.08.00**Fixed in versions:** 3.00.01**Problem Description:**

What happens (symptoms):

DET error PDUR_E_PDU_INSTANCES_LOST is reported in case of N:1 communication interface routings with upper layer due to N:1.

This DET could occur in different situation. And a chapter which explain the scenarios is missing.

Hint: In case of a mixed N:1 Routing with upper layer, where the forwarding path has a confirmation, the destination is locked until the confirmation for the upper layer is called. If a gateway transmit function is called in the meantime this transmit request fails because the destination is not ready to receive the PDU and the DET is reported.

When does this happen:

During runtime

In which configuration does this happen:

Mixed Communication Interface N:1 routings with gateway and upper layer sources. And a TxConfirmation is configured for the forwarding pathway.

Resolution Description:

Workaround:

In case of communication interface N:1 routing with upper layer, use no buffer routing for the gateway path and routings without TxConfirmation for the forwarding path.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095441 [Only in case of multiple CAN driver are used] FIFO behavior is not fulfilled for transmission of CAN messages

Component@Subcomponent: If_AsrIfCan@GenTool_GeneratorMsr

First affected version: 4.03.00

Fixed in versions: 4.05.00

Problem Description:

What happens (symptoms):

FIFO behavior is not fulfilled for transmission of CAN messages

-> In case of transmission of CAN messages via the CanIf using the Tx-buffer of handling type: FIFO the CAN messages are not sent in order First in First Out but prioritized according to CAN identifier

When does this happen:

- At runtime

In which configuration does this happen:

Multiple CAN driver are configured

AND

At least one CAN driver has the parameter "CanMultiplexedTransmission" enabled

AND

There is at least one Tx-buffer of handling type FIFO configured for the CAN driver with enabled feature "CanMultiplexedTransmission"

(== at least one object "CanIfBufferCfg" is configured with "CanIfTxBufferHandlingType == FIFO")

Please note in this case the parameter "CanMultiplexedTransmission" must be disabled in order to fulfill FIFO behavior. Please ensure by your own that for all CAN drivers which has a Tx-buffer of handling type FIFO configured the feature "CanMultiplexedTransmission" is disabled.

Resolution Description:

Workaround:

Please ensure by your own that for all CAN drivers which has a Tx-buffer of handling type FIFO configured the feature "CanMultiplexedTransmission" is disabled.

-> After correction of configuration please generate, compile and load your configuration into ECU

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095651 Wrong software check for correct AuthID	
Component@Subcomponent:	DrvCry_Rh850Icum@Implementation
First affected version:	1.00.00
Fixed in versions:	1.01.00
Problem Description:	
What happens (symptoms):	

The AuthId is not checked correctly against the KeyId during the key update protocol. All combinations are evaluated as valid.	
An error will be generated in the hardware anyways. This error will then be reported to the CSM.	
Additionally the use case of updating a BOOT_MAC with the BOOT_MAC_KEY is missing as a valid combination.	
When does this happen:	

when the keyExtractUpdate is called with 65 Bytes of dataLength.	
In which configuration does this happen:	

all configurations	
Resolution Description:	
Workaround:	

Rely on the Hardware to detect the invalid M1-M3	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095653 Primitive returns with error with valid input parameters (using keyID on 2nd keypage)**Component@Subcomponent:** DrvCry_Rh850Icum@Implementation**First affected version:** 1.00.00**Fixed in versions:** 1.01.00**Problem Description:**

What happens (symptoms):

A primitive returns with CSM_E_NOT_OK even if the input arguments are valid.

The AES-128 Encryption, AES-128 Decryption, CMAC AES-128 Generation and CMAC AES-128 Verification fails and cannot provide a valid result.

When does this happen:

The issue can be noticed during runtime for following API function calls:

- Cry_30_Mpc577xC_AesDecryptUpdate
- Cry_30_Mpc577xC_AesEncryptUpdate
- Cry_30_Mpc577xC_CmacAes128GenUpdate
- Cry_30_Mpc577xC_CmacAes128VerFinish

If a raw configuration for the corresponding primitive is used and in the Start function of the corresponding primitive a keyID is used which is located on the second keypage.

In which configuration does this happen:

In all configurations if a raw-configuration is used for the occurred primitives.**Resolution Description:**

Workaround:

Only use mapped configurations for referencing key-IDs.

OR

Only use keys from the first key page (key 1 - 10).

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095919 Services 0x23/0x3D/0x2C: Unexpected response behavior on invalid memory Id request	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	7.02.00
Fixed in versions:	8.04.00
Problem Description: What happens (symptoms): ----- A: ECU sends RCR-RP responses endlessly. OR B: ECU sends RCR-RP until limit is reached and NRC 0x10 is sent. When does this happen: ----- Each and every time one of the following services 0x23/0x3D/0x2C is requested with an invalid memory identifier (MID). In which configuration does this happen: ----- For symptoms A and B: - Service 0x23 is supported (in Dcm_Cfg.h: DCM_SVC_23_SUPPORT_ENABLED == STD_ON) OR - Service 0x3D is supported (in Dcm_Cfg.h: DCM_SVC_3D_SUPPORT_ENABLED == STD_ON) OR - Service 0x2C 0x02 is supported (in Dcm_Cfg.h: DCM_SVC_2C_02_SUPPORT_ENABLED == STD_ON) AND - Memory identifier is supported (in Dcm_Cfg.h: DCM_MEMMGR_MID_SUPPORT_ENABLED == STD_ON) For symptom B: - The DCM is configured to limit the number of RCR-RP responses (in Dcm_Cfg.h: DCM_DIAG_RCRRP_LIMIT_ENABLED == STD_ON)	
Resolution Description: Workaround: ----- 1.) Configure system supplier specific request notification function (please see technical reference on chapter "How to Get Notified on a Diagnostic Service Execution Start and End" for further details). 2.) Within the Indication() operation the MID parameter has to be verified for the affected services. In case of invalid MID the NRC 0x31 shall be returned to Dcm. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00095963 Service 0x31: Erroneous implementation of request minimum length check

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 2.00.00

Fixed in versions: 8.04.00

Problem Description:

What happens (symptoms):

- ECU sends a positive response for service 0x31 (RoutineControl) instead of NRC 0x13.
OR

- The application C/S operation (i.e. Xxx_Start()/Xxx_Stop()/Xxx_RequestResults()) receives a very large value for the current request length value (i.e. a negative integer value represented as a huge unsigned value greater than 4095 for CAN or in general 32767). This may lead to a NRC 0x13 but also to a positive response transmission if the application does verify the request length for a minimum but not for maximum value, since DCM already does that.

When does this happen:

When any multi signal RID containing a signal with dynamic length is requested, but not all signals behind the RID parameter are available in the request.

In which configuration does this happen:

- Service 0x31 is supported (in Dcm_Cfg.h: DCM_SVC_31_SUPPORT_ENABLED == STD_ON)
AND

- At least one Routine with dynamic length signal in request is supported (in Dcm_Cfg.h: DCM_RIDMGR_DYN_REQ_LEN_ENABLED == STD_ON) that:

Additionally to the signal with dynamic length the Routine contains at least one more signal in the request (there is a /Dcm/DcmConfigSet/DcmDsp/DcmDspRoutineInfo with a DcmDspRoutineStartIn | DcmDspRoutineStopIn | DcmDspRoutineRequestResIn container including more than one DcmDspStartRoutineInSignal resp. DcmDspStopRoutineInSignal or DcmDspRoutineRequestResInSignal sub-containers, where the last one has DcmDspRoutineSignalType == VARIABLE_LENGTH).

Resolution Description:

Workaround:

Within the implementation of the C/S operations of the affected RIDs (i.e. Rte_Call_xxx_ROUTINE_NAME_Start/Stop/RequestResult() with more than one IN signal parameter where the last one represents array with dynamic length), please do verify:

- On a CAN ECU: that the function parameter "DataLength" does not have values greater than 4095.

- On a other bus system ECU that supports longer messages (e.g. up to 65535): that the function parameter "DataLength" does not have values greater than the configured DCM diagnostic buffer or in most cases the value 32767 shall be enough to detect a negative value.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096099 Main function tick time resolution smaller than 1 ms is not supported	
Component@Subcomponent:	Tp_Asr4TpCan@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

AUTOSAR defines a float data type to configure the main function tick time. The generator truncates the configured tick time to a millisecond. e.g. 1.5 ms tick time is configured, but the counter values generated in the code are based on a 1 ms tick time.	
As a result, wrong timer values are used by the module	
When does this happen:	

always	
In which configuration does this happen:	

If a main function tick time with a resolution smaller than 1 ms is configured	
Resolution Description:	
Workaround:	

Use a main function period which is an integer multiple of 1 ms.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096195 Rte_LdComCbKTriggerTransmit returns invalid error code	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.04.00
Fixed in versions:	1.16.00
Problem Description:	
What happens (symptoms):	

The LDCOM trigger transmit callback returns RTE_E_HARD_TRANSFORMER_ERROR instead of E_NOT_OK.	
When does this happen:	

This happens when the transformation of the data failed.	
In which configuration does this happen:	

This happens when SomeIpXf is configured for a LDCOM PDU that has trigger transmit enabled.	
Resolution Description:	
Workaround:	

Implement a trigger transmit callback in the application, overwrite the callback in the LDCOM configuration with the application callback.	
Call the RTE callback in the application callback and adjust the return code to E_NOT_OK when it is RTE_E_HARD_TRANSFORMER_ERROR.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096248 Validator PDUR12501 is not shown as error for PduRDestPduRef	
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	9.01.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- Validator message PDUR12501 is shown on a PduRDestPduRef. It is shown as information only, but in reality shall be an error message. ----- When does this happen: ----- During live validation in the DaVinci Configurator 5. ----- In which configuration does this happen: ----- The validator message is shown if the global Pdu of PduRDestPduRef is referenced by more than one other container. This kind of 1:N routing path is not supported.	
Resolution Description: Workaround: ----- Check if the validation message is shown for a PduRDestPduRef. If No, then you're not affected. If Yes, check if this is correctly configured. The PduR will only forward the Pdu to one of the destination container (the destination is chosen randomly while generating due to the internal Java structure). The routing to this destination container will then work as configured. ----- Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096369 CAN driver stops sending message.	
Component@Subcomponent:	DrvCan__coreAsr@Implementation
First affected version:	5.03.00
Fixed in versions:	5.03.05, 5.06.01, 5.04.05, 5.07.01, 5.05.06
Problem Description:	
What happens (symptoms):	

CAN driver stops sending message for one send mailbox (group of message or single message in case of FullCAN)	
A DET will occur because of FIFO overrun after some messages written to the FIFO.	
When does this happen:	

Only under specific circumstances:	
- in case a BusOff event interrupts a CanIf_Transmit() or Can_Write() call.	
In which configuration does this happen:	

CanIf uses FIFO buffer --> CanIfTxBufferHandlingType == FIFO	
(none AutoSar extension in CanIf)	
Resolution Description:	
Workaround:	

Use "CanBusoffProcessing" in "Polling" mode with lower priority than any message send operation.	
or	
call "STOP" and "START" transition after FIFO notifies overrun to DET or timeout for message confirmation occur	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096716 Queued sender/receiver N:1 connections not detected	
Component@Subcomponent:	Rte_Analyzer@Application
First affected version:	0.09.00
Fixed in versions:	1.01.00
Problem Description:	
What happens (symptoms):	

RteAnalyzer reports that configuration contains more queued connections and/or more APIs with queues.	
When does this happen:	

The error is issued during analysis of the code by RteAnalyzer in case the configuration is as described below.	
In which configuration does this happen:	

This happens for configurations with queued sender/receiver N:1 communication over partition boundaries, when all senders are mapped to the same OsApplication.	
Resolution Description:	
Workaround:	

Ensure that the reported connections exist in the code. Afterwards, the message can be disregarded,	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096901 Possible incorrect interpretation of last page status	
Component@Subcomponent:	If_AsrIfFeeSmallSector@Implementation
First affected version:	1.00.02
Fixed in versions:	2.00.01
Problem Description:	
What happens (symptoms):	

Depending on the content, a block might not be readable although it has been written correctly before. FEE may interpret content incorrectly, in case FlsBlankCheck is disabled for a block's partition.	
When does this happen:	

If the last flash page of an instance is entirely filled with FlsEraseValue. FEE reads the last page of an instance to determine whether an erase abort has occurred. If the content of the last flash page of an instance matches the FlsEraseValue, FEE will interpret the last page as erased and consequently assumes an erase abort.	
In which configuration does this happen:	

This issue is only relevant if parameter Fee/FeePartitionConfiguration/FeeFlsBlankCheckApi is set to FALSE. This issue is _NOT_ relevant for FEE usage on RH850 platform, because RH850 requires parameter Fee/FeePartitionConfiguration/FeeFlsBlankCheckApi to be set to TRUE.	
Resolution Description:	
Workaround:	

On RH850 platform it is highly recommended to enable parameter Fee/FeePartitionConfiguration/FeeFlsBlankCheckApi. With this parameter enabled, this issue does not apply. If SmallSectorFee is used on a different platform, where no FlsBlankCheck is necessary, the last page of an instance should not match the FlsEraseValue. The user has to make sure that the last page of block is _NOT_ filled with the erase value.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096926 a2l: Calculation of communication parameter does not consider PropSeg parameter	
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	3.00.00
Problem Description: What happens (symptoms): ----- The calculation of the communication parameters (BTL Cycles, Sample Point) only consider input parameter Seg1 and Seg2 This is not entirely correct for AUTOSAR as there is an additional parameter PropSeg which must also be considered. As a result the parameters in the CanXcp.a2l fragment are not correct which might lead to communication issues with the Xcp master. When does this happen: ----- Always and immediately In which configuration does this happen: ----- all configurations	
Resolution Description: Workaround: ----- The generated CanXcp.a2l fragment is only used by the Xcp Master Tool and can be modified manually to contain the correct BTL value. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097045 Dem_SetEventAvailable returns E_OK for events unavailable in variant	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	7.00.00
Fixed in versions:	12.01.04, 14.01.00
Problem Description:	
What happens (symptoms):	

API Dem_SetEventAvailable returns E_OK when called for an event unavailable in variant, although availability of this event cannot be changed during runtime.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

Dem/DemGeneral/DemAvailabilitySupport == DEM_EVENT_AVAILABILITY AND Dem/DemConfigSet/DemEventParameter/DemEventAvailableInVariant == FALSE for the event the API Dem_SetEventAvailable is called for.	
Resolution Description:	
Workaround:	

Don't call API Dem_SetEventAvailable for an event unavailable in variant. At least don't assume that an event unavailable in variant is set available during runtime although API returns E_OK.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097052 Data send completed trigger fired to early when Rte_ComSendSignalProxyPeriodic is used	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.02.00
Fixed in versions:	1.17.00
Problem Description:	
What happens (symptoms):	

A call to Rte_Write immediately triggers a runnable with data send completed trigger although COM did not call the confirmation callback, yet.	
When does this happen:	

During runtime.	
In which configuration does this happen:	

This happens when a signal is sent from a different partition than the partition that contains the COM module and when transmission acknowledgement is configured.	
Moreover a runnable needs to be configured to be triggered on data send completion.	
It only happens when there are no additional internal receivers and when /MICROSAR/Rte/RteGeneration/RteEnforceIoc is enabled in the RTE configuration.	
Resolution Description:	
Workaround:	

Configure an additional internal receiver.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097141 The Xcp Module ID is not according to AR 4	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	1.00.01, 3.00.00
Problem Description:	
What happens (symptoms):	

The service Xcp_GetVersionInfo() currently reports a MODULE_ID = 26. This value was used during the AR3 phase but is not according to AR4. In AR4 the MODULE_ID shall be 212.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

All configurations.	
Resolution Description:	
Workaround:	

The version information returned by Xcp_GetVersionInfo() can be interpreted accordingly (with different MODULE_ID)	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097176 When the XCP Master requests Timestamps but none are configured no negative response is returned**Component@Subcomponent:** Cp_Asr4Xcp@Implementation**First affected version:** 1.00.00**Fixed in versions:** 3.00.00**Problem Description:**

What happens (symptoms):

When the XCP Master requests timestamps to be active and no timestamps are configured in the XCP Slave, the Slave ignores this request. No negative response is returned to inform about this missing feature.

The resulting XCP frames do not contain timestamps while the XCP Master expects one. As a result the measurement is invalid.

When does this happen:

Always and immediately

In which configuration does this happen:

When an XCP Master requests timestamps but none are configured in the Slave.

Resolution Description:

Workaround:

Enable timestamps in the XCP configuration

or

Disable Slave timestamps in the XCP Master Tool

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097264 Service 0x2C: Valid DDDID definition requests always responded with NRC 0x31	
Component@Subcomponent:	Diag_Asr4Dcm@Description
First affected version:	1.02.00
Fixed in versions:	9.01.00
Problem Description: What happens (symptoms): ----- A diagnostic request for service 0x2C (DynamicallyDefineDataIdentifier) for definition of a DDDID (DynamicallyDefinedDID) is responded unexpectedly with NRC 0x31. After the issue occurred, the ECU is not blocked. All other services of the Dcm can be used normally. When does this happen: ----- Every time the affected by the configuration DDDID is requested for definition with service 0x2C. In which configuration does this happen: ----- - DCM supports and implements diagnostic service 0x2C (DynamicallyDefineDataIdentifier) AND - The affected DDDID is configured with 256 SourceItems (ECUC parameter /Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidAccess/DcmDspDidDefine/DcmDspDDDidMaxElements = 256)	
Resolution Description: Workaround: ----- Define only DDDID (DynamicallyDefinedDID) with "DcmDspDDDidMaxElements" up to 255 as restricted per AR DCM SWS 4.x. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097288 When fixed timestamps are configured but the XCP Master does not request them, no negative response is returned**Component@Subcomponent:** Cp_Asr4Xcp@Implementation**First affected version:** 1.00.00**Fixed in versions:** 3.00.00**Problem Description:**

What happens (symptoms):

When the XCP Slave has fixed timestamps configured (i.e. timestamps are always transmitted) but the XCP Master does not request timestamps in the SET_DAQ_LIST_MODE command, no error code is returned.

The resulting XCP frames do contain timestamps while the XCP Master expects none. As a result the measurement is invalid.

When does this happen:

Always and immediately

In which configuration does this happen:

When fixed timestamps are configured but the XCP Master requests none.

Resolution Description:

Workaround:

Enable Slave timestamps in the XCP Master Tool

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097333 ComM_BusSM_ModeIndication called with locked interrupts - OS ErrorHook on Os API Invocation

Component@Subcomponent: Ccl_Asr4SmCan@Implementation

First affected version: 2.00.00

Fixed in versions: 2.13.01

Problem Description:

What happens (symptoms):

The ComM triggers a BswM indication within the call context of the ComM_BusSM_ModeIndication().

Symptom 1)

OS API Invocation ends up with a call to ErrorHook.

Error Cause of this ErrorHook is that OS API is called whereas Interrupts are locked.

Errorcodes would then be:

osdErrATIntAPIDisabled 0x1104U,
osdErrHTMultipleActivation 0x1305U,
osdErrSEIntAPIDisabled 0x4105U,
osdErrSECallContext 0x4106U or
osdErrWEInterruptsDisabled 0x4403U

Symptom 2)

Rte Runnable that is triggered on TxError will not be called when Communication is shut down and Sending Request is outstanding

When does this happen:

Every time function "ComM_BusSM_ModeIndication" ends up in an Os API Invocation (for example ComM triggers BswM and BswM Rule to stop Ipdu Groups AND a transmission is still ongoing AND TxError is configured to trigger a runnable in the Rte)

AND

the function "ComM_BusSM_ModeIndication" is called with locked interrupts.

In which configuration does this happen:

If any BswM rule (which belongs to a ComM BswM indication), is configured as immediate

AND

an OS function is called within the call context of the BswM rule

AND

this OS function must not be called with locked interrupts.

e.g.

Std Rules in BswM are configured

AND

A Rte Runnable is triggered on Tx Error

AND

Osek OS which implements the requirement "If ActivateTask / SetEvent is called with locked Interrupts, reject the action and call an ErrorHook" is used (for example Vector Os)

Resolution Description:

ESCAN00097333 ComM_BusSM_ModeIndication called with locked interrupts - OS ErrorHook on Os API Invocation

Workaround:

Configure the BswM port of the ComM indication as deferred.

If the startup is not affected and a fast startup is desired it is possible to "copy" the port and set one as deferred (e.g. used for the shut down rules) and the other port as immediate (e.g. used for the startup rules).

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097361 Service 0x2A: Wrong prioritisation between NRC 0x13 and 0x31

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 1.01.00

Fixed in versions: 9.04.00

Problem Description:

What happens (symptoms):

ECU response with NRC 0x31 instead of NRC 0x13.

When does this happen:

If a service 0x2A is requested with an unsupported transmissionMode and no periodicDataIdentifier.

In which configuration does this happen:

- Service 0x2A is supported (in Dcm_Cfg.h: DCM_SVC_2A_SUPPORT_ENABLED == STD_ON)

Resolution Description:

Workaround:

Use supplierIndication function notification to catch this length check.

Note: Since the supplier specific Xxx_Indication() calls is performed after the SID level checks are executed, no session, security or other verification shall be performed within this callout - only the length check.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097377 Communication not possible in Multi Channel use case on channels > 0	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	3.00.00
Problem Description:	
What happens (symptoms):	

In a multi channel setup with different bus systems (e.g. CAN and TcpIp), communication is only possible on channel 0 (CAN).	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

In a multi bus system setup with the option "Multi Client Support" disabled.	
Resolution Description:	
Workaround:	

Activate the option Multi Client Support /MICROSAR/Xcp/XcpGeneral/XcpMultiClientSupport	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097381 Service 0x27: PowerOn delay time not started on single false access attempt

Component@Subcomponent: Diag_Asr4Dcm@Doc_TechRef

First affected version: 1.00.00

Fixed in versions: 9.03.00

Problem Description:

What happens (symptoms):

The ECU allows at power-on/reset immediate security access attempt. In other words: the first security access request for getting a new seed is positively responded, instead of sending NRC 0x37 (DelayTimeNotExpired).

When does this happen:

If the following sequence is ran:

- Security access request for getting seed of level X is positively responded.
- Security access request sending a key of level X failed with NRC 0x35 (i.e. prior reaching the false access attempt count limit).
- ECU is reset/powerd down.
- ECU is online/powerd on, already entered in a session supporting SID 0x27 and false access attempt delay time is not yet elapsed.
- Security access request for getting seed of level X is positively responded. <--- According to ISO14229-1:2013 NRC 0x37 is expected here.

In which configuration does this happen:

-
- Service 0x27 (SecurityAccess) is supported and implemented by DCM
- AND
- The brute-force-attack prevention via access attempt count and delay time is enabled (in Dcm_Cfg.h #define DCM_STATE_SEC_RETRY_ENABLED == STD_ON)
- AND
- The OEM does not override the defined in ISO14229-1:2013 defined behavior (i.e. the OEM may specify that the delay time at power-on/reset is started only if the restored attempt counter values is equal or greater than the specified limit other than single attempt)
- AND
- The attempt counter(s) are stored into non-volatile memory (in Dcm_Cfg.h #define DCM_STATE_SEC_ATT_CNTR_EXT_STORAGE_ENABLED == STD_ON)
- AND
- The false access attempt count limit is > 1 (in DCM ECUC there is at least one parameter: /Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDspSecurityNumAttDelay > 1)
- AND
- The above affected level supports the non-volatile storage of its attempt counter (in DCM ECUC the corresponding parameter of container DcmDspSecurityRow: DcmDspSecurityAttemptCounterEnabled = TRUE)

Resolution Description:

ESCAN00097381 Service 0x27: PowerOn delay time not started on single false access attempt

Workaround:

The DCM application of the affected project shall implement the Xxx_SetSecurityAttemptCounter() API as follows:

- If DCM passed a value = 0, just store it into NvM
- If DCM passes a value > 0, and the last stored in NvM values is 0, then and only then store the value 255.

Note: Value 255 is chosen to be configuration independent, in case the attempt count changes for instance from two to three.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097520 Callout Dcm_FilterDidLookupResult() is called with unexpected OpStatus

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 5.00.00

Fixed in versions: 9.01.00

Problem Description:

What happens (symptoms):

The callout Dcm_FilterDidLookupResult() is initially called with OpStatus DCM_PENDING instead of DCM_INITIAL.

When does this happen:

When a specific DID within a DID range is requested over any DID related diagnostic service. And the return value of the callout IsDidAvailable() is at least one time DCM_E_PENDING. And the final return values of the very same callout are E_OK and DCM_DID_SUPPORTED.

In which configuration does this happen:

- Any DID related diagnostic service is supported
- AND
- DID ranges with gaps are supported (in Dcm_Cfg.h: #define DCM_DIDMGR_OPTYPE_RANGE_ISAVAIL_ENABLED == STD_ON)
- AND
- External DID look up filtering is supported (in Dcm_Cfg.h: #define DCM_DIDMGR_EXTENDED_LOOKUP_ENABLED == STD_ON)

Resolution Description:

Workaround:

- Just ignore the OpStatus within Dcm_FilterDidLookupResult() if applicable (e.g. if filtering is done synchronously)

OR

- Manage the operation status by the application

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097602 CAN interrupt lost	
Component@Subcomponent:	DrvCan_Rh850McanAsr@Implementation
First affected version:	2.09.00
Fixed in versions:	2.20.01
Problem Description:	
What happens (symptoms):	

OS system error "Missing interrupt request" (E_OS_SYS_ASSERTION) occurs.	
When does this happen:	

At runtime in case an interrupt occurs while interrupts are not enabled by the CAN Driver.	
In which configuration does this happen:	

In all configurations except pure polling configurations.	
Resolution Description:	
Workaround:	

Configure Disable and Restore CAN Interrupts to be handled by the application.	
The application callback function shall handle the synchronization of the MCAN and INTC interrupt disabling as follows:	
For Rh850 (P1xC derivatives) the update of the MCAN.IE register has to be synchronized with the INTC:	
step 1: DI - Global Interrupt Disable, already accomplished by the Caller	
step 2: MCAN.IE = 0 - disable MCAN interrupt after storing the current content	
step 3: Dummy = MCAN.IE - dummy read of MCAN register	
step 4: SYNC - ensure that all previous instructions have been completed and the read data requested from a peripheral has arrived at the CPU.	
step 5: EI - Global Interrupt Enable, already accomplished by the Caller	
The application callback function shall handle the synchronization of the MCAN and INTC interrupt restoring as follows:	
step 6: restore the MCAN interrupt flags which were stored in step 2	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097637 EcuM calls the Mcu_SetMode API for setting the normal mcu mode with an invalid mcu mode	
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	8.00.03
Problem Description:	
What happens (symptoms):	

The Mcu_SetMode API is called with a wrong mcu mode which can lead to an array out of bounds write access.	
When does this happen:	

After wakeup from sleep.	
In which configuration does this happen:	

The Mcu_ModeType values start with 0 instead of 1.	
Resolution Description:	

ESCAN00097637 EcuM calls the Mcu_SetMode API for setting the normal mcu mode with an invalid mcu mode

Workaround:

The Mcu_SetMode API is called inside the callout EcuM_McuSetMode with the passed (and potentially wrong) McuMode.

To avoid passing the wrong mode to the Mcu the callout EcuM_McuSetMode can be adapted like the following:

```
FUNC(void, ECUM_CODE) EcuM_McuSetMode(Mcu_ModeType McuMode)
{
/
*****

* DO NOT CHANGE THIS COMMENT! <USERBLOCK EcuM_McuSetMode> DO NOT CHANGE THIS
COMMENT!

*****

/* Add implementation of EcuM_McuSetMode() */

if(McuMode == 0)
{
Mcu_SetMode(McuConf_McuModeSettingConf_McuModeSettingConf); // <=== Use symbolic
names provided by the Mcu here.
}

return;
/
*****

* DO NOT CHANGE THIS COMMENT! </USERBLOCK> DO NOT CHANGE THIS COMMENT!

*****

} /* End of EcuM_McuSetMode() */
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097731 Service 0x10: Jump to FBL performed although forced RCR-RP could not be sent	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	7.02.00
Fixed in versions:	9.03.00
Problem Description:	
What happens (symptoms):	

For a HIS compatible jump to FBL DCM triggers always a reset, independently of whether the forced RCR-RP could be sent by the transmission layer or not.	
When does this happen:	

Whenever a jump to FBL is requested, e.g. via a 0x10 0x02 request.	
In which configuration does this happen:	

- Service 0x10 is supported (in Dcm_Cfg.h: #define DCM_SVC_10_SUPPORT_ENABLED == STD_ON)	
AND	
- Jump to bootloader is supported (in Dcm_Cfg.h: #define DCM_SVC_10_JMP2BOOT_ENABLED == STD_ON)	
AND	
- RCR-RP on jumping to the FBL is supported (in Dcm_Cfg.h: #define DCM_DIAG_RCRRP_ON_BOOT_ENABLED == STD_ON)	
Resolution Description:	
Workaround:	

The workaround for this issue involves overriding default service handling for service "DiagnosticSessionControl" (0x10).	
Please contact Vector for technical details and support on how to implement it in order to avoid any unwanted side effects.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097925 Memory read access exception due to incorrect address conversion	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	3.00.01
Problem Description:	
What happens (symptoms):	

The component raises an memory access exception on Posix systems due to an read access to a invalid address. This happens when the XCP command GET_DAQ_EVENT_INFO is used. This command is activated when the option: /MICROSAR/Xcp/XcpCmdConfig/XcpDaqAndStim/XcpGetDAQEventInfo is enabled.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

When the feature Event info is used	
AND	
the component is used on a operating system that uses relative addresses like Windows or Posix and the addresses are converted from virtual to physical addresses in the Xcp_GetPointer call-back.	
Resolution Description:	
Workaround:	

Get Daq Event Info is an optional command. It can be deactivated and the information from XCP_events.a2I can be used instead.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098007 Declined Immediate Nm Transmissions is retried later than expected

Component@Subcomponent: Nm_Asr4NmCan@Doc_TechRef

First affected version: 1.00.00

Fixed in versions: 7.00.00

Problem Description:

What happens (symptoms):

The technical reference contains unprecise information about the procedure when an immediate Nm transmission was declined.

Current documentation states in chapter 3.15 that transmit request is retried after Immediate Nm message cycle time.

It should be: After main function cycle time.

/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmImmediateNmCycleTime
/MICROSAR/CanNm/CanNmGlobalConfig/CanNmMainFunctionPeriod

When does this happen:

-

In which configuration does this happen:

-

Resolution Description:

Workaround:

Note in chapter 3.15 should be described as follows:

Note

If the send request of an 'immediate transmission' is rejected by the lower layer (e.g. CanIf), the rejected send request is not considered as 'immediate transmission'. That means that the counter that counts the number of 'immediate transmissions' ImmediateNmMsgCount is not decremented. The transmission will be retried in the next main function cycle.

Example: Let 'Immediate Nm Transmissions' := 2. The initial counter value of ImmediateNmMsgCount is 1.

1. When Repeat Message has just been entered, the first transmission request TReq A is rejected. ImmediateNmMsgCount is not decremented.
2. CanNm waits until the next main function cycle of CanNm (first interval t int1st).
3. CanNm sends the NM message successfully. ImmediateNmMsgCount is decremented to 0.

4. CanNm waits 'Immediate Msg CycleTime' (second interval t int2nd).

5. CanNm sends the next NM message successfully.

6. Then 'Msg Cycle Time' is waited until the next NM message is sent because ImmediateNmMsgCount is already 0.

If the first NM transmission request TReq A was successful (step 1), the second interval time t int2nd would be 'Msg Cycle Time' instead of 'Immediate Msg CycleTime'.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098036 Cyclic PDUs with a ComGwRoutingTimeout are triggered when started if ComTxDIMonTimeBase is configured**Component@Subcomponent:** Il_AsrComCfg5@Implementation**First affected version:** 8.01.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

Periodic or Mixed Tx ComIPdu's with a configured ComGwRoutingTimeout might get triggered after their respective ComIPduGroup is started.

When does this happen:

This issue occurs at runtime after the ComIPduGroup is started within the next Com_MainFunctionTx call.

In which configuration does this happen:

Tx ComIPdu is part of any routing relation
AND
ComGwRoutingTimeout is configured for the Tx ComIPdu
AND
ComMainfunctionTimingDomainSupport is enabled
AND
ComTxDIMonTimeBase is configured

Resolution Description:

Workaround:

Deactivate ComTxDIMonTimeBase

OR

Set resolution of ComTxDIMonTimeBase higher than ComTxCycleCounterTimeBase, such that ComTxCycleCounterTimeBase is a multiple of ComTxDIMonTimeBase.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098095 PduInfoPtr of API Appl_GenericConfirmation() contain DLC instead of message length**Component@Subcomponent:** DrvCan__coreAsr@Implementation**First affected version:** 4.01.00**Fixed in versions:** 6.00.00**Problem Description:**

What happens (symptoms):

PduInfoPtr of API Appl_GenericConfirmation() contain DLC instead of message length

When does this happen:

while runtime always and immediate when a FD-message is received.

In which configuration does this happen:

CanGenericConfirmation is set to API2 (CAN_GENERIC_CONFIRMATION == CAN_API2)
AND
CAN-FD messages supported (DLC / Length > 8byte)**Resolution Description:**

Workaround:

calculate length out of DLCDlcToLengthMap[16] =
{
 0, 1, 2, 3,
 4, 5, 6, 7,
 8, 12, 16, 20,
 24, 32, 48, 64
};

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098361 Service 0x2F: Wrong ReturnControlToECU operations are called on session/security state change

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 4.00.00

Fixed in versions: 9.04.00

Problem Description:

What happens (symptoms):

DCM calls wrong ReturnControlToECU() operations on session/security state change for active IO controls.

When does this happen:

- During transition to default session.

OR

- During transition to any non-default session if any state restriction of an active IO control is no longer fulfilled.

In which configuration does this happen:

- Service 0x24 is supported and handled by DCM (in Dcm_Cfg.h: #define DCM_SVC_24_SUPPORT_ENABLED == STD_ON)

AND

- Service 0x2F is supported and handled by DCM (in Dcm_Cfg.h: #define DCM_SVC_2F_SUPPORT_ENABLED == STD_ON)

AND

- The operation ReturnControlToEcu is configured for any IO control (in Dcm_Cfg.h: #define DCM_DIDMGR_OPTYPE_IO_RETCTRL2ECU_ENABLED == STD_ON)

AND

- At least one DID supports scaling and any IO operation (in Dcm_Cfg.h: #define DCM_DIDMGR_OP_INFO_COMBINED_ENABLED == STD_ON)

AND

- Any DID supporting the scaling operation has an ID lower than that one of any IO DID

Resolution Description:

Workaround:

1.) Configure system supplier specific request notification function (please see technical reference on chapter "How to Get Notified on a Diagnostic Service Execution Start and End" for further details).

2.) Within the Indication() operation keep track of the active IO DIDs.

3.) Call the ReturnControlToECU() operation on each session/security change for each IO DID no longer supported in the new state.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099078 RTE generator does not trigger NvM_MainFunction when a NvBlock SWC has no NvBlockDescriptors	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.19.00
Problem Description: What happens (symptoms): ----- NV memory is not restored or persisted. When does this happen: ----- Always. During runtime. In which configuration does this happen: ----- This happens when the configuration contains NvBlockSWCs without NvBlockDescriptors. The main functionality of NvBlockSWCs is to provide access to NvBlocks. It is therefore assumed that this is an uncommon configuration that only occurs during development.	
Resolution Description: Workaround: ----- Create a NvBlockDescriptor in the NvBlockSWC. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00099239 Enabled event memory that has no events assigned leads to wrong code generation

Component@Subcomponent: Diag_Asr4Dem@GenTool_GeneratorMsr

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Generated code leads to undefined behavior in DEM.

When does this happen:

When requesting services (e.g. ClearDTC) for an event memory.

In which configuration does this happen:

```
(
  Dem/DemGeneral/DemMaxNumberEventEntrySecondary > 0
  AND
  No event with Dem/DemConfigSet/DemEventParameter/DemEventClass/DemEventDestination
  == DEM_DTC_ORIGIN_SECONDARY_MEMORY configured
)
OR
(
  Dem/DemGeneral/DemMaxNumberEventEntryPrimary > 0
  AND
  No event with Dem/DemConfigSet/DemEventParameter/DemEventClass/DemEventDestination
  == DEM_DTC_ORIGIN_PRIMARY_MEMORY configured
)
```

Resolution Description:

Workaround:

Set Dem/DemGeneral/DemMaxNumberEventEntrySecondary == 0, if no event with Dem/DemConfigSet/DemEventParameter/DemEventClass/DemEventDestination == DEM_DTC_ORIGIN_SECONDARY_MEMORY is configured
 If no event with Dem/DemConfigSet/DemEventParameter/DemEventClass/DemEventDestination == DEM_DTC_ORIGIN_PRIMARY_MEMORY is configured, set Dem/DemGeneral/DemMaxNumberEventEntryPrimary == 1
 and configure dummy event with Dem/DemConfigSet/DemEventParameter/DemEventClass/DemEventDestination == DEM_DTC_ORIGIN_PRIMARY_MEMORY

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099440 RTE sends unexpected data when activation reasons are configured for a runnable with implicit write accesses	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.15.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

A software component receives unexpected data.	
When does this happen:	

During runtime when a runnable is triggered several times but is executed only once.	
In which configuration does this happen:	

This happens when activation reasons are configured for a runnable with implicit write accesses.	
Resolution Description:	
Workaround:	

Use explicit instead of implicit communication or enable the parameter /MICROSAR/Rte/RteGeneration/RteInitializeImplicitBuffers.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099458 Channel restarts when all Partial Networks are released and channel is supposed to shut down

Component@Subcomponent: Nm_Asr4NmCan@GenTool_GeneratorMsr

First affected version: 5.00.00

Fixed in versions: 9.00.00

Problem Description:

What happens (symptoms):

The network is restarted shortly before it is supposed to shut down.

The ECU is potentially kept awake forever.

As a consequence NM and application messages are transmitted unexpectedly.

When does this happen:

-
1. A PNC is requested internally or a PNC request is received from another ECU at least once.
 2. All internal and externally requested PNC have been released, the shutdown sequence is initiated.
 3. The affected channel is restarted.

In which configuration does this happen:

Partial Networking must be enabled and either Pn Eira Calculation or Pn Era Calculation is enabled.

Additionally, the Pn Reset Time must be bigger than the Timeout Time.

```
- '/MICROSAR/ComM/ComMGeneral/ComMPncPrepareSleepTimer' is not configured
AND
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmPnEnabled'
AND
(
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmPnEraCalcEnabled' on at
least one channel
OR
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmPnEiraCalcEnabled'
)
AND
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmPnResetTime' > '/MICROSAR/CanNm/
CanNmGlobalConfig/CanNmChannelConfig/CanNmTimeoutTime'
```

Resolution Description:

ESCAN00099458 Channel restarts when all Partial Networks are released and channel is supposed to shut down

Workaround:

Set the TimeoutTime greater than PnResetTime + PnCPrepareSleepTimer.
Refer to TechnicalReference_Asr_ComM.pdf for further details.

If PnCPrepareSleepTimer is not used, assume it has value zero.

Affected parameters:

- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmPnResetTime'
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmTimeoutTime'
- '/MICROSAR/ComM/ComMGeneral/ComMPncPrepareSleepTimer'

Resolution:

A validator is modified that prevents the generation of the wrong configuration.

ESCAN00099480 Dirty flags not handled when multiple implicit accesses for the same NvBlock descriptor are configured and implicit write is skipped

Component@Subcomponent: Rte_Core@Implementation

First affected version: 1.08.00

Fixed in versions: 1.19.00

Problem Description:

What happens (symptoms):

A NvBlock is not written to NVRAM.

When does this happen:

During runtime, when the application writes NvBlocks with implicit write accesses and does not call Rte_IWrite or Rte_IWriteRef for all configured port accesses in every runnable call.

In which configuration does this happen:

This happens when the configuration contains multiple write accesses to the same NvBlockDescriptor in the same runnable. Moreover the dirty flag needs to be configured for the NvBlockDescriptor and InitializeImplicitBuffers needs to be configured to allow that Rte_IWrite and Rte_IWriteRef calls are skipped.

Resolution Description:

Workaround:

Call Rte_IWrite or Rte_IWriteRef for all data elements with dirty flag handling.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099536 Safety Manual does not fit to current description file / generator output

Component@Subcomponent: If_Asr4IfWd@root

First affected version: 2.01.00

Fixed in versions: 2.01.03

Problem Description:

What happens (symptoms):

The safety manual was not updated after a update of the description file. Therefore the generator output did not fit to the generated output. Within the safety manual, the user has to verify some parameters / pre processor switches which were no more available.

These parameters were:

- WdgIfUseAutosarDrvApi
- WdgIfInternalTickCounterRef
- WdgIfStateCombinerUseManualMode

Due to the fact that the WdgIfStateCombinerUseManualMode was removed, also the generator output has changed. Therefore also the Safety Manual was updated. The WdgIf_Interface does no more contain wdgif_statecombiner_common_config and wdgif_statecombiner_manual_config, but only wdgif_statecombiner_config.

WdgIf_StateCombinerManualConfigType does no more exist - instead of WdgIf_StateCombinerSlaveTriggerPatternType was introduced.

When does this happen:

This issue is no issue depending to a runtime error. This issue does only effect the manual code verification procedure.

This issue is active only if the last entry of the change history in WdgIf_bswmd.arxml is 6.01.00.

In which configuration does this happen:

Always during verification procedure described in safety manual.

Resolution Description:

ESCAN00099536 Safety Manual does not fit to current description file / generator output

Workaround:

Use the new safety manual (r.2.624.118):

1 Safety Manual WdgIf

1.1 Safety features

SMI-519

This component provides the following safety features:

ID Safety feature

CREQ-107414 WdgIf shall provide a service to set the mode of a watchdog device

CREQ-107415 WdgIf shall provide a service to set the trigger condition for a watchdog device

CREQ-107416 WdgIf shall support a mechanism to combine statuses of different cores and handle one watchdog for different cores

1.2 Configuration constraints

SMI-522

If a state combiner is used, the user of MICROSAR Safe shall configure

o WdgIfStateCombinerSpinlockID and

o WdgIfStateCombinerStartUpSyncCycles

for each core that is used by the state combiner.

SMI-523

If a state combiner is used and WdgIfStateCombinerStartUpSyncCycles is set to a value *s*, the user of MICROSAR Safe shall consider that for the first *s* SupervisionCycles of the master, the master does not monitor the slave triggers.

However, reset requests from a slave within the first *s* SupervisionCycles, are escalated by the master with the next call of the master's WdgM_MainFunction().

1.3 Additional verification measures

SMI-525

The user of MICROSAR Safe shall verify that the output path of the generator is empty before the generator is started.

The output path can be defined by the command line argument OUTPUT-DIRECTORY.

SMI-526

The user of MICROSAR Safe shall inspect the messages of the generator execution.

If the generator aborts the generation process with an error message, the (partially) generated output files shall not be used in the system.

If the generator detects an error, a message starting with "ERROR" is displayed on the standard output.

If the generator shows a warning message starting with "WARNING", the user of MICROSAR Safe shall ensure that the cause of the warning does not invalidate the generated output files.

SMI-527

The user of MICROSAR Safe shall verify that the following preprocessor directives are defined with the correct value independent from whether a state combiner is configured or not:

Preprocessor Directive Value

WDGIF_VERSION_INFO_API STD_ON if WdgIfVersionInfoApi is TRUE, otherwise STD_OFF.

WDGIF_DEV_ERROR_DETECT STD_ON if WdgIfDevErrorDetect is TRUE, otherwise STD_OFF.

WDGIF_USE_STATECOMBINER STD_ON if WdgIfUseStateCombiner is TRUE, otherwise STD_OFF.

The defines can be found in WdgIf_Cfg_Features.h.

SMI-528

The user of MICROSAR Safe shall verify that the following preprocessor directives are defined with the correct value only if a state combiner is configured:

Preprocessor Directive Value

WDGIF_STATECOMBINER_USE_OS_SPIN_LOCK STD_ON if WdgIfStateCombinerUseOsSpinlock is TRUE, otherwise STD_OFF.

This define can be found in WdgIf_Cfg_Features.h.

SMI-529

The user of MICROSAR Safe shall verify that the following preprocessor directives are defined with the correct value independent from whether a state combiner is configured or not:

ESCAN00099536 Safety Manual does not fit to current description file / generator output

Preprocessor Directive Value
WDGIF_NUMBER_OF_WDGIFDEVICES The number of configured WD Interface Devices.
This define can be found in WdgIf_LCf.h.
SMI-530
The user of MICROSAR Safe shall verify that the following preprocessor directives are defined with the correct value only if a state combiner is configured:
Preprocessor Directive Value
WDGIF_NUMBER_OF_SLAVES The configured number of slave cores. Cores that are not attached to a State Combiner (i.e. they run independent from other cores) do not count as slave.
This define can be found in WdgIf_LCf.h.
SMI-531
The user of MICROSAR Safe shall verify that the C-struct const WdgIf_InterfaceType WdgIf_Interface is defined in WdgIf_LCf.c.
WdgIf_Interface shall contain the following fields:
Field Value Description
1st WDGIF_NUMBER_OF_WDGIFDEVICES The number of WD Interface Devices.
2nd WdgIf_FunctionsPerWdg A reference to the list of WD Interface Devices.
If a state combiner is configured, WdgIf_Interface shall also contain the following field:
Field Value Description
3rd &wdgif_statecombiner_config A reference to the state combiner data structure.
SMI-532
The user of MICROSAR Safe shall verify that the array static const WdgIf_InterfaceFunctionsPerWdgDeviceType WdgIf_FunctionsPerWdg [WDGIF_NUMBER_OF_WDGIFDEVICES] is defined in WdgIf_LCf.c.
WdgIf_FunctionsPerWdg shall refer all – and only - the WD Interface Devices that are configured in the EDF.
WdgIf_FunctionsPerWdg[i] shall refer the underlying WD Interface Device in the EDF with WdgIfDeviceIndex = i.
The fields in an array element in WdgIf_FunctionsPerWdg shall be set as follows:
Field Value Description
1st &device_functions If the underlying WD Interface Device is directly linked to a WD driver for device, then the field refers to the C-struct device_functions.
1st NULL_PTR If the underlying WD Interface Device shares the WD driver with other WD Interface Devices using a state combiner, then the linked WD device is referred in wdgif_statecombiner_config.
If a state combiner is configured, an array element in WdgIf_FunctionsPerWdg shall also contain the following field:
Field Value Description
2nd WdgInstance A number that uniquely identifies the WD Interface Device instance for the underlying platform. All instances on the same platform are numbered consecutively starting with 0. Example: One platform has instances A and B, another platform has instances C and D. Then A, B, C, and D have WdgInstance set (in this order) as: 0,1,0,1.
SMI-533
If a state combiner is configured, the user of MICROSAR Safe shall verify that the array static const WdgIf_StateCombinerConfigType wdgif_statecombiner_config is defined in WdgIf_LCf.c.
The fields in wdgif_statecombiner_config shall be set as follows:
Field Value Description
1st WDGIF_NUMBER_OF_SLAVES The number of configured slaves.
2nd WdgIfStateCombinerSpinlockID The value of field WdgIfStateCombinerSpinlockID in the EDF.
3rd WdgIfStateCombinerStartUpSyncCycles The value of field WdgIfStateCombinerStartUpSyncCycles in the EDF.
4th &device_functions A reference to the WD driver API functions for device.
5th wdgif_statecombiner_shared_memory A reference to the shared memory for the state combiners of the Watchdog Interfaces.
5th wdgif_statecombiner_slave_trigger_pattern A reference to the array of the state combiners

ESCAN00099536 Safety Manual does not fit to current description file / generator output

trigger pattern of each slave.

SMI-534

If a state combiner is configured, the user of MICROSAR Safe shall verify that the array `WdgIf_StateCombinerSharedMemory wdgif_statecombiner_shared_memory [WDGIF_NUMBER_OF_SLAVES]` is defined in `WdgIf_LCf.c`.

The `WdgIf` writes to this array, hence it is not const.

`wdgif_statecombiner_shared_memory` shall contain one array element for each configured slave and the fields in the element shall be set as shown in the following table:

Field	Value	Description
1st	0u	Initial value for the slave's counter.
2nd	(uint32)~0u	Inverse initial value for the slave's counter.

SMI-536

If the state combiner is configured in manual mode, the user of MICROSAR Safe shall verify that for a configured slave with ID the C-struct static const

`WdgIf_StateCombinerSlaveTriggerPatternType wdgif_statecombiner_config_slave<ID>` is defined in `WdgIf_LCf.c`.

`wdgif_statecombiner_config_slave<ID>` shall be set as follows:

Field	Value	Description
1st	<code>WdgIfStateCombinerReferenceCycle</code>	The value of field <code>WdgIfStateCombinerReferenceCycle</code> in the EDF.
2nd	<code>WdgIfStateCombinerSlaveIncrementsMin</code>	The value of field <code>WdgIfStateCombinerSlaveIncrementsMin</code> in the EDF.
3rd	<code>WdgIfStateCombinerSlaveIncrementsMax</code>	The value of field <code>WdgIfStateCombinerSlaveIncrementsMax</code> in the EDF.

SMI-537

The user of MICROSAR Safe shall verify that for each configured platform the C-struct static const `WdgIf_InterfaceFunctionsType <WdgIfDevice>_functions` is defined in `WdgIf_LCf.c`.

The fields for each C-struct `<WdgIfDevice>_functions` shall be set as follows:

Field	Value	Description
1st	<code>Wdg__SetMode_</code>	A reference to the WD driver's API function to set the mode of the device referred to by infix.
2nd	<code>Wdg__SetTriggerCondition_</code>	A reference to the WD driver's API function to set the trigger condition of the device referred to by infix.

1.4 Safety features required from other components

SMI-520

This component requires the triggering of the watchdog and setting the triggering mode as a safety feature from the watchdog driver.

This requirement is fulfilled if a watchdog driver by Vector is used.

SMI-3085

The `WdgIf` shall be used in order to call the services of underlying Watchdog driver(/s) to set the mode and the trigger condition as expected by the drivers.

The user of MICROSAR Safe shall verify that the `WdgIf` services are only called by the `WdgM`.

This requirement is fulfilled for all components of MICROSAR (if Vector's `WdgM` is used).

If e.g. the trigger condition is called by another component, this may lead to unintended triggering of the watchdog.

If the watchdog stack is not properly set up, it may not provide the expected protection.

1.5 Dependencies to hardware

This component does not use a direct hardware interface.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099665 Xcp_Event throws DET check if called before Xcp_Init

Component@Subcomponent: Cp_Asr4Xcp@Implementation

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

The DET error XCP_E_UNINIT is thrown with the function id XCP_SID_EVENT when the Xcp_Event function is called before the component is initialized. This might be a valid use case when e.g. the Xcp_Event function is called from a different core with a different initialization duration or if it is called from interrupt functions.

If DET is entirely deactivated the function is called while the XCP might be in an uninitialized state. In this case the following might happen:

RAM is initialized to 0 by startup code: No negative side effects will happen

RAM is not initialized: XCP might express undefined behaviour and sample data which is not valid.

The safe way is to keep the SafeBsw Checks enabled as described in chapter workaround.

When does this happen:

Always and immediately

In which configuration does this happen:

When DET reporting is on
(/MICROSAR/Xcp/XcpGeneral/XcpDevErrorDetect is enabled)

Resolution Description:

Workaround:

As a temporary work around disable DET error reporting while the Safe BSW checks are still enabled.

Disable: /MICROSAR/Xcp/XcpGeneral/XcpDevErrorDetect

Enable: /MICROSAR/Xcp/XcpGeneral/XcpSafeBswChecks

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00100047	Consecutive failed cycle counter not persisted to NvRAM
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	9.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	
----- After ECU restart updates to Consecutive Failed Cycle Counter can be lost, although synchronization to NvRAM was triggered. As a side effect selected status bits like PDTC bit may wrongly be restored by the DEM during startup, although it was reset before shutdown.	
When does this happen:	
----- 1. Consecutive Failed Cycle Counter contains a non zero value which is persisted to NvRAM. 2. Consecutive Failed Cycle counter is reset at the end of a following operation cycle with a passed result. 3. Synchronization to NvRAM is triggered e.g. by Dem_Shutdown or Dem_RequestNvSynchronization. 3. Re-Initialize the Dem.	
In which configuration does this happen:	
----- Event has Dem/DemGeneral/DemDataClass/DemDataElementInternalData element 'CONSECUTIVE_FAILED_CYCLES' configured in a snapshot or extended data record AND /Dem/DemGeneral/DemUseNvm == TRUE	
Resolution Description:	

ESCAN00100047 Consecutive failed cycle counter not persisted to NvRAM

Workaround:

During the ECU startup, after restoring the NV contents, create a copy of the cycle counters from the Dem NV data:

```
Dem_Cfg_Primary/SecondaryEntry_0.ConsecutiveFailedCycleCounter
Dem_Cfg_Primary/SecondaryEntry_1.ConsecutiveFailedCycleCounter
...
Dem_Cfg_Primary/SecondaryEntry_<N>.ConsecutiveFailedCycleCounter
```

After Dem_Shutdown, check whether the copies still contain the values stored in Dem_Cfg_Primary/SecondaryEntry_<N>

If the values differ, mark the block as modified (In Autosar systems using NvM_SetRamBlockStatus()).

Example for primary memory:

After Initialization:

```
uint8 backup[DEM_CFG_GLOBAL_PRIMARY_SIZE];
for (IBlockIndex = 0; IBlockIndex < DEM_CFG_GLOBAL_PRIMARY_SIZE; IBlockIndex++)
{
    backup[IBlockIndex] = (Dem_Cfg_MemoryDataPtr[DEM_CFG_MEMORY_PRIMARY_INDEX +
    IBlockIndex])->ConsecutiveFailedCycleCounter;
}
```

After Shutdown:

```
for (uint8 IBlockIndex = 0; IBlockIndex < DEM_CFG_GLOBAL_PRIMARY_SIZE; IBlockIndex++)
{
    if (backup[IBlockIndex] != (Dem_Cfg_MemoryDataPtr[DEM_CFG_MEMORY_PRIMARY_INDEX +
    IBlockIndex])->ConsecutiveFailedCycleCounter)
    {
```

```
        NvM_SetRamBlockStatus(NvM_BlockIdType)Dem_Cfg_MemoryBlockId[DEM_CFG_MEMORY_PRIMARY
        + IBlockIndex], TRUE);
    }
}
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00100243 incorrect ODT messages assembled if ODT Entries are initialized incompletely

Component@Subcomponent: Cp_Asr4Xcp@Implementation

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

When the XCP Master uses a sequence like the following to only partially initialize the DAQ list:

```
ALLOC 1 DAQ
ALLOC ODT (2 ODTs )
ALLOC 7 ODT entries in ODT 0
ALLOC 7 ODT entries in ODT 1
```

```
Set DAQ pointer in ODT 0
Write DAQ
Set DAQ pointer in ODT 1
Write DAQ
Write DAQ
```

In each ODT, uninitialized entries remain (with address and length = 0). This will cause the Slave to assemble incomplete ODTs, because such uninitialized entries are ignored and cause an internal index not to be incremented correctly.

The impact is the following: The resulting DAQ list, sent on the bus, is incomplete.

When does this happen:

When a DAQ list is initialized partially like described above.

In which configuration does this happen:

When DAQ measurement is used
/MICROSAR/Xcp/XcpCmdConfig/XcpDaqAndStim

Resolution Description:

Workaround:

A Master Tool like CANape always initializes all ODT Entries that it has allocated, so this issue will only arise if a own Tool or script is used to create DAQ lists and only partially initializes them. The tools used must ensure that all Entries that are allocated are also initialized correctly.

Resolution:

The described issue is corrected by modification of all affected work-products.

2.4 Not Released Functionality

Not released functionalities (BETA) are either complete software modules or features in the software module that have not yet passed a complete development cycle (they are e.g. not or only partly tested). If a BETA issue ticket affects a complete software module, the software module must not be used for serial production. If a BETA issue ticket affects a feature in the software module, the user has to ensure that all BETA features are disabled as indicated for the serial production release of the ECU.

Index

ESCAN00088830	BETA version - the BSW module has a feature with BETA state (Memory Protection in trusted applications) <small>Os_CoreGen7@Implementation</small>
ESCAN00089701	BETA version - the BSW module has a feature with BETA state (Executing trusted applications in non privileged mode) <small>Os_CoreGen7@Implementation</small>
ESCAN00091204	BETA version - the Nm module has a feature with BETA state (FEAT-1865) <small>Nm_Asr4NmIf@Implementation</small>
ESCAN00091218	BETA version - the BSW module has a feature with BETA state (FEAT-371) <small>Diag_Asr4Dcm@Implementation</small>
ESCAN00091231	BETA version - the BSW module has a feature with BETA state (FEAT-1899) <small>Diag_Asr4Dcm@GenTool_GeneratorMsr</small>
ESCAN00092470	BETA version - the BSW module has a feature with BETA state (FEAT-1454) <small>SysService_Asr4BswMCfg5@GenTool_GeneratorMsr</small>
ESCAN00092868	BETA version - the BSW module is in BETA state <small>DrvCry_Rh850Icum@Implementation</small>
ESCAN00093797	BETA version - the BSW module has a feature with BETA state (Barriers) <small>Os_CoreGen7@Implementation</small>
ESCAN00093813	BETA version - the BSW module has a feature with BETA state (Fast Trusted Functions) <small>Os_CoreGen7@Implementation</small>
ESCAN00094381	BETA version - the BSW module has a feature with BETA state (FEAT-2160) <small>Diag_Asr4Dcm@Implementation</small>
ESCAN00094537	BETA version - the BSW module has a feature with BETA state (FEAT-2084) <small>Cp_Asr4Xcp@Implementation</small>

ESCAN00088830	BETA version - the BSW module has a feature with BETA state (Memory Protection in trusted applications)
Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What is the impact of BETA software:	

BETA feature:	
<ul style="list-style-type: none"> - must not be used in productive projects as they may result in unpredictable ECU behavior - may not provide all features intended for the productive project - is not tested and not all quality measures have taken place 	
Which functionality is BETA:	

The following feature/function is in BETA state.	
<ul style="list-style-type: none"> - Memory Protection in trusted applications. 	
To ensure that only productive code is used verify that:	
<ul style="list-style-type: none"> - OsTrustedApplicationWithProtection is false for all applications. 	
Resolution Description:	
Workaround:	

Do not use memory protection for trusted.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00089701 BETA version - the BSW module has a feature with BETA state (Executing trusted applications in non privileged mode)**Component@Subcomponent:** Os_CoreGen7@Implementation**First affected version:** 1.00.00**Fixed in versions:****Problem Description:**

What is the impact of BETA software:

BETA feature:

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not tested and not all quality measures have taken place

Which functionality is BETA:

The following feature/function is in BETA state.

- Executing trusted applications in non privileged mode is implemented as a BETA feature:

To ensure that only productive code is used verify that:

- IsPrivileged is TRUE for all trusted applications

Resolution Description:

API Extensions:

No extension of the API.

API Changes:

No modification of the API.

Module handling changes:

No modification of the module handling.

For a detailed description of the API and the handling of the module refer to the Technical Reference.

ESCAN00091204 BETA version - the Nm module has a feature with BETA state (FEAT-1865)

Component@Subcomponent: Nm_Asr4NmIf@Implementation

First affected version: 10.00.00

Fixed in versions:

Problem Description:

What is the impact of BETA software:

BETA software

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not or only partly tested and not all quality measures have taken place

Which functionality is BETA:

The following feature/function is in BETA state.

The NmOsek has to support the specific coordination use cases:

- Use different intervals between the Nm_SynchronizationPoint() function call and the expected NmOsek_NetworkRelease() call in dependency of the state of NmOsek
- Use different shutdown times for CanNm and NmOsek on the same channel

To ensure that only productive code is used verify that:

- If Nm Coordination is active in Nm and NmOsek is used in the configuration, then check that in NmOsek the configuration parameter "Wait Bus Sleep Extensions" is inactive

-Nm_Cfg.h contains the following line:

```
#define NM_WAIT_BUS_SLEEP_EXTENSIONS STD_OFF
```

Resolution Description:

ESCAN00091218 BETA version - the BSW module has a feature with BETA state (FEAT-371)

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 7.00.00

Fixed in versions: 8.03.00

Problem Description:

What is the impact of BETA software:

BETA software

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not or only partly tested and not all quality measures have taken place

Which functionality is BETA:

The following feature/function is in BETA state.

- The sender/receiver access to the application data.

To ensure that only productive code is used verify that:

- Parameter DcmDspDataUsePort is not one of: USE_DATA_SENDER_RECEIVER
- Parameter DcmDspDidUsePort is not one of: USE_ATOMIC_SENDER_RECEIVER_INTERFACE, USE_ATOMIC_NV_DATA_INTERFACE

Resolution Description:

-

ESCAN00091231 BETA version - the BSW module has a feature with BETA state (FEAT-1899)

Component@Subcomponent: Diag_Asr4Dcm@GenTool_GeneratorMsr

First affected version: 7.00.00

Fixed in versions: 8.03.00

Problem Description:

What is the impact of BETA software:

BETA software

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not or only partly tested and not all quality measures have taken place

Which functionality is BETA:

The following feature/function is in BETA state.

- Sender/Receiver Ports for NVM or complex types data.

To ensure that only productive code is used verify that:

- ECUC parameter /Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort == USE_DATA_ELEMENT_SPECIFIC_INTERFACES

Resolution Description:

-

ESCAN00092470 BETA version - the BSW module has a feature with BETA state (FEAT-1454)**Component@Subcomponent:** SysService_Asr4BswMcfg5@GenTool_GeneratorMsr**First affected version:** 10.00.00**Fixed in versions:****Problem Description:**

What is the impact of BETA software:

BETA software

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not or only partly tested and not all quality measures have taken place

Which functionality is BETA:

The following feature/function is in BETA state.

- Configuration of Switch Ports (Mode Request Port (BswM_EthIf_PortGroupLinkStateChg))

Additonal:

Currently the BswM general switch BswMEthIfEnabled is not set via a Auto-Validation. During fixing of this BETA ESCAN a validation has to be implemented which ensures that the BswMEthIfEnabled is true if the EthIf calls this API and if the Mode Request Port is configured in BswM.

Resolution Description:**ESCAN00092868 BETA version - the BSW module is in BETA state****Component@Subcomponent:** DrvCry_Rh850Icum@Implementation**First affected version:** 1.00.00**Fixed in versions:****Problem Description:**

What is the impact of BETA software:

BETA software

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not or only partly tested and not all quality measures have taken place

Which functionality is BETA:

The complete BSW module is in BETA state**Resolution Description:**

ESCAN00093797 BETA version - the BSW module has a feature with BETA state (Barriers)**Component@Subcomponent:** Os_CoreGen7@Implementation**First affected version:** 2.00.00**Fixed in versions:****Problem Description:**

What is the impact of BETA software:

BETA software

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not or only partly tested and not all quality measures have taken place

Which functionality is BETA:

The following feature/function is in BETA state.

- Using barriers to synchronize tasks is implemented as a BETA feature.

To ensure that only productive code is used verify that:

- No barriers (/MICROSAR/Os/OsBarrier) are configured in the configuration tool.

Resolution Description:

API Extensions:

No extension of the API.

API Changes:

No modification of the API.

Module handling changes:

No modification of the module handling.

For a detailed description of the API and the handling of the module refer to the Technical Reference.

ESCAN00093813 BETA version - the BSW module has a feature with BETA state (Fast Trusted Functions)

Component@Subcomponent: Os_CoreGen7@Implementation

First affected version: 2.00.00

Fixed in versions:

Problem Description:

What is the impact of BETA software:

BETA software

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not or only partly tested and not all quality measures have taken place

Which functionality is BETA:

The following feature/function is in BETA state.

- Os_CallFastTrustedFunction() API

To ensure that only productive code is used verify that:

- No elements are configured in /MICROSAR/Os/OsApplication/OsApplicationFastTrustedFunction

Resolution Description:

ESCAN00094381 BETA version - the BSW module has a feature with BETA state (FEAT-2160)

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 7.02.00

Fixed in versions: 8.03.00

Problem Description:

What is the impact of BETA software:

BETA software

- must not be used in productive projects as they may result in unpredictable ECU behavior
- may not provide all features intended for the productive project
- is not or only partly tested and not all quality measures have taken place

Which functionality is BETA:

The following feature/function is in BETA state.

- DEM API AR 4.3.0

To ensure that only productive code is used verify that:

- Switch "/Dcm/DcmConfigSet/DcmGeneral/DcmDemApiVersion" is not set to "DCM_DEM_API_4_03_00" in configuration tool.
- #define DCM_DEM_API_430_ENABLED in Dcm_Cfg.h is always STD_OFF

Resolution Description:

-

ESCAN00094537 BETA version - the BSW module has a feature with BETA state (FEAT-2084)	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	2.01.00
Problem Description:	
What is the impact of BETA software:	

BETA software	
<ul style="list-style-type: none"> - must not be used in productive projects as they may result in unpredictable ECU behavior - may not provide all features intended for the productive project - is not or only partly tested and not all quality measures have taken place 	
◆	
Which functionality is BETA:	

The following feature/function is in BETA state.	
<ul style="list-style-type: none"> - Multi-Core support in XCP 	
To ensure that only productive code is used verify that:	
<ul style="list-style-type: none"> - That no EcuC Hardware reference is configured in the Xcp (/MICROSAR/Xcp/XcpConfig/XcpCoreDefinition/XcpEcuCCoreDefinitionRef is not defined) - That the Xcp_Event function is only called from the BSW core. 	
Resolution Description:	
No workaround possible	

2.5 Apparent Issues

Apparent issues are detected immediately when using the software module. If an issue does not show up while working with the software module, the ECU project is not affected by the issue. Apparent issues may or may not have workarounds.

Index

ESCAN00073545	Final FBL response not cancelled on protocol preemption <small>Diag_Asr4Dcm@Implementation</small>
ESCAN00079399	Linker error: '<Cdd>_Transmit' : undeclared identifier (or '<Cdd_RxIndication>') <small>Cdd_AsrCddCfg5@Description</small>
ESCAN00087948	Update Bits are not cleared if Com_IpduGroupControl is called with initialize = false <small>IL_AsrComCfg5@Implementation</small>
ESCAN00087958	Wrong return value of GetTaskState when called from PostTaskHook <small>Os_CoreGen7@Implementation</small>
ESCAN00087977	Compiler error: PduR_Lcfg.c: 'PDUR_FCT_IPDUMTX' : undeclared identifier <small>Gw_AsrPduRCfg5@GenTool_GeneratorMsr</small>
ESCAN00089109	Software stack monitoring for non trusted functions not supported <small>Os_CoreGen7@Implementation</small>
ESCAN00089287	Dem APIs are incompatible to the application <small>Diag_Asr4Dem@GenTool_GeneratorMsr</small>
ESCAN00090666	Linker error caused by wrong memory section name <small>SysService_AsrCryFord@Implementation</small>
ESCAN00090998	Configuration tool reports Rte90005 exception because of java.lang.IllegalArgumentException <small>Rte_Asr4@GenTool_GeneratorMsr</small>
ESCAN00091118	EcuM causes a Rte Det error (RTE_E_DET_UNINIT) in the shutdown sequence while the Nvm write all is performed <small>SysService_Asr4EcuM@Implementation</small>
ESCAN00091322	Validation error message cannot be solved: Error at validator runtime Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModelView <small>Nm_Asr4NmIf@GenTool_GeneratorMsr</small>
ESCAN00092058	Inconsistent data types in interface DcmIf <small>Diag_Asr4Dem@GenTool_GeneratorMsr</small>
ESCAN00092245	TechRef: Integration of secret key is not correct <small>Diag_AsrSwcSecAccess_Ford@Doc_TechRef</small>
ESCAN00092505	The start address for CAN Message RAM only works for 64KByte alignment. <small>DrvCan_Mpc5700McanLI@GenTool_GeneratorMsr</small>
ESCAN00092569	Compiler error: identifier "pduInfo_var_id" is undefined <small>DrvCan_Mpc5700McanLI@Implementation</small>
ESCAN00092622	A change of the main function period does not lead to a rebuild of the SWC description <small>SysService_Asr4EcuM@GenTool_GeneratorMsr</small>
ESCAN00092718	<MSN>90005 - Generator (<Generator Name>) failure, because of an exception "exception in <Msn> generator during Validation encountered: java.lang.NullPointerException" <small>CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr</small>
ESCAN00092720	DataRenamer not working for MICROSAR Define block <small>CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr</small>
ESCAN00092955	Compiler error: incompatible types - from 'const <MSN>_PCConfigType *' to 'const <MSN>ConfigType *const' <small>SysService_Asr4EcuM@GenTool_GeneratorMsr</small>

Index

ESCAN00093294	Invalid key accepted due to inconsistent Csm and CryFord job processing configuration Diag_AsrSwcSecAccess_Ford@Implementation
ESCAN00093317	Value Calculations does not act as expected CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr
ESCAN00093405	Auto Configuration - Invalid multiplicity after manual adaptations of container BswMAvailableActions SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00093449	A2L compu method RTE_CM_BOOLEAN cannot be used to calibrate boolean values Rte_Core@Implementation
ESCAN00093502	Technical Reference: Wrong API description for Csm_SymKeyExtractStart SysService_AsrCsm@Doc_TechRef
ESCAN00093634	CAN-FD format (Bosch V1.0, ISO-11898) inconsistent DrvCan_Mpc5700McanLI@GenTool_GeneratorMsr
ESCAN00093839	CFG5 Exception or Compile Error "Too many initializer values" CommonAsr_ComStackLib@GenTool_GeneratorMsr
ESCAN00094259	Auto-Configuration Communication Control shows an error in case of not available module Com SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00094298	The Ecu does not startup properly in some MultiCore configurations SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00094319	Auto-Configuration Communication Control: Init Mode of Lin Schedule Indication is missing SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00094355	[Error] CANIF10027 None CAN-channel has multiple BasicCAN Tx-objects. Hence the feature "CanIfMultipleBasicCANTxObjects" is not required in current configuration and must be disabled. If_AsrIfCan@GenTool_GeneratorMsr
ESCAN00094416	Linker error: undefined reference to ComM_Nm_PreparesBusSleepMode Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr
ESCAN00094506	API CanIf_CheckBaudrate is not described / the API CanIf_ChangeBaudrate is described twice If_AsrIfCan@Doc_TechRef
ESCAN00094541	Auto-Configuration Communication Control: Rules without expressions are created and so validation errors are shown SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00094612	WdgM_GetTickCount is called with suspended interrupts SysService_Asr4WdM@Implementation
ESCAN00094806	Compiler error: Undefined symbol RTE_MODE_DcmDiagnosticSessionControl_DEFAULT_SESSION Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00094875	Compiler error: dld.exe: warning: Undefined symbol 'MemIf_*_WriteWrapper' in file 'obj/MemIf_Cfg.o' If_AsrIfMem@GenTool_GeneratorMsr
ESCAN00094883	Improper workaround for MCAN Erratum #10 DrvCan_Mpc5700McanLI@Implementation
ESCAN00094972	Compiler error: Missing declaration for APIs Dcm_OptimizedSetNegResponse and Dcm_OptimizedProcessingDone Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00094980	Generator version check fails Tp_Asr4TpCan@ElisaPlugin
ESCAN00094991	Generated SW-C template for S/R data interface does not comply with AR 4.x standard Diag_Asr4Dcm@GenTool_GeneratorMsr

Index

ESCAN00095065	Compiler error: Missing declaration for API BswM_Dcm_RequestResetMode() Diag_Asr4Dcm@Implementation
ESCAN00095083	Compiler error: #20: identifier "EcuM_WakeupSourceType" is undefined DrvTrans__baseCanxAsr@GenTool_GeneratorMsr
ESCAN00095259	Compiler error: WdgIf uses undefined memory sections If_Asr4IfWd@GenTool_GeneratorMsr
ESCAN00095262	Missing record layout for uint64 and sint64 data types Rte_Core@Implementation
ESCAN00095296	Validation error: Reference to XcpOnCan/XcpOnCanVariableDlc not found Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00095310	Compiler error: identifier EcuM_GlobalConfigRoot not declared SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00095312	Generation cannot be started because of an error COMM90500 - A generated value is not in range of the specified datatype Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr
ESCAN00095342	Compiler error: identifier PduR Routing Manager APIs not declared Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00095432	RTE generator generates too many compu scales Rte_Core@Implementation
ESCAN00095490	Compiler error: Cannot open include file: 'Det.h' Diag_Asr4Dcm@Implementation
ESCAN00095519	ConsistencyRT00002 Error at validator runtime: CanSMBorTxConfPollingValidator if CanIf is missing Ccl_Asr4SmCan@GenTool_GeneratorMsr
ESCAN00095528	Compiler error: Undeclared Identifier 'COM_UINT8_APPLTYPEOFRXACCESSINFO' IL_AsrComCfg5@Implementation
ESCAN00095553	Compiler error: Undefined identifier *PtrType to VARs of simple types with <MSN>_USE_INIT_POINTER to STD_ON CommonAsr_ComStackLib@GenTool_GeneratorMsr
ESCAN00095570	Auto-Configuration Communication Control: Activation of a PNC on one channel does also affect other channels with same PNC SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00095571	EcuM causes a Rte warning about a not existing mode request type map SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00095591	Compiler error: Invalid suffix 'U' on floating constant Rte_Core@Implementation
ESCAN00095642	Compiler error: Service 0x2A is enabled, but no periodic messages have been configured for Dcm Diag_Asr4Dcm@Doc_TechRef
ESCAN00095660	[Applies only if hardware: Tle9252 is used] BETA version - the BSW module is in BETA state DrvTrans_Tja1043CandioAsr@Implementation
ESCAN00095684	esl_decryptAES* causes memory corruption SysService_CryptoCv@Impl_ESLib
ESCAN00095690	RTE Analyzer fails due to duplicated runnable functions Rte_Core@Implementation
ESCAN00095730	COM02300: Proposed ComBitSize may exceed allowed range for UINT8_N and UINT8_DYN signals IL_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00095747	Generator error for duplicate runnable symbols not triggered Rte_Core@Implementation
ESCAN00095769	Validation Error : ConsistencyRT00002 - COM90008 - Model access request failure in ComSignalTimeoutValidatorAs4 IL_AsrComCfg5@GenTool_GeneratorMsr

Index

ESCAN00095785	RTE49999 error message: "Invalid data type mapping" for mapped DataType in CalibrationPort PortInterfaces Rte_Core@Implementation
ESCAN00095786	Wrong generated limits for CharacteristicTables in Rte.a2l when physical constraints are configured Rte_Core@Implementation
ESCAN00095808	RTE49999 when mode disablings are used in a task with schedule tables Rte_Core@Implementation
ESCAN00095840	[Only in case of using a PN-CanTrcv] ConsistencyRT00002 - Error at validator runtime: CanTrcvPnConfigurationValidator DrvTrans__baseCanxAsr@GenTool_GeneratorMsr
ESCAN00095854	Configuration tool reports RTE90005 java.lang.NullPointerException during validation phase Rte_Asr4@GenTool_GeneratorMsr
ESCAN00095891	Auto-Configuration - SDLC: Wizard leads to validation errors in case of unconfigured channels SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00095900	RTE Analyzer reports false out of bounds access GenTool_IRAnalyzer@Application
ESCAN00095937	RTE49999 when a curve use a value data type with texttable compu method Rte_Asr4@Generator
ESCAN00095940	Missing check for UINT8_DYN signals when creating the Gw-Routing Key Il_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00095942	Undefined a2l data type in record layout Rte_Core@Implementation
ESCAN00095944	Missing compu method in A2L Rte_Core@Implementation
ESCAN00096004	Compiler error: NON RTE USE CASE - Redeclared typedef Csm_ReturnType SysService_AsrCsm@Implementation
ESCAN00096021	[Error] ConsistencyRT00002 - Error at validator runtime: CanIfTxBufferSupportValidator If_AsrIfCan@GenTool_GeneratorMsr
ESCAN00096024	Incorrect parameter in API Dem_PostRunRequested Diag_Asr4Dem@Doc_TechRef
ESCAN00096026	RteAnalyzer reports incompatible pointer type passing for inter-runnable variables with multi-dimensional arrays Rte_Core@Implementation
ESCAN00096061	Auto-Configuration - SDLC: Support of CAN FD PDUs is not working SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00096126	Com_Init may lead to long interrupt locks Il_AsrComCfg5@Implementation
ESCAN00096128	Generator Exception in case the PduRSrcPdu global Pdu is referenced by more than one other container Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00096129	MEMMAP_SW_MINOR_VERSION / MEM_SW_MINOR_VERSION is not correct CommonAsr__Common@Impl__MemMap
ESCAN00096177	RTE49999 when no trigger is assigned to a server operation Rte_Core@Implementation
ESCAN00096187	RTE49999 when the base type on network representation has no size Rte_Core@Implementation
ESCAN00096299	During generation of code the error occurs: "CANTRCV01009 No valid wakeup source specified." DrvTrans__baseCanxAsr@GenTool_GeneratorMsr

Index

ESCAN00096311	Validation message "CANIF30001 At least one BasicCAN Tx-PDU is configured. Hence it is recommended to enable the Tx-buffer..." leads to invalid configuration If_AsrIfCan@GenTool_GeneratorMsr
ESCAN00096335	Compiler error: Unknown identifier Dcm_DemApiNrcMapSelectDTC Diag_Asr4Dcm@Implementation
ESCAN00096391	Compiler error: function "CanHL_WakeupProcessed" / "CanHL_SleepProcessed" was referenced but not defined DrvCan__coreAsr@Implementation
ESCAN00096422	Validation rule inhibits generation in an valid (special) use case SysService_Asr4WdM@GenTool_GeneratorMsr
ESCAN00096516	Compiler error: Wrong generated Rte_IocSend calls for queued communication Rte_Core@Implementation
ESCAN00096521	Compile error on Unix systems due to upper case in include file SysService_CryptoCv@Impl_actCLib
ESCAN00096532	Compiler error: undeclared identifier "WDGM_GLOBAL_STATUS_xxx" SysService_Asr4WdM@GenTool_GeneratorMsr
ESCAN00096559	Wrong signal type for data conversion Rte_Core@Implementation
ESCAN00096581	PduRTxBuffer references are incorrectly validated for transport protocol 1:N/N: 1 routing paths with API forwarding PduRDestPdu Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00096582	Non-interactive Mode fails _3rdParty_McalIntegration_Helper@VectorIntegration
ESCAN00096594	C Standard Library is used SysService_CryptoCv@Impl_actCLib
ESCAN00096615	RTE Analyzer fails due to duplicated runnable functions Rte_Core@Implementation
ESCAN00096629	Linker error: unresolved symbol error for not existing callout function referenced in Det_Cfg.o in case of disabled DET SysService_AsrDet@GenTool_GeneratorMsr
ESCAN00096748	Nonexistent "TimerSTMin" struct member in a2I file Tp_Asr4TpCan@GenTool_GeneratorMsr
ESCAN00096774	Compiler error: Duplicated variable definitions in analyzer stubs Rte_Core@Implementation
ESCAN00096900	Compiler error: identifier EcuM_Get<***> not declared SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00096969	Misleading function return code description for API Dcm_GetTesterSourceAddress Diag_Asr4Dcm@Doc_TechRef
ESCAN00096982	AssertionError: The getMaxUnsignedValueForNumBytes utility allows only up to 4 bytes! Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00097053	Compiler error: Empty struct DCM_DIDMGROPTYPEPEREADCONTEXTTYPE_TAG Diag_Asr4Dcm@Implementation
ESCAN00097056	Compiler error: 'Offset' : is not a member of 'DCM_DIDMGROPTYPEPEREADCONTEXTTYPE_TAG' Diag_Asr4Dcm@Implementation
ESCAN00097073	Sub-chapter "ResponseOnEvent Specifics" seems to be part of chapter "Handling with DID Ranges" Diag_Asr4Dcm@Doc_TechRef
ESCAN00097086	Compiler error: Undefined reference to 'Com_GetTxFilterInitValueArrayBasedFilterInitValueLengthOfTxSigInfo' Il_AsrComCfg5@Implementation

Index

ESCAN00097087	Null pointer exception when a data element is missing Rte_Asr4@GenTool_GeneratorMsr
ESCAN00097092	Compiler error: Identifier "sint"<X> is undefined Il_AsrComCfg5@Implementation
ESCAN00097096	Compiler error: incompatible pointer type / loss of volatile qualifier in pointer cast DrvCan__coreAsr@Implementation
ESCAN00097143	WdgMLocalStateChangeCbK not created for all Supervised Entities SysService_Asr4WdM@GenTool_GeneratorMsr
ESCAN00097148	WdgMGlobalStateChangeCbK / WdgMLocalStateChangeCbK function prototype generated with incompatible signature compared to RTE SysService_Asr4WdM@GenTool_GeneratorMsr
ESCAN00097151	Incomplete Mirror Data DrvCan_Mpc5700McanLI@Implementation
ESCAN00097168	EcuM debug data cannot be found in the map file SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00097174	WdgM_UpdateTickCount is declared not as "void" in SWC file SysService_Asr4WdM@GenTool_GeneratorMsr
ESCAN00097203	Compiler error: "conversion of data types, possible loss of data" in case of large buffers Diag_Asr4Dcm@Implementation
ESCAN00097240	CanIf debug data cannot be found in the map file If_AsrIfCan@GenTool_GeneratorMsr
ESCAN00097257	Compiler error: error C2065: 'CanNm_CancelTransmit' : undeclared identifier Nm_Asr4NmCan@Implementation
ESCAN00097274	Compiler error: Incompatible Rte_MemCpy prototypes Rte_Core@Implementation
ESCAN00097291	Compiler error: Use of undeclared identifier Rte_Appl_AckFlags Rte_Core@Implementation
ESCAN00097298	Mismatching data type Dem_DTCTOriginType Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00097303	Compiler error: Call to job finished runnable misses parameters Rte_Core@Implementation
ESCAN00097341	Mode Transition Value of Mode Declaration Group WdgM_Mode is set to 255 SysService_Asr4WdM@GenTool_GeneratorMsr
ESCAN00097355	Auto-Configuration Ecu State Handling: Self run request timeout value is not shown correct in case of 0 SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00097457	Matrix dimensions are swapped for two dimensional arrays in A2L file Rte_Core@Implementation
ESCAN00097476	RTE01004 error during contract phase generation (Could not read back DVCfgRteGen data) Rte_Core@Implementation
ESCAN00097477	Code generation is not possible due to error RTE13068 - Insufficient data type to represent mode value Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr
ESCAN00097525	RTE49999 when transformers are not configured for all fan-out signals Rte_Core@Implementation
ESCAN00097531	a2I: In Variant use case only the first variant is generated for the bus specific a2I files Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00097649	Compiler error: Rte_Write writes a variable that does not exist Rte_Core@Implementation
ESCAN00097683	A generated value is not in range of the specified datatype Il_AsrComCfg5@GenTool_GeneratorMsr

Index

ESCAN00097684	Warning RTE49999 when XcpEvent support is enabled Rte_Core@Implementation
ESCAN00097802	Activation reason data type uses bit access Rte_Core@Implementation
ESCAN00097876	Generated data streams toggle with each code generation if <MSN>ReduceDataByStreaming is enabled SysService_Asr4BswMCFG5@GenTool_GeneratorMsr
ESCAN00097910	Dcm_Swc.arxml: Missing values of Mode-Declarations in Mode-Declaration-Groups Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00097950	Compiler error: 'CanTp_GetRxSduCfgInd' undefined Tp_Asr4TpCan@Implementation
ESCAN00097971	RTE49999: Mismatching constant values Rte_Asr4@Generator
ESCAN00098057	Generated data streams toggle with each code generation if <MSN>ReduceDataByStreaming is enabled IL_AsrComCFG5@GenTool_GeneratorMsr
ESCAN00098062	RTE49999: When InitializeImplicitBuffers is configured for implicit connections to NvBlock SWCs Rte_Core@Implementation
ESCAN00098068	Null pointer exception when a service dependency contains an invalid pim reference Rte_Asr4@GenTool_GeneratorMsr
ESCAN00098104	RTE Analyzer reports false out of bound accesses GenTool_IRAnalyzer@Application
ESCAN00098155	Inconsistencies of Technical Reference regarding Dem usage SysService_Asr4WdM@Doc_TechRef
ESCAN00098167	RTE01081 Model object </MICROSAR/IoHwAb_swc/ComponentTypes/IoHwAb> of command line parameter -m is invalid. Rte_Asr4@GenTool_GeneratorMsr
ESCAN00098187	RTE generator generates wrong compu method in case data type names are not unique Rte_Core@Implementation
ESCAN00098204	RTE01060: Could not read Rte_Needs.ecuc.arxml when exclusive areas with implementation method resource are accessed from multiple partitions Rte_Core@Implementation
ESCAN00098260	Erroneous validation message "CanIfMultipleBasicCANTxObjects is not required" If_AsrIfCan@GenTool_GeneratorMsr
ESCAN00098424	a2I: OPTIONAL_CMD GET_DAQ_EVENT_INFO generated unconditionally. Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00098469	Unused Interrupt enabled DrvCan_Mpc5700McanLI@Implementation
ESCAN00098487	PDUR_EXCLUSIVE_AREA_1 is created by tool but not used in embedded code Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00098523	GeneralDiagnosticInfo interface is named "GeneralEventInfo" instead of "GeneralEvtInfo" Diag_Asr4Dem@GenTool_GeneratorMsr
ESCAN00098583	Generator Error Message ""XCP90110 Undefined DefinitionRef for Parameter." - misleading problem indication Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00098584	NvM NVM01036 validation does not clearly describe the problem MemService_AsrNvM@GenTool_GeneratorMsr

Index

ESCAN00098599	RTE49999 when a data element without init value is connected to a data element without port accesses Rte_Asr4@Generator
ESCAN00098646	RTE generator creates NvMRomBlockDataAddress with ROM variables that do not exist Rte_Core@Implementation
ESCAN00098679	Compiler error: incompatible declaration of ComM_ConfigPtr Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr
ESCAN00098712	Linker error: DemTriggerEventDataChangedCallback==FALSE and DemTriggerEventStatusChangedCallback==FALSE will only suppress the creation/usage of SWC port interfaces, but not the underlying function calls Diag_Asr4Dem@GenTool_GeneratorMsr
ESCAN00098820	RTE generator reports unexpected exit code Rte_Core@Implementation
ESCAN00098822	RTE Generator reports unexpected exit code Rte_Analyzer@Application
ESCAN00098865	RTE49999 when sender and receiver use different init value constants with the same values Rte_Core@Implementation
ESCAN00098866	Service 0x3E: Misleading caution box regarding external service implementation Diag_Asr4Dcm@Doc_TechRef
ESCAN00098883	/MICROSAR/Xcp/XcpGeneral/XcpTimestampTicks limited in range to uint8 Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00098887	Wrong linker section used DrvCan_Mpc5700McanLI@Implementation
ESCAN00098893	Compiler error: Missing Rte_MemClr when activation reason is used in a system with OS applications Rte_Core@Implementation
ESCAN00098918	Compiler error: Duplicated implicit APIs in analyzer stubs when the same port access is declared twice Rte_Core@Implementation
ESCAN00098922	NmStateChangeCallback is not called if same states are passed in Nm_StateChangeNotification Nm_Asr4NmCan@Implementation
ESCAN00098939	RTE49999 when application data type with texttable compu method is mapped to float implementation data type Rte_Core@Implementation
ESCAN00098955	RTE49999 when a compu method declared CompuScales that would result in the same symbol Rte_Asr4@Generator
ESCAN00098966	Missing reference about the cluster index for the Nm Gateway Coordination Extension Nm_Asr4NmIf@Doc_TechRef
ESCAN00099018	NPE during generation with invalid ComRxDataTimeoutSubstitutionValue Il_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00099049	RTE49999: when task type is set to basic and cyclic trigger implementation is set to none Rte_Core@Implementation
ESCAN00099057	EcuM Wakeup Source defines are generated multiple times with numerical postfix in case of variance SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00099105	Compiler error: Rte.c accesses Rte_ActivationVector variable that is declared static in Rte_<osappl>.c Rte_Core@Implementation

Index

ESCAN00099156	Missing description of OS_VTH_FORCED_TERMINATION and OS_VTHP_THREAD_CLEANUP Os_CoreGen7@Doc_TechRef
ESCAN00099160	Patch action fails because file path is too long _3rdParty_McalIntegration_Helper@VectorIntegration
ESCAN00099169	Compiler error/warning: Unreferenced formal parameter watchdog in actX25519.c SysService_CryptoCv@Impl_actCLib
ESCAN00099179	Compiler error: MemMap_Common.h: Wrong pragma command / unknown memory section used Tp_Asr4TpCan@Implementation
ESCAN00099186	Compiler error: Inconsistent setting for number of channels; with dynamic channel assignment, more SDUs than channels are expected Tp_Asr4TpCan@Implementation
ESCAN00099189	a2I: Calculation of CAN-FD parameter SECONDARY_SAMPLE_POINT in CanXcp.a2I is incorrect Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00099274	Null pointer exception when data mapping exists in all variants but signal group does not Rte_Asr4@GenTool_GeneratorMsr
ESCAN00099313	Test case 109 of WdgM Verifier does not fail anymore SysService_Asr4WdM@root
ESCAN00099318	Test case 107 of WdgM Verifier is marked as NOT PASSED SysService_Asr4WdM@root
ESCAN00099345	Exception in Generator when XcpCalibration container is not present Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00099398	Compiler error: Incorrect expansion of Com_ReceiveShadowSignal with COM_RECEIVE_SIGNAL_MACRO_API IL_AsrComCfg5@Implementation
ESCAN00099413	Compiler error: Duplicated variable definition in case of N:M communication with external and internal receivers Rte_Core@Implementation
ESCAN00099473	The value <X> is not in the range of the specified datatype UINT_16 IL_AsrComCfg5@GenTool_GeneratorMsr
ESCAN00099474	a2I: Parameter XCP_MAX_ODT_ENTRY_SIZE fixed to 7 Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00099518	CanXcp.a2I not variant specific Cp_Asr4Xcp@GenTool_GeneratorMsr
ESCAN00099525	CanTpEnableSynchronousTransmit cannot be used with non MICROSAR Components Tp_Asr4TpCan@Doc_TechRef
ESCAN00099526	CanTpEnableSynchronousTransmit cannot be used with non MICROSAR Components Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00099539	RTE49999: N:1 Inter-Partition Client-Server Communication with IOCs Rte_Core@Implementation
ESCAN00099548	InitMonitorReason DEM_INIT_MONITOR_REENABLED is missing in callback description Diag_Asr4Dem@Doc_TechRef
ESCAN00099553	State diagram of the EcuM with fixed state machine shows call of EcuM_AL_DriverRestart in the wrong transition. SysService_Asr4EcuM@Doc_TechRef
ESCAN00099582	Compiler error: actAES.h:23 missing argument for macro P2FUNC SysService_CryptoCv@Impl_actCLib

Index

ESCAN00099596	Compiler error: Missing dataSrq variables Rte_Core@Implementation
ESCAN00099599	Dem_SetOperationCycleState can not be called in the pre-initialized mode Diag_Asr4Dem@Doc_TechRef
ESCAN00099644	Compiler error: QOverflowType struct without declaration Rte_Core@Implementation
ESCAN00099667	Null pointer exception when no BswImplementation for LDCOM exists Rte_Asr4@GenTool_GeneratorMsr
ESCAN00099693	Compiler error: Incompatible Declaration in Rte_Csm.h and Csm.h SysService_AsrCsm@Implementation
ESCAN00099714	Compiler error/warning: argument type of string of Ioc_Write does not match prototype for implicit write Rte_Core@Implementation
ESCAN00099814	Wrong references to CanTp_Cfg.c exist Tp_Asr4TpCan@Doc_TechRef
ESCAN00099816	Compiler error: Missing buffer definition within struct Rte_tsRB Rte_Core@Implementation
ESCAN00099948	Compiler error: Duplicated lower limit variable in RTE Analyzer stubs Rte_Core@Implementation
ESCAN00099951	Compiler error: CanNm_SetMsgRequest undefined Nm_Asr4NmCan@Implementation
ESCAN00099953	Compiler error: Too many struct initializers when InitializeImplicitBuffers is configured Rte_Core@Implementation
ESCAN00099959	Compiler error: undefined reference to 'CanTp_IsTxSduCfgIndUsedOfRxPduMap' Tp_Asr4TpCan@Implementation
ESCAN00099970	Wrong error message for ScheduleTable ExpiryPoint offset Os_CoreGen7@GenTool_GeneratorMsr
ESCAN00099987	RTE49999: when union is used for N:1 connections or PR ports Rte_Core@Implementation
ESCAN00099988	Description: Lower Limit for parameter DemExtendedDataRecordNumber wrong Diag_Asr4Dem@Description
ESCAN00100169	Parameter description of CheckRemoteSleepIndication is incorrect Nm_Asr4NmCan@Doc_TechRef
ESCAN00100193	Improve description AN-ISC-8-1184_Compiler_Warnings@Doc_ApplicationNote
ESCAN00100240	RTE generator creates duplicated A2L group objects in case of PR Ports Rte_Core@Implementation

ESCAN00073545 Final FBL response not cancelled on protocol preemption	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.05.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

The ECU will process the FBL final response even if there is higher protocol request sent.	
When does this happen:	

When immediately after reprogramming of the ECU has ended, the very first request after ECU powers on in the application is a hi-priority one (i.e. OBD).	
In which configuration does this happen:	

- Any configuration where the ECU shall be able to send a final response without request after reset.	
AND	
- Protocol prioritisation is to be supported (i.e. OBD vs. UDS).	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00079399 Linker error: '<Cdd>_Transmit' : undeclared identifier (or '<Cdd_RxIndication>')

Component@Subcomponent: Cdd_AsrCddCfg5@Description

First affected version: 2.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Linker error in PduR_Lcfg.c: '<Cdd>_Transmit' : undeclared identifier

Linker error in PduR_Lcfg.c: '<Cdd>_RxIndication' : undeclared identifier

The Cdd_AsrCddCfg5 is not derived according to the ASR 4.0.3 rules and allows a LOWER-MULTIPLICITY of 0 for the CddPduRLowerLayerRxPdu and CddPduRLowerLayerTxPdu instead of the LOWER-MULTIPLICITY of 1.

The generic ASR PduR according to the ASR 4.0.3 Specification has no information to deactivate a communication direction (e.g. a Parameter in the PduRBSwModules).

When does this happen:

The error is issued by the linker after compilation of the code in case the configuration is as described below.

In which configuration does this happen:

Rx only Cdd with a CddPduRLowerLayerContribution (just receive pathways exits)

The <CddName>.h file contains the following define:

<CddName>_LOWERLAYERCOMIF_TX is defined to STD_OFF

OR

Tx only Cdd with a CddPduRLowerLayerContribution (just transmit pathways exits)

The <CddName>.h file contains the following define:

<CddName>_LOWERLAYERCOMIF_RX is defined to STD_OFF

Resolution Description:

Workaround:

Implement the not required '<Cdd>_Transmit' API on your own in a c and h file of your choice and add the header file with a user config file to the PduR configuration that the compiler does not throw a warning.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00087948		Update Bits are not cleared if Com_IpduGroupControl is called with initialize = false
Component@Subcomponent:	Il_AsrComCfg5@Implementation	
First affected version:	1.00.00	
Fixed in versions:	15.00.00	
Problem Description:		
What happens (symptoms):		

After a IpduGroup is started with initialize = false a Signal is transmitted with set Update Bit although signal was not updated since the IpduGroup was stopped.		
When does this happen:		

during the call of Com_IpduGroupControl or Com_IpduGroupStart.		
In which configuration does this happen:		

If Tx UpdateBits are used		
AND		
if Com_IpduGroupControl/ Com_IpduGroupStart is used with initialize = false		
Resolution Description:		
Workaround:		

No workaround available.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00087958 Wrong return value of GetTaskState when called from PostTaskHook	
Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms): ----- GetTaskState returns SUSPENDED for current task when called from PostTaskHook. Return 'RUNNING' instead.	
When does this happen: ----- In PostTaskHook the task is still running.	
In which configuration does this happen: ----- Configuration invariant.	
Resolution Description:	
Workaround: ----- Do not use the API GetTaskState for the current task in the PostTaskHook.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

**ESCAN00087977 Compiler error: PduR_Lcfg.c:
'PDUR_FCT_IPDUMTX' : undeclared identifier****Component@Subcomponent:** Gw_AsrPduRCfg5@GenTool_GeneratorMsr**First affected version:** 2.03.00**Fixed in versions:** 12.00.00**Problem Description:**

What happens (symptoms):

Compiler error: PduR_Lcfg.c: 'PDUR_FCT_IPDUMTX' : undeclared identifier

Hint: If a module without routing paths is configured the validation can not determine the communication type.

Ensure that the /MICROSAR/PduR/PduRBswModules Parameter are configured suitable to the post- build scenario.

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

/ActiveEcuC/PduR/PduRGeneral[PduRDevErrorDetect] == true AND
Impl. Config Variant == VARIANT-POST-BUILD-LOADABLE AND
/ActiveEcuC/PduR/IpduM exists (/MICROSAR/PduR/PduRBswModules) AND
No routing path for IpduM exists AND the /ActiveEcuC/PduR/IpduM[PduRCommunicationInterface]
== false**Resolution Description:**

Workaround:

The communication type must be configured for BSW modules without routing paths.

Configure /ActiveEcuC/PduR/IpduM[PduRCommunicationInterface] to "true"

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00089109 Software stack monitoring for non trusted functions not supported	
Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

Stack monitoring in software is not supported for non trusted functions.	
When does this happen:	

Always	
In which configuration does this happen:	

In systems where non trusted functions and software stack checks are used:	
In configuration:	
/ActiveEcuC/Os/<Application>/OsApplicationNonTrustedFunction is used and /ActiveEcuC/Os/OsOS[OsStackMonitoring] = TRUE	
Resolution Description:	
Workaround:	

If a MPU is available, protect stacks by MPU.	
In case that no MPU is available, user has to ensure that stack overflow does not occur during execution of non trusted functions. Otherwise non trusted functions shall not be used.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00089287 Dem APIs are incompatible to the application	
Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	9.00.00
Problem Description:	
What happens (symptoms):	

Application interfaces cannot be connected to the Dem ports because of incompatible type definitions.	
The Dem APIs use Dem_ExtendedStatusByteType which uses an incompatible compu-method compared to Dem_UdsStatusType.	
When does this happen:	

Always when trying to connect an application software to the Dem	
In which configuration does this happen:	

Applications using port definitions according to Autosar 4.1.2	
Resolution Description:	
Workaround:	

Change the datatype used in the application SWC to enumeration instead of bitfield.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00090666 Linker error caused by wrong memory section name	
Component@Subcomponent:	SysService_AsrCryFord@Implementation
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

A linker error occurs due to a missing memory section. E.g. OsAppBswNonTrusted	
When does this happen:	

If a special memory mapping configuration in MemMap.h was used to build the library. E.g. if memory protection is used and the BSW is mapped in the memory section of a special OS application.	
In which configuration does this happen:	

If the customer uses a different name for the related memory sections or OS applications.	
Resolution Description:	
Workaround:	

Renaming the section due to the Vector Configuration	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00090998 Configuration tool reports Rte90005 exception because of java.lang.IllegalArgumentException	
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.08.00
Fixed in versions:	4.16.00
Problem Description: What happens (symptoms): ----- The configuration tool reports Rte90005 - Generator (MICROSAR RTE Generator) failure, because of an exception - Exception in Rte generator during Generation encountered: java.lang.IllegalArgumentException When does this happen: ----- This happen during generation phase. In which configuration does this happen: ----- This can happen sometimes in configurations that contain RTE errors found in calculation or validation phase.	
Resolution Description: Workaround: ----- Solving the reported RTE errors messages. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00091118 EcuM causes a Rte Det error (RTE_E_DET_UNINIT) in the shutdown sequence while the Nvm write all is performed

Component@Subcomponent: SysService_Asr4EcuM@Implementation

First affected version: 3.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

The Rte throws a Det error with the ID RTE_E_DET_UNINIT during the shutdown sequence.

When does this happen:

Always during the NvM_WriteAll() is performed.

In which configuration does this happen:

Only in configurations with all the following parameters are set to true:

/ActiveEcuC/EcuM/EcuMGeneral/EcuMEnableFixBehavior
/ActiveEcuC/EcuM/EcuMFixedGeneral/EcuMModeSwitchRteAck
/ActiveEcuC/EcuM/EcuMFixedGeneral/EcuMIncludeNvramMgr
/ActiveEcuC/Rte/RteGeneration/RteDevErrorDetect

Resolution Description:

Workaround:

The only workaround is to filter this DET message.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00091322	Validation error message cannot be solved: Error at validator runtime Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModel
Component@Subcomponent:	Nm_Asr4NmIf@GenTool_GeneratorMsr
First affected version:	9.00.00
Fixed in versions:	
Problem Description:	

ESCAN00091322 Validation error message cannot be solved: Error at validator runtime Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModel

What happens (symptoms):

The following validation error message appears in the Validation view in DaVinci Configurator:

ConsistencyRT00002 Error at validator runtime (1 message)

ConsistencyRT00002 Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModelView

This is not a configuration problem but an internal implementation error. Please contact Vector for support.

Validator-Class:

com.vector.cfg.gen.Nm_Asr4NmIf.validation.NmGlobalCoordinatorTimeAllNmOsekInNormalValidator

Validator-Description: NmGlobalCoordinatorTimeAllNmOsekInNormalValidator validates that the setting NmGlobalCoordinatorTimeAllNmOsekInNormal is defined if it required.

Further runtime errors of this validator won't be reported in the UI.

Ex: com.vector.cfg.gen.core.moduleinterface.exceptions.ValidationFailedException: [Error]

NM01003 - A Specific Channel configuration is missing for the NmChannelConfig

- The corresponding CanNmChannelConfig is missing for this NmChannelConfig

We are sorry, but due to this internal error, code generation of /[ANY]/CanNm, /MICROSAR/NmOsek, /[ANY]/FrIf, /[ANY]/FrNm, /[ANY]/UdpNm, /[ANY]/ComM, /MICROSAR/Nm has to be blocked. As a workaround, you may try to restart DaVinci Configurator. Otherwise, please call Vector for support

/ActiveEcuC/ComM

FrIf

UdpNm

CanNm

/ActiveEcuC/Nm

FrNm

/ActiveEcuC/NmOsek

Apparently, the message cannot be resolved.

When does this happen:

During configuration with DaVinci Configurator.

In which configuration does this happen:

Any configuration with Nm where a NmChannelConfig container exists that does not have a correspondent BusNmChannelConfig container (e.g. CanNmChannelConfig, FrNmChannel, LinNmChannelConfig, UdpNmChannelConfig, J1939NmChannelConfig, NmOsekChannelConfig, ...)

AND

(
'Coordinator Support Enabled' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/
NmCoordinatorSupportEnabled) is turned OFF in the NmGlobalFeatures container in Nm in the
'Network Management General' / 'Basic Editor' in DaVinci Configurator-> Nm_Cfg.h contains
#define NM_COORDINATOR_SUPPORT_ENABLED STD_OFF)

AND/OR

('Wait Bus Sleep Extensions' (/MICROSAR/NmOsek/NmOsekGlobalConfig/
NmOsekWaitBusSleepExtensions) is turned OFF or not defined or cannot be found in the

ESCAN00091322

Validation error message cannot be solved: Error at validator runtime Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModel

NmOsekGlobalConfig container in NmOsek in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator -> NmOsek_Cfg.h does not contain #define NMOSEK_WAIT_BUS_SLEEP_EXTENSIONS)

AND/OR

('Synchronizing Network' (/MICROSAR/Nm/NmChannelConfig/NmSynchronizingNetwork) is turned OFF for at least one NmChannelConfig container in Nm in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator)

AND/OR

('Coord Cluster Index' (/MICROSAR/Nm/NmChannelConfig/NmCoordClusterIndex) is undefined or set to 255 for at least one NmChannelConfig container in Nm in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator)

)

Please note that this is an invalid configuration because either the NmChannelConfig container without a BusNmChannelConfig must be deleted or the corresponding BusNmChannelConfig container must be created.

Resolution Description:

Workaround:

In DaVinci Configurator:

1) Create the corresponding BusNmChannelConfig container and configure its parameters and sub-containers.

OR

2) Delete the NmChannelConfig container that lacks a corresponding BusNmChannelConfig container.

Afterwards (no matter whether 1) or 2) has been applied), save the configuration, close it and re-open it.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00092058 Inconsistent data types in interface DcmIf	
Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	10.00.00
Problem Description:	
What happens (symptoms):	

SWC-Validation of Dcm and Dem generates inconsistent data types in interface DcmIf. Limits of the corresponding enumeration data types do not match.	
When does this happen:	

At Dem/Dcm SWC template import time in DaVinci Developer.	
In which configuration does this happen:	

/Dem/DemGeneral/DemDcmSupport = True within an AR 3.x RTE.	
Resolution Description:	
Workaround:	

Import first the Dem_Swc.arxml file, then the Dcm_Swc.arxml file to override the Dem data types (imported through Dcm).	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00092245 TechRef: Integration of secret key is not correct	
Component@Subcomponent:	Diag_AsrSwcSecAccess_Ford@Doc_TechRef
First affected version:	1.00.00
Fixed in versions:	2.00.00
Problem Description:	
What happens (symptoms):	

The Technical Reference is slightly outdated. In Chapter 4.1.1 the TechRef states out: "SwcSecAccessFord_Cfg.c In this file the secret keys can be stored"	
This is no longer correct since the implementation version 1.00.00. The key data are now fetched by using the DCM service "GetSecurityLevelFixedBytes".	
When does this happen:	

Implementation version > 1.00.00	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00092505 The start address for CAN Message RAM only works for 64KByte alignment.	
Component@Subcomponent:	DrvCan_Mpc5700McanLI@GenTool_GeneratorMsr
First affected version:	3.00.01
Fixed in versions:	3.03.01
Problem Description:	
What happens (symptoms):	

No CAN communication on CAN Bus. Just CAN ID "0" with dlc "0" is sent by the ECU.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

It happens when the start address is not aligned to a 64KByte block.	
Resolution Description:	
Workaround:	

Do only configure Start Addresses for the CAN Message RAM which are aligned to 64KByte boundaries.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00092569 Compiler error: identifier "pduInfo_var_id" is undefined	
Component@Subcomponent:	DrvCan_Mpc5700McanLI@Implementation
First affected version:	2.08.00
Fixed in versions:	2.10.00
Problem Description:	
What happens (symptoms):	

Compiler error: identifier "pduInfo_var_id" is undefined	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Only if the MCAN Revision is less than 3.1.0 (CAN_MCAN_REVISION < 0x0310)	
AND	
CAN FD is activated (CAN_FD_SUPPORT != CAN_NONE).	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00092622 A change of the main function period does not lead to a rebuild of the SWC description	
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- The SWC description file is not updated after a change of the EcuM main function period. When does this happen: ----- After change of the parameter /MICROSAR/EcuM/EcuMGeneral/EcuMMainFunctionPeriod. In which configuration does this happen: ----- In all configurations.	
Resolution Description: Workaround: ----- Adapt another parameter which leads to a rebuild of the SWC description, e.g. rename of a sleepmode [/EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMSleepMode]. After rebuild the name of this sleepmode can be switched back to the old name, the rename is only necessary to trigger a rebuild. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00092718 **<MSN>90005 - Generator (<Generator Name>) failure, because of an exception "exception in <Msn> generator during Validation encountered: java.lang.NullPointerException"**

Component@Subcomponent: CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr

First affected version: 5.01.00

Fixed in versions: 5.02.00

Problem Description:

What happens (symptoms):

During code generation, an error message similar to the following one:

[Error] <MSN>90005 - Generator (<Generator Name>) failure, because of an exception
- Exception in <Msn> generator during Validation encountered:
java.lang.NullPointerException

Erroneous CEs:

[DefinitionRef: /MICROSAR/<Msn>]

CEs:

[DefinitionRef: /MICROSAR/<Msn>]

AutoSolvingAction: <none>

PreferredSolvingAction: <none>

SolvingActions: <none>

GeneratorPackage: <Generator Name>(<Generator version> - com.vector.cfg.gen.<module name>)

When does this happen:

During code generation with DaVinci Configurator.

In which configuration does this happen:

Any

Resolution Description:

Workaround:

Use

com.vector.cfg.gen.CommonAsr_ComGenericGenLib.data.access.GenericGenAccess.getStruct(String
instead

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00092720 DataRenamer not working for MICROSAR Define block	
Component@Subcomponent:	CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr
First affected version:	5.01.00
Fixed in versions:	5.02.00
Problem Description:	
What happens (symptoms):	

The data renamer does not work for the MICROSAR define block.	
When does this happen:	

During code generation with DaVinci Configurator.	
In which configuration does this happen:	

Any	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00092955	Compiler error: incompatible types - from 'const <MSN>_PCConfigType *' to 'const <MSN>ConfigType *const'
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	
Problem Description:	

ESCAN00092955 Compiler error: incompatible types - from 'const <MSN>_PCConfigType *' to 'const <MSN>ConfigType *const'

What happens (symptoms):

The compiler throws an error like the following:

```
1> EcuM_Init_Cfg.c
1>GenData/EcuM_Init_Cfg.c(86): error C4133: 'initializing' : incompatible types - from 'const CanNm_PCConfigType *' to 'const EcuM_PbConfigType *const '
1>GenData/EcuM_Init_Cfg.c(87): error C4133: 'initializing' : incompatible types - from 'const EcuM_PCConfigType *' to 'const SchM_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(88): error C4133: 'initializing' : incompatible types - from 'const SchM_ConfigType *' to 'const Can_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(89): error C4133: 'initializing' : incompatible types - from 'const Can_PCConfigType *' to 'const CanIf_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(90): error C4133: 'initializing' : incompatible types - from 'const CanIf_PCConfigType *' to 'const Com_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(91): error C4133: 'initializing' : incompatible types - from 'const Com_PCConfigType *' to 'const PduR_PBConfigType *const '
1>GenData/EcuM_Init_Cfg.c(92): error C4133: 'initializing' : incompatible types - from 'const PduR_PCConfigType *' to 'const CanSM_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(93): error C4133: 'initializing' : incompatible types - from 'const CanSM_PCConfigType *' to 'const CanNm_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(103): error C4133: 'initializing' : incompatible types - from 'const CanNm_PCConfigType *' to 'const EcuM_PbConfigType *const '
1>GenData/EcuM_Init_Cfg.c(104): error C4133: 'initializing' : incompatible types - from 'const EcuM_PCConfigType *' to 'const SchM_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(105): error C4133: 'initializing' : incompatible types - from 'const SchM_ConfigType *' to 'const Can_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(106): error C4133: 'initializing' : incompatible types - from 'const Can_PCConfigType *' to 'const CanIf_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(107): error C4133: 'initializing' : incompatible types - from 'const CanIf_PCConfigType *' to 'const Com_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(108): error C4133: 'initializing' : incompatible types - from 'const Com_PCConfigType *' to 'const PduR_PBConfigType *const '
1>GenData/EcuM_Init_Cfg.c(109): error C4133: 'initializing' : incompatible types - from 'const PduR_PCConfigType *' to 'const CanSM_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(110): error C4133: 'initializing' : incompatible types - from 'const CanSM_PCConfigType *' to 'const CanNm_ConfigType *const '
```

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

In variant configurations with modules which uses different EcuC init phases in different variants (/MICROSAR/EcuC/EcucGeneral/BswInitialization/InitFunction/InitPhase).

E.g.

VARIANT_1: InitPhase = NO_INIT

VARIANT_2: InitPhase = INIT_TWO_MCAL

Resolution Description:

ESCAN00092955 Compiler error: incompatible types - from 'const <MSN>_PCConfigType *' to 'const <MSN>ConfigType *const

Workaround:

To resolve this the content of the `CONT EcuM_GlobalConfigRoot` in `EcuM_Init_Cfg.c` has to be reordered to fit to the struct `EcuM_GlobalConfigRootType`.

e.g.

```
CONST(EcuM_GlobalConfigRootType, ECUM_CONST) EcuM_GlobalConfigRoot =
{
{
BswM_Config_CanNm_Ptr,
EcuM_Config_CanNm_Ptr,
CanNm_Config_CanNm_Ptr,
},
{
BswM_Config_ClassB_Ptr,
CanNm_Config_ClassB_Ptr, <===== Wrong position, must be the last one
EcuM_Config_ClassB_Ptr,
},
{
BswM_Config_ClassC_Ptr,
CanNm_Config_ClassC_Ptr, <===== Wrong position, must be the last one
EcuM_Config_ClassC_Ptr,
}
};
```

typedef struct

```
{
CONSTP2CONST(BswM_ConfigType, TYPEDEF, BSWM_INIT_DATA) CfgPtr_BswM_Init;
CONSTP2CONST(EcuM_PbConfigType, TYPEDEF, ECUM_INIT_DATA) CfgPtr_EcuM_Init;
CONSTP2CONST(CanNm_ConfigType, TYPEDEF, CANNM_INIT_DATA) CfgPtr_CanNm_Init;
} EcuM_GlobalPCConfigType;
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00093294 Invalid key accepted due to inconsistent Csm and CryFord job processing configuration	
Component@Subcomponent:	Diag_AsrSwcSecAccess_Ford@Implementation
First affected version:	1.00.00
Fixed in versions:	2.00.00
Problem Description: What happens (symptoms): ----- SecurityAccess_FunctionFinish() declares key as valid even if the verification has failed. When does this happen: ----- While processing the security access key. In which configuration does this happen: ----- If configured Csm job processing type and CryFord job processing type are inconsistent. E.g. Csm configured for sync job processing and CryFord for async processing.	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00093317 Value Calculations does not act as expected	
Component@Subcomponent:	CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- An Value Calculation has only influence to values which are added to the ComStackLib. The values of the Define Creator are directly taken from the corresponding EcuC parameter. Furthermore, Value Calculations for references (i.e convert a reference to an integer of the target) do not work either. When does this happen: ----- Always during the generation In which configuration does this happen: ----- Generators which uses ValueCalculators for parameters which are created by the Microsar Define creator. or Value calculation is used for a reference	
Resolution Description: Workaround: ----- Use a struct extender / custom #define for the DefRef and blacklist the DefRef instead Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00093405 Auto Configuration - Invalid multiplicity after manual adaptations of container BswMAvailableActions

Component@Subcomponent: SysService_Asr4BswMCfg5@GenTool_GeneratorMsr

First affected version: 10.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

User-modifications about a changed BswMAvailableActions subcontainer are recognized by the Auto Configuration assistant but even if they should be kept, the assistant will re-create the original action. This leads to an invalid model because the user modification is not removed by the assistant.

Example:

- Configure Communication Control is used and Reinitialize TX is turned ON, Finish is clicked.
- the /MICROSAR/BswM/BswMConfig/BswMModeControl/BswMAAction CC_EnableDM_<I-PDU-Group> has a BswMDeadlineMonitoringControl container which is deleted within the Basic Editor
- Instead another BswMAvailableActions subcontainer is created of another type, e.g.

BswMComMMModelimitation

- Configure Communication Control is used once again and Finish is clicked. An option is offered to either keep this modification or to restore it, but independent of the choice, the original BswMDeadlineMonitoringControl is restored without removing the user modification. Because the user modification is not removed the multiplicity of the container BswMAvailableActions[0...1] is violated.

When does this happen:

During the configuration with DaVinci Configurator in the BSW Management Editor in the following sequence:

- Configure <Auto Configuration> is clicked
- Finish is clicked
- Some objects like a /MICROSAR/BswM/BswMConfig/BswMModeControl/BswMAAction/BswMAvailableActions/BswMDeadlineMonitoringControl container are deleted or changed
- Configure <Auto Configuration> is clicked once again
- Finish is clicked
- the dialog 'Manual Adaptions' does pop up
- Finish is clicked in the 'Manual Adaptions' dialog

In which configuration does this happen:

Any configuration using one of the Auto Configurations in BSW Management in DaVinci Configurator

Resolution Description:

Workaround:

Redo the previously manual changes that have been overwritten.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00093449 A2L compu method RTE_CM_BOOLEAN cannot be used to calibrate boolean values	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.16.00
Problem Description: What happens (symptoms): ----- Only FALSE can be selected in calibration tools for boolean calibration values ----- When does this happen: ----- During calibration ----- In which configuration does this happen: ----- If calibration is configured for a boolean element	
Resolution Description: Workaround: ----- Configure an integer element with a dedicated compu method instead of a boolean element. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00093502 Technical Reference: Wrong API description for Csm_SymKeyExtractStart**Component@Subcomponent:** SysService_AsrCsm@Doc_TechRef**First affected version:** 1.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

On TechnicalReference_Csm, Csm_SymKeyExtractStart prototype is
Csm_ReturnType Csm_SymKeyExtractStart (Csm_ConfigIdType cfgId, const Csm_SymKeyType *keyPtr)

But has to be

FUNC(Csm_ReturnType, CSM_CODE) Csm_SymKeyExtractStart(Csm_ConfigIdType cfgId)

There is no second parameter.

When does this happen:

-

In which configuration does this happen:

-

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00093634 CAN-FD format (Bosch V1.0, ISO-11898) inconsistent	
Component@Subcomponent:	DrvCan_Mpc5700McanLI@GenTool_GeneratorMsr
First affected version:	3.00.01
Fixed in versions:	3.04.00
Problem Description: What happens (symptoms): ----- After changing the Mcan Revision to M_CAN_REV_315 the parameter CanFd NISO is grey'ed and cannot be edited. This is not correct for M_CAN_REV_315 as for this revision both CAN-FD formats are available. When does this happen: ----- During configuration time. In which configuration does this happen: ----- In every configuration.	
Resolution Description: Workaround: ----- Mark CanFd NISO parameter as "Set user defined", configure the parameter as required, mark CanFd NISO parameter as "Set NOT user defined". Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00093839 CFG5 Exception or Compile Error "Too many initializer values"

Component@Subcomponent: CommonAsr_ComStackLib@GenTool_GeneratorMsr

First affected version: 4.00.00

Fixed in versions: 8.06.00, 8.05.01

Problem Description:

What happens (symptoms):

CFG5 shows the following error message
"Exception in <MSN> generator during Generation encountered"
and no files are generated.
The detailed error description is: java.lang.NullPointerException

OR

the compiler informs about arrays of structs with too many initializer values.

When does this happen:

at generation time

OR

at compile time

In which configuration does this happen:

Any configuration where the postbuild-selectable support is enabled for this module
AND
the generator uses the API setRequiresIndexUsedArray() with the parameter true.

Resolution Description:

Workaround:

deactivate ComStackLib boolean deduplications
AND
configure <MSN>StructBoolDataUsage to a value different from "BITMASKING"
AND
generate again.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094259 Auto-Configuration Communication Control shows an error in case of not available module Com**Component@Subcomponent:** SysService_Asr4BswMCfg5@GenTool_GeneratorMsr**First affected version:** 2.01.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

Auto-Configuration shows the following error:

Configuration *error*

Reason for *error*:

Could not collect all necessary informations. Solve errors in depending Modules first!

To see following errors in the Validation view execute on-demand generator validation!

Container ComConfig does not exist. Element def.: /[ANY]/Com/ComConfig

When does this happen:

Always during configuration.

In which configuration does this happen:

In all configurations without the module Com.**Resolution Description:**

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094298 The Ecu does not startup properly in some MultiCore configurations**Component@Subcomponent:** SysService_Asr4EcuM@GenTool_GeneratorMsr**First affected version:** 2.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

The Ecu does not startup properly, e.g. communication does not start , the application is not started, the configuration variant is not detected correct etc.

When does this happen:

Always after startup of the Ecu.

In which configuration does this happen:

In MultiCore configurations with parameter "/EcuM/EcuMGeneral/EcuMBswCoreId" set to another value than to the ID of the master core

AND

The OS symbols OS_CORE_ID_<X> are provided as a enumeration and not as a preprocessor define.

Resolution Description:

Workaround:

Create a header file with the following content and add it to /MICROSAR/EcuM/EcuMGeneral/EcuMUserConfigurationFile:

```
#if defined(ECUM_CFG_H)

# undef ECUM_CORE_ID_STARTUP
# undef ECUM_CORE_ID_BSW
# define ECUM_CORE_ID_STARTUP <Core Id as Numerical Value, e.g. '0>
# define ECUM_CORE_ID_BSW <Core Id as Numerical Value, e.g. '1>

#endif
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094319 Auto-Configuration Communication Control: Init Mode of Lin Schedule Indication is missing**Component@Subcomponent:** SysService_Asr4BswMCfg5@GenTool_GeneratorMsr**First affected version:** 10.01.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

A validator in Cfg5 reports the following warning:

BSWM01057 Init Mode of CC_LinScheduleIndication_<Schedule Name> is not known.
Set BswMBswModeInitValueMode(value=) to LinSMConf_LinSMSchedule_<NAME>
/ActiveEcuC/BswM/BswMConfig/BswMArbitration/
CC_LinScheduleIndication_LIN00_<Schedule_Name>/BswMModeInitValue/
BswMBswModeInitValue[BswMBswModeInitValueMode]
/ActiveEcuC/BswM/BswMConfig/BswMArbitration/
CC_LinScheduleIndication_LIN00_<Schedule_Name>

When does this happen:

Always after configuring the Auto-Configuration Communication Control.

In which configuration does this happen:

Only in configurations with at least one Lin channel

AND

Auto-Configuration Communication Control is configured.

Resolution Description:

Workaround:

Set the normal schedule via the provided solving action.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094355 [Error] CANIF10027 None CAN-channel has multiple BasicCAN Tx-objects. Hence the feature "CanIfMultipleBasicCANTxObjects" is not required in current configuration and must be disabled.

Component@Subcomponent: If_AsrIfCan@GenTool_GeneratorMsr

First affected version: 4.02.00

Fixed in versions:

Problem Description:

What happens (symptoms):

One of the following validation messages always occurs during configuration:

[Error] CANIF10027 - A feature is not supported in current configuration and shall be disabled.
- None CAN-channel has multiple BasicCAN Tx-objects. Hence the feature "CanIfMultipleBasicCANTxObjects" is not required in current configuration and must be disabled.
Solving action: Disable parameter: "CanIfMultipleBasicCANTxObjects".

-> After executing of this solving action you get the following validation message within the CanDrv:

[Error] CAN02002 - An invalid value is configured
- CanMultipleBasicCANTxObjects is not active but multiple TX BasicCANs used on some controller.
Solving action: Enable parameter: "CanMultipleBasicCANTxObjects"

-> After execution of this solving action you get the validation message mentioned above

When does this happen:

During configuration

In which configuration does this happen:

In case there is at least one CAN channel with no BasicCAN Tx-hardware object (there is no "CanHardwareObject" with "CanHandleType" == BASIC and "CanObjectType" == TRANSMIT)
-> The configuration has only FullCAN-objects or no Tx-objects at all on at least one channel

Resolution Description:

Workaround:

Make sure you get [Error] CANIF10027 (i.e. solve [Error] CAN02002 if present).
Set the parameter "CanIfMultipleBasicCANTxObjects" to user defined and keep it enabled.
[Error] CANIF10027 is then demoted to a warning that can be ignored.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094416 Linker error: undefined reference to ComM_Nm_PrepareBusSleepMode

Component@Subcomponent: Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr

First affected version: 7.00.00

Fixed in versions: 8.01.01

Problem Description:

What happens (symptoms):

A Compiler warning or error occurs similar to:

- missing prototype
 - 'ComM_Nm_PrepareBusSleepMode' undefined; assuming extern returning int
 - Implicit parameter-declaration for 'ComM_Nm_PrepareBusSleepMode'
 - function ComM_Nm_PrepareBusSleepMode declared implicitly
- ComM_Nm_PrepareBusSleepMode(nmNetworkHandle)

Additionally a linker error occurs similar to:

- Nm.obj : unresolved external symbol _ComM_Nm_PrepareBusSleepMode referenced in function _Nm_BusSleepMode

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

Configurations with CAN channels only where J1939Nm is attached on each CAN channel and no CanNm
(/MICROSAR/Nm/NmChannelConfig/NmBusType/NmStandardBusNmConfig/NmStandardBusType
= NM_BUSNM_J1939NM exists in each NmChannelConfig container)
=> ComM_Cfg.h contains #define COMM_SILENTSUPPORTOFCHANNEL STD_OFF

AND

On at least one channel, another BusNm is configured as generic BusNm
(/MICROSAR/Nm/NmChannelConfig/NmBusType/NmGenericBusNmConfig is defined)

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094506 API CanIf_CheckBaudrate is not described / the API CanIf_ChangeBaudrate is described twice	
Component@Subcomponent:	If_AsrIfCan@Doc_TechRef
First affected version:	6.00.00
Fixed in versions:	6.12.00
Problem Description:	
What happens (symptoms):	

The API: "CanIf_CheckBaudrate" is not described, but the API: "CanIf_ChangeBaudrate" is described twice	
When does this happen:	

- During reading of technical reference	
In which configuration does this happen:	

-	
Resolution Description:	
Workaround:	

Please see the CanIf-implementation or the AUTOSAR-specification for more information about the API: CanIf_CheckBaudrate().	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00094541 Auto-Configuration Communication Control: Rules without expressions are created and so validation errors are shown**Component@Subcomponent:** SysService_Asr4BswMCfg5@GenTool_GeneratorMsr**First affected version:** 11.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

The validation tab shows the following message:

AR-ECUC02008 Invalid multiplicity (3 messages)
AR-ECUC02008 Mandatory parameter BswMRuleExpressionRef is missing in
CC_<CHANNELNAME>_<PNCNAME>_RX
BswMRuleExpressionRef
/ActiveEcuC/BswM/BswMConfig/BswMArbitration/CC_<CHANNELNAME>_<PNCNAME>
AR-ECUC02008 Mandatory parameter BswMRuleExpressionRef is missing in
CC_<CHANNELNAME>_<PNCNAME>_RX_DM
BswMRuleExpressionRef
/ActiveEcuC/BswM/BswMConfig/BswMArbitration/CC_<CHANNELNAME>_<PNCNAME>
AR-ECUC02008 Mandatory parameter BswMRuleExpressionRef is missing in
CC_<CHANNELNAME>_<PNCNAME>_TX
BswMRuleExpressionRef
/ActiveEcuC/BswM/BswMConfig/BswMArbitration/CC_<CHANNELNAME>_<PNCNAME>

When does this happen:

Always after execution of the Communication Control assistant.

In which configuration does this happen:

In configurations with PNCs where at least one PduGroup is mapped to different PNCs

AND

Not all PNCs of a channel are configured (selected) in the Communication Control assistant.

Resolution Description:

Workaround:

Rules must be deleted manually from configuration.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094612 WdgM_GetTickCount is called with suspended interrupts	
Component@Subcomponent:	SysService_Asr4WdM@Implementation
First affected version:	5.02.00
Fixed in versions:	5.02.03
Problem Description:	
What happens (symptoms):	

WdgM_GetTickCount is called with suspended interrupts. If the timebase source is configured to be an Os counter, this results in an error. It is not allowed to call Os-APIs with suspended interrupts. The consequence is that the Os will run in error hook and will shutdown.	
When does this happen:	

Always and immediately if an Os Counter is used as timebase.	
In which configuration does this happen:	

/MICROSAR/WdgM/WdgMGeneral/WdgMTimebaseSource == WDGM_OS_COUNTER_TICK	
Resolution Description:	
Workaround:	

Implement WDGM_EXCLUSIVE_AREA_0 with an OS resource and assign all tasks to this resource, to which the following runnable/schedulabe entities are mapped:	
<ul style="list-style-type: none">- WdgM_Init- WdgM_MainFunction- All runnables calling any CheckpointReached operation	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00094806 Compiler error: Undefined symbol RTE_MODE_DcmDiagnosticSessionControl_DEFAULT_SESSION

Component@Subcomponent: Diag_Asr4Dcm@GenTool_GeneratorMsr

First affected version: 4.00.00

Fixed in versions: 8.02.00

Problem Description:

What happens (symptoms):

Typical compiler error explanations may be:

Symbol RTE_MODE_DcmDiagnosticSessionControl_DEFAULT_SESSION not defined.

Note: The above symbol is only an example. Typically all used diagnostic session related RTE mode defines will invoke an error, such as EXTENDED_SESSION, PROGRAMMING_SESSION etc.

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

- DCM is licensed for AR 4.2.1 or newer version.

AND

- An application code (SW-C) uses the AR 4.2.X or newer RTE mode names (e.g. RTE_MODE_DcmDiagnosticSessionControl_DEFAULT_SESSION).

Resolution Description:

Workaround:

Create compatibility defines in the SW-C code or one of the included files (e.g. in worst case in Compiler_Cfg.h) to link the two AR standards:

Example:

```
#define RTE_MODE_DcmDiagnosticSessionControl_DEFAULT_SESSION
RTE_MODE_DcmDiagnosticSessionControl_DefaultSession
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00094875		Compiler error: dld.exe: warning: Undefined symbol 'MemIf_*_WriteWrapper' in file 'obj/MemIf_Cfg.o'
Component@Subcomponent:	If_AsrIfMem@GenTool_GeneratorMsr	
First affected version:	5.02.00	
Fixed in versions:		
Problem Description:		
What happens (symptoms):		

Compiler error: dld.exe: warning: Undefined symbol 'MemIf_*_WriteWrapper' in file 'obj/MemIf_Cfg.o'		
When does this happen:		

During linking the project		
In which configuration does this happen:		

Windriver Diab compiler for PPC version is used (tested with version 5.9.4.8)		
Resolution Description:		
Workaround:		

Redefine MEMIF_LOCAL_INLINE to MEMIF_LOCAL (e.g. in Compiler_Cfg.h)		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00094883 Improper workaround for MCAN Erratum #10	
Component@Subcomponent:	DrvCan_Mpc5700McanLI@Implementation
First affected version:	2.04.00
Fixed in versions:	2.15.00
Problem Description:	
What happens (symptoms):	

In case of a (Re-)Init while a MCAN TX Scan is in progress the MCAN TX Handler stops. After going back to Normal Mode the MCAN TX Handler does not execute any transmission requests.	
When does this happen:	

At runtime	
In which configuration does this happen:	

Only if the Workaround for MCAN Erratum #10 is activated (CAN_BOSCH_ERRATUM_010 == STD_ON).	
The MCAN Erratum #10 is only relevant for MCAN Revisions less than 3.1.0.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00094972		Compiler error: Missing declaration for APIs Dcm_OptimizedSetNegResponse and Dcm_OptimizedProcessingDone
Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr	
First affected version:	1.00.00	
Fixed in versions:	8.01.00	
Problem Description:		
What happens (symptoms):		

Compiler error due to missing declaration for APIs Dcm_OptimizedSetNegResponse and Dcm_OptimizedProcessingDone.		
When does this happen:		

The error is issued by the compiler during compilation of the code in case the configuration is as described below.		
In which configuration does this happen:		

- At least one user sub-service is configured (ECUC parameter DcmDsdSubServiceFnc is not empty)		
AND		
- There is no user service configured (all ECUC parameters DcmDsdSidTabFnc are empty/missing)		
AND		
- In Dcm_Ext.c is no service or sub-service handler active (switch name depends on the extension type)		
Resolution Description:		
Workaround:		

Add the following lines in a user-configuration file for DCM:		
/* ESCAN00094972 WA BEGIN */		
# undef DCM_DIAG_EXTERN_SVC_HANDLING_ENABLED		
# define DCM_DIAG_EXTERN_SVC_HANDLING_ENABLED STD_ON		
/* ESCAN00094972 WA END */		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00094980 Generator version check fails	
Component@Subcomponent:	Tp_Asr4TpCan@ElisaPlugin
First affected version:	1.00.01
Fixed in versions:	1.01.00
Problem Description:	
What happens (symptoms):	

The following line appears in the MICROSAR Safe Silence Verification Report:	
error: assertion 'CANTP_CFG_GEN_MINOR_VERSION: "2U" == "1U" does not hold	
When does this happen:	

When running the Asr4TpCan MSSV checks.	
In which configuration does this happen:	

This happens only when using the affected plugin to verify data from 4.02.xx CanTp generators. No errors should arise when verifying data from 4.01.xx CanTp generators.	
Resolution Description:	
Workaround:	

The version check could be ignored. Even if the generator has been changed, the required checks remain the same, they do run and are properly showed in the report.	
Resolution:	

The version check is now correct.	

ESCAN00094991 Generated SW-C template for S/R data interface does not comply with AR 4.x standard	
Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	8.02.00
Problem Description:	
What happens (symptoms):	

Direct port mapping between DCM and DEM not possible.	
RTE validation check message ID RTE51008 is issued in DaVinci configurator.	
When does this happen:	

At port mapping time.	
In which configuration does this happen:	

- DCM and DEM support S/R data access.	
AND	
- Direct mapping between DCM and DEM shared data is required.	
Resolution Description:	
Workaround:	

Apply port interface mapping manually.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095065 Compiler error: Missing declaration for API BswM_Dcm_RequestResetMode()**Component@Subcomponent:** Diag_Asr4Dcm@Implementation**First affected version:** 5.01.00**Fixed in versions:** 8.01.00**Problem Description:**

What happens (symptoms):

Compiler error due to missing declaration of BswM_Dcm_RequestResetMode() API .

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

- Dcm is integrated in ASR 3 environment (Dcm_Cfg.h:
DCM_BSW_ENV_ASR_VERSION_3XX_ENABLED == STD_ON)
AND
- Final response after reset from FBL is not supported (Dcm_Cfg.h:
DCM_DIAG_JUMPFROMFBL_ENABLED == STD_OFF)
AND
- Service 0x28 is not handled by DCM (Dcm_Cfg.h: DCM_SVC_28_SUPPORT_ENABLED ==
STD_OFF)
AND
- At least one sub-function of service 0x11 is handled by Dcm (Dcm_Cfg.h:
DCM_MODE_ECU_RESET_ENABLED == STD_ON || DCM_MODE_RPD_SHTDWN_ENABLED ==
STD_ON)
OR
- At least one diagnostic session is configured for jumping to FBL (Dcm_Cfg.h:
DCM_DIAG_JUMPTOFBL_ENABLED == STD_ON)**Resolution Description:**

Workaround:

Enable feature jump from FBL. Set the ECUC parameter DcmFinalResponseToFblEnabled = TRUE.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095083 Compiler error: #20: identifier "EcuM_WakeupSourceType" is undefined**Component@Subcomponent:** DrvTrans__baseCanxAsr@GenTool_GeneratorMsr**First affected version:** 3.00.00**Fixed in versions:** 3.03.00**Problem Description:**

What happens (symptoms):

Compiler error occurs: "error #20: identifier "EcuM_WakeupSourceType" is undefined "

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

See file: CanIf_Cfg.h

- CANIF_WAKEUP_SUPPORT == STD_OFF

AND

- CANIF_WAKEUP_VALIDATION == STD_OFF

AND

- CANIF_CONFIG_VARIANT != CANIF_CFGVAR_POSTBUILDTIME

AND CanIf-Version is >= 6.13.00 (See CanIf.h -> define: IF_ASIFCAN_VERSION >= 0x0613u)

Resolution Description:

Workaround:

1) Enable wakeup within a CAN channel or a CAN transceiver channel, depending on change generate afterwards CanDrv / CanTrcv and CanIf once again [if applicable]

or

2) Create a user configuration file for CanIf and add #include "EcuM_Cbk.h", generate afterwards at least CanIf once again [recommended]

or

3) Just enable the parameter "CanIfWakeupSupport" and set it to user defined and generate afterwards at least the CanIf once again

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095259 Compiler error: WdgIf uses undefined memory sections

Component@Subcomponent: If_Asr4IfWd@GenTool_GeneratorMsr

First affected version: 2.01.00

Fixed in versions:

Problem Description:

What happens (symptoms):

WdgIf uses memory section which are not defined. The WdgIf assumes erroneously that Os provides these sections. This error leads to a compiler error like: #error "MemMap.h, wrong pragma command"

The sections of the WdgIf

- WDGIF_START_SEC_VAR_INIT_8BIT / WDGIF_STOP_SEC_VAR_INIT_8BIT
- WDGIF_START_SEC_VAR_INIT_16BIT / WDGIF_STOP_SEC_VAR_INIT_16BIT
- WDGIF_START_SEC_VAR_INIT_32BIT / WDGIF_STOP_SEC_VAR_INIT_32BIT

are mapped to

(Gen6)

- <ApplicationName>_START_SEC_VAR_<InitPolicy>_8BIT /
- <ApplicationName>_STOP_SEC_VAR_<InitPolicy>_8BIT
- <ApplicationName>_START_SEC_VAR_<InitPolicy>_16BIT /
- <ApplicationName>_STOP_SEC_VAR_<InitPolicy>_16BIT
- <ApplicationName>_START_SEC_VAR_<InitPolicy>_32BIT /
- <ApplicationName>_STOP_SEC_VAR_<InitPolicy>_32BIT.

(Gen7)

- OS_START_SEC_<ApplicationName>_VAR_<InitPolicy>_8BIT /
- OS_STOP_SEC_<ApplicationName>_VAR_<InitPolicy>_8BIT
- OS_START_SEC_<ApplicationName>_VAR_<InitPolicy>_16BIT /
- OS_STOP_SEC_<ApplicationName>_VAR_<InitPolicy>_16BIT
- OS_START_SEC_<ApplicationName>_VAR_<InitPolicy>_32BIT /
- OS_STOP_SEC_<ApplicationName>_VAR_<InitPolicy>_32BIT.

Os currently supports only "InitPolicy" {-, NOINIT, ZEROINIT}. The actually needed init policy is "INIT".

When does this happen:

Only if a multi core platform is used and the WdgIf is configured to use the state combiner functionality.

In which configuration does this happen:

If a container "/MICROSAR/WdgIf/WdgIfStateCombiner" is configured
AND
if /MICROSAR/WdgIf/WdgIfGeneral/WdgIfUseStateCombiner == true

Resolution Description:

Workaround:

Provide the missing memory sections and locate them in a proper memory section.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095262 Missing record layout for uint64 and sint64 data types**Component@Subcomponent:** Rte_Core@Implementation**First affected version:** 1.00.00**Fixed in versions:** 1.16.00**Problem Description:**

What happens (symptoms):

The A2L file misses a record layout for 64 bit measurement objects

E.g. the generator generates:

```

/begin MEASUREMENT Rte_c_PerInstanceMemory_3 "MICROSAR RTE"
A_UINT64 NO_COMPU_METHOD 0 0 0 18446744073709551615
ECU_ADDRESS 0
/end MEASUREMENT

```

but does not generate

```

/begin RECORD_LAYOUT RTE_RL_A_UINT64
FNC_VALUES 1 A_UINT64 ROW_DIR DIRECT
/end RECORD_LAYOUT

```

When does this happen:

During generation

In which configuration does this happen:

This happens when the configuration contains sender-receiver ports, inter-runnable variables, calibration parameters or per-instance memories that use 64 bit integer data types and when measurement is enabled for the objects.**Resolution Description:**

Workaround:

Manually add the record layout to the skeleton A2L file that includes Rte.a2l.

```

/begin RECORD_LAYOUT RTE_RL_A_UINT64
FNC_VALUES 1 A_UINT64 ROW_DIR DIRECT
/end RECORD_LAYOUT

```

```

/begin RECORD_LAYOUT RTE_RL_A_INT64
FNC_VALUES 1 A_INT64 ROW_DIR DIRECT
/end RECORD_LAYOUT

```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095296 Validation error: Reference to XcpOnCan/XcpOnCanVariableDlc not found	
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	2.01.00
Problem Description: What happens (symptoms): ----- The validator throws an error about a missing reference to /MICROSAR/Xcp/XcpConfig/XcpTransportLayer/XcpOnCan/XcpOnCanVariableDlc if the Container /MICROSAR/Xcp/XcpConfig/XcpTransportLayer/XcpOnCan is deleted when CAN is not used. When does this happen: ----- Always and immediately In which configuration does this happen: ----- If XcpOnCan is not used	
Resolution Description: Workaround: ----- After restart of Cfg5 the validation error is gone. It will re-appear if the container is created and deleted again. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00095310 Compiler error: identifier EcuM_GlobalConfigRoot not declared	
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	8.00.04
Problem Description:	
What happens (symptoms):	

The compiler throws the following (or similar) error:	
"../../../../external/BSW/EcuM/EcuM.c", line 2959: error (dcc:1525): identifier EcuM_GlobalConfigRoot not declared	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Only in PB-S configurations with MCAL modules which do only support PB-L.	
Resolution Description:	

ESCAN00095310 Compiler error: identifier EcuM_GlobalConfigRoot not declared

Workaround:

1. Workaround:

This workaround does only work if in /ActiveEcuC/EcuC/EcucGeneral/BswInitialization no entry exists for one of those MCAL modules with entry "AdditionalInitCode" and without a ConfigPtrName set.

Add MCAL modules which do only support PB-L to the list of individual Postbuild modules in the configuration of the module EcuC:

EcuC/EcucGeneral/PostbuildLoadable/IndividualPostBuildLoadableModule

HINT: Consider the .groovy script in the attachments, this will add the individual Postbuild modules mentioned in workaround 1.

2. Workaround:

If Workaround 1 is not working because of the restrictions mentioned in the 1. workaround please use the following workaround:

Adaption of the files EcuM_Init_Cfg.h and EcuM_Init_Cfg.c. The adaption of these files is okay because these are template files.

Adapt the EcuM_Init_Cfg.h as follows:

```
...
extern CONST(EcuM_GlobalConfigRootType, ECUM_CONST) EcuM_GlobalConfigRoot;
...
```

Adapt the EcuM_Init_Cfg.c as follows:

```
....
CONST(EcuM_GlobalConfigRootType, ECUM_CONST) EcuM_GlobalConfigRoot =
{
<CONTENT>
};
....
```

Resolution:

The described issue is corrected by modification of all affected work-products.

**ESCAN00095312 Generation cannot be started because of an error
COMM90500 - A generated value is not in range of
the specified datatype****Component@Subcomponent:** Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr**First affected version:** 3.00.00**Fixed in versions:** 8.01.01**Problem Description:**

What happens (symptoms):

[Error] COMM90500 - A generated value is not in range of the specified datatype
- The value 256 with comment () is not in the range of the specified datatype UINT_8.

When does this happen:

During code generation in the calculation or validation phase.

In which configuration does this happen:

The number of connections between ComMChannels and ComMUsers exceeds 255.

Example: assume there are 15 ComMChannels and 17 ComMUsers.

If each ComMChannel is connected to each ComMUser there are $15 * 17 = 255$ connections, the issue does not occur.

After adding a new ComMUser and connecting it to a ComMChannel the number of connections increases by 1 and becomes 256 and the issue occurs.

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095342 Compiler error: identifier PduR Routing Manager APIs not declared	
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	11.00.00
Fixed in versions:	12.00.00
Problem Description:	
What happens (symptoms):	

Compiler error: identifier PduR_RmTp_CopyRxData(), PduR_RmTp_CopyTxData(), PduR_RmTp_TpRxIndication() and PduR_RmTp_TpTxConfirmation not declared. or Compiler error: identifier PduR_RmIf_RxIndication(), PduR_RmIf_RxIndication_MultiIf and PduR_RmIf_TxConfirmation not declared.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

- configuration variant: Postbuild selectable	
and one variant has gateway pathways and a other variant does not have any gateway pathways (forwarding only).	
Resolution Description:	
Workaround:	

Use a gateway path in one of the variants as an invariant path. In the variants without gateway the path function as a dummy pathway.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095432 RTE generator generates to many compu scales	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.16.00
Problem Description:	
What happens (symptoms):	

Measurement or characteristic objects in the A2L files reference compu methods with COMPU_VTAB_RANGE although the used data types do not define any enumeration literals.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains application data types with texttable compu methods and when the application data types are mapped to implementation data types that are also used without mapping them to the same application data type. E.g. when an application data type is mapped to a platform type, the compu scale literals of the application data type will be generated for all measurement objects that use the platform type.	
Resolution Description:	
Workaround:	

Create separate implementation data types for all interfaces.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095490 Compiler error: Cannot open include file: 'Det.h'	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.03.00
Fixed in versions:	8.03.00
Problem Description:	
What happens (symptoms):	

DCM will not compile due to missing DET component header file.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

- Development error detection is supported (in Dcm_Cfg.h: DCM_DEV_ERROR_DETECT == STD_ON)	
AND	
- Development error reporting is not supported (in Dcm_Cfg.h: DCM_DEV_ERROR_REPORT == STD_OFF)	
AND	
- DET component is not available (Det.h is not provided in the project)	
Resolution Description:	
Workaround:	

Provide an empty Det.h file.	
Resolution:	

Change dependency for inclusion of Det.h from DCM_DEV_ERROR_DETECT to DCM_DEV_ERROR_REPORT.	

ESCAN00095519 ConsistencyRT00002 Error at validator runtime: CanSMBorTxConfPollingValidator if CanIf is missing	
Component@Subcomponent:	Ccl_Asr4SmCan@GenTool_GeneratorMsr
First affected version:	3.02.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

ConsistencyRT00002 - Error at validator runtime	
When does this happen:	

If the CanIf Module is deleted or a configuration without CanIf is loaded	
In which configuration does this happen:	

If CanIf is deleted or missing	
Resolution Description:	
Workaround:	

Activate CanIf Module or remove the CanSM Module and reload the project	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095528 Compiler error: Undeclared Identifier 'COM_UINT8_APPLTYPEOFRXACCESSINFO'	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	9.00.00
Fixed in versions:	13.03.00
Problem Description:	
What happens (symptoms):	

The compiler reports an undeclared Identifier 'COM_UINT8_APPLTYPEOFRXACCESSINFO'.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

In configurations without any uint8 signal or groupSignal and array access enabled for at least one signalGroup.	
Resolution Description:	
Workaround:	

Add a dummy signal with appl. Type uint8	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095553		Compiler error: Undefined identifier *PtrType to VARs of simple types with <MSN>_USE_INIT_POINTER to STD_ON
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr	
First affected version:	8.04.00	
Fixed in versions:	8.09.00	
Problem Description:		
What happens (symptoms):		

Compile error occurs in the doxygen group *PBRootValueTypes. The type definition of the *PtrType used for VARs of simple types is undefined.		
When does this happen:		

The error is issued by the compiler during compilation of the code in case the configuration is as described below.		
In which configuration does this happen:		

Any configuration where <MSN>_USE_INIT_POINTER is defined to STD_ON AND the generator instantiates objects of the type IVarVar (simple VARs in the config root) in the Configuration Class LINK time or POST-BUILD.		
Resolution Description:		
Workaround:		

No workaround available.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

**ESCAN00095570 Auto-Configuration Communication Control:
Activation of a PNC on one channel does also affect
other channels with same PNC****Component@Subcomponent:** SysService_Asr4BswMCfg5@GenTool_GeneratorMsr**First affected version:** 11.00.00**Fixed in versions:** 12.00.00, 11.00.04**Problem Description:**

What happens (symptoms):

The communication control shows all PDU Groups of a PNC, even if the PDU Groups are not relevant for this channel. On activation of one PDU Group, other PDU Groups of the same PNC on other channels are also affected.

Example:

ChannelONE ->PNCA -> PDUGroupA

ChannelTWO ->PNCA -> PDUGroupA

An activation of PDUGroupA on ChannelOne does also affect the configuration of ChannelTWO.

When does this happen:

Always during configuration.

In which configuration does this happen:

In configurations with PNCs which are mapped to more than one channel on the same ECU.

Resolution Description:

ESCAN00095570 Auto-Configuration Communication Control: Activation of a PNC on one channel does also affect other channels with same PNC

Workaround:

Do not activate PDU Groups in context of a channel where the PDU Group is not relevant for this channel. Additional ensure that always the channel, the PNC and all relevant PDU Groups are activated (Marked with a check in the checkbox).

Example:

ChannelONE
-> PNCA
-> PDUGroupAChannelONE
-> PDUGroupBChannelTWO
ChannelTWO
-> PNCA
-> PDUGroupAChannelONE
-> PDUGroupBChannelTWO

Ensure that the following checkboxes are set to start PDUGroupAChannelONE when ChannelONE and PNCA are in Full Communication:

X ChannelONE
X PNCA
X PDUGroupAChannelONE
O PDUGroupBChannelTWO (State of the checkbox might be wrong but don't care for this channel)

Ensure that the following checkboxes are set to start PDUGroupAChannelTWO when ChannelTWO and PNCA are in Full Communication:

X ChannelTWO
X PNCA
O PDUGroupAChannelONE (State of the checkbox might be wrong but don't care for this channel)
X PDUGroupBChannelTWO

Additional check the following three logical expressions (if existing):

CC_PNC_<CHANNELNAME>_<PNCNAME>_RX
CC_PNC_<CHANNELNAME>_<PNCNAME>_RX_DM
CC_PNC_<CHANNELNAME>_<PNCNAME>_TX

All three expressions must be like this:

CC_PNC_<PNCNAME> != COMM_PNC_NO_COMMUNICATION
&& CC_CanSMIndication_<CHANNELNAME> != CANSM_BSWM_NO_COMMUNICATION

If one condition is missing please extend the logical expression to be like described here.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095571 EcuM causes a Rte warning about a not existing mode request type map	
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

During validation the Rte throws the following warning:	
Mode Declaration Group <EcuM_Mode> of Component <EcuM> has no mode request type map. Each Mode Declaration Group used in the SW-C's ports has to have a unique mapping to an implementation data type. The Mode Group Data Type is set to <uint8>.	
Help:	
- define a mode request type map.	
When does this happen:	

During validation of the Rte.	
In which configuration does this happen:	

If /MICROSAR/EcuM/EcuMGeneral/EcuMEnableFixBehavior is set	
OR	
If /MICROSAR/EcuM/EcuMFlexGeneral/EcuMModeHandling is set	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095591 Compiler error: Invalid suffix 'U' on floating constant	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.16.00
Problem Description:	
What happens (symptoms):	

Compilation fails because the RTE generates an integer constant / upper limit using floating point notation and suffix 'U', e.g. 1.84467440737096e+019U.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when the configuration contains application data types with high upper limits.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095642 Compiler error: Service 0x2A is enabled, but no periodic messages have been configured for Dcm

Component@Subcomponent: Diag_Asr4Dcm@Doc_TechRef

First affected version: 7.02.00

Fixed in versions: 8.06.00

Problem Description:

What happens (symptoms):

Compiler issues the following error message:

Service 0x2A is enabled, but no periodic messages have been configured for Dcm. Please, refer to the Dcm TechRef for SID 0x2A configuration aspect.

Note: This is a DCM error check but in the Technical Reference it is not described that periodic transmission will not be considered in case Safe BSW Check is enabled.

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

This happens in all configurations where

- Service 0x2A is configured in Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTables/DcmDsdServiceTable/DcmDsdServices

AND

- at least one Periodic Transmission is configured (/ActiveEcuC/Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/<ProtocolName>/DcmDslPeriodicTransmission)

AND

- Safe BSW Check (/MICROSAR/Dcm/DcmConfigSet/DcmGeneral/DcmSafeBswChecks) is enabled

Resolution Description:

Workaround:

In case you do not have DCM delivered as Safe BSW component disable Safe BSW Check (/MICROSAR/Dcm/DcmConfigSet/DcmGeneral/DcmSafeBswChecks).

In the other case you cannot use periodic transmission as this is in current versions not supported in combination with Safe BSW.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095660 [Applies only if hardware: Tle9252 is used] BETA version - the BSW module is in BETA state	
Component@Subcomponent:	DrvTrans_Tja1043CandioAsr@Implementation
First affected version:	4.02.00
Fixed in versions:	4.03.00
Problem Description: What is the impact of BETA software: ----- BETA software - only in case of CAN transceiver hardware: Tle9252 is used in conjunction with this driver. - must not be used in productive projects as they may result in unpredictable ECU behavior - may not provide all features intended for the productive project - is not or only partly tested and not all quality measures have taken place Which functionality is BETA: ----- The complete BSW module is in BETA state only in case of CAN transceiver hardware: Tle9252 is used in conjunction with this driver. -> Especially the wake-up detection does not allow the distinction between WUP/LWU. Wake-up is reported always as WUP.	
Resolution Description: Resolved in STORYC-3911	

ESCAN00095684 esl_decryptAES* causes memory corruption**Component@Subcomponent:** SysService_CryptoCv@Impl_ESLib**First affected version:** 1.00.00**Fixed in versions:** 2.09.00**Problem Description:**

What happens (symptoms):

AES decryption may lead to memory corruption (write access out of bounds)

Additionally AES decryption cannot complete; eslFinalizeDecryptAES* will fail, returning error code ESL_ERR_WS_TOO_SMALL.

When does this happen:

It happens always under following conditions:

- esl_initWorkSpaceHeader was called with a workspace size of ESL_MINSIZEOF_WS_AES128.
- esl_initDecryptAES* was successfully called with this workspace, and
- esl_decryptAES* is called with an input length that is a multiple of AES_BLOCK_SIZE (16bytes).

In which configuration does this happen:

It happens in all configurations.**Resolution Description:**

Workaround:

Hand over a workspace with the right buffer size ESL_MAXSIZEOF_WS_AES128.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095690 RTE Analyzer fails due to duplicated runnable functions	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.16.00
Problem Description: What happens (symptoms): ----- Rte Analyzer aborts with error message: ERROR: Linking globals named 'Dem_GetDTCOfEvent': symbol multiply defined! or: ERROR: Linking globals named 'WdgM_CheckpointReached': symbol multiply defined!	
When does this happen: ----- The error is issued by RTE Analyzer during compilation of the code in case the configuration is as described below.	
In which configuration does this happen: ----- This happens when the configuration contains multiple service components with different names and the same runnable entity symbol.	
Resolution Description: Workaround: ----- Remove the implementation of the regarded service component runnable from all but one RteAnalyzer stubs.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00095730 COM02300: Proposed ComBitSize may exceed allowed range for UINT8_N and UINT8_DYN signals	
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	13.03.00
Problem Description:	
What happens (symptoms):	

COM02300 proposes a ComBitSize for UINT8_N /DYN signals that matches the ComSignalLength. However, the proposed ComBitSize may exceed the max. allowed length (64 Bits).	
When does this happen:	

- Live validation warning is issued and preferred solving action is proposed	
In which configuration does this happen:	

- ComSignalType of ComSignal or ComGroupSignal is set to UINT8_N or UINT8_DYN	
- ComSignalLength > 8 Bytes	
Resolution Description:	
Workaround:	

Delete ComBitSize Parameter for Array Based ComSignals / ComGroupSignals.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095747 Generator error for duplicate runnable symbols not triggered	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.16.00
Problem Description:	
What happens (symptoms):	

The linker aborts with an error message that a runnable function is defined multiple times. No error is reported during generation.	
When does this happen:	

The error is issued by RTE Analyzer during compilation of the code, or any linker that works on the complete project, in case the configuration is as described below.	
In which configuration does this happen:	

This happens when in a system without applications two runnable entities are configured to have the same symbol.	
Resolution Description:	
Workaround:	

Define unique symbol names for the affected runnables.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095769	Validation Error : ConsistencyRT00002 - COM90008 - Model access request failure in ComSignalTimeoutValidatorAs4
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	13.03.00
Problem Description:	
What happens (symptoms):	

Validation error is occurs:	
ConsistencyRT00002 Error at validator runtime (1 message)	
ConsistencyRT00002 Consistency encountered a validationResult with empty CE list[Error]	
COM90008 - Model access request failure	
When does this happen:	

Error may occur while configuring a ComSignal/ ComSignalGroup in following sequence:	
1) Create ComSignal/ ComSignalGroup	
2) Refer created ComSignal/ ComSignalGroup from some other module, for example by /	
MICROSAR/ComM/ComMConfigSet/ComMPnc/ComMPncComSignal/ComMPncComSignalRef	
Issue is only present as long as the created Item has not been referred by a ComIPdu	
In which configuration does this happen:	

- Any Configuration	
Resolution Description:	
Workaround:	

Complete configuration. A ComSignal/ ComSignalGroup must referred by exactly one ComIPdu. As	
soon as the ComSignal/ ComSignalGroup is referred by a ComIPdu,	
the error will disappear.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095785 RTE49999 error message: "Invalid data type mapping" for mapped DataType in CalibrationPort PortInterfaces	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.16.00
Problem Description: What happens (symptoms): ----- Configuration tool reports an "ApplicationDataType <DataTypeName> is not mapped to an ImplementationDataType" message for a DataType that is mapped. When does this happen: ----- This happen during calculation phase. In which configuration does this happen: ----- This happens for configuration that have connected CalibrationPorts and both ports uses same DataTypes but from different DataType Packages in PortInterface.	
Resolution Description: Workaround: ----- Use the same data types for the calibration SWC and the connected SWCs. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00095786 Wrong generated limits for CharacteristicTables in Rte.a2l when physical constraints are configured	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.16.00
Problem Description:	
What happens (symptoms):	

Wrong upper and lower limits are generated into Rte.a2l file.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains a characteristic table that references physical data constraints and when the value axis data type specifies a compu method with scaling.	
Resolution Description:	
Workaround:	

Either configure upper and lower limit for the compu scale or specify internal limits instead of physical limits in the data constraints.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095808 RTE49999 when mode disablings are used in a task with schedule tables	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.05.00
Fixed in versions:	1.16.00
Problem Description: What happens (symptoms): ----- RTE generation aborts with an error message RTE49999 An unexpected error occurred. (1 message) RTE49999 An unexpected error occurred. When does this happen: ----- During generation. In which configuration does this happen: ----- This happens when the configuration contains OS applications and when the configuration contains a basic task with multiple runnables that use different cycle times and when one of the runnables uses mode disablings.	
Resolution Description: Workaround: ----- Map the runnable entity with the mode disabling to a separate tasks or remove the mode disabling (the mode can be checked inside the runnable). Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00095840	[Only in case of using a PN-CanTrcv] ConsistencyRT00002 - Error at validator runtime: CanTrcvPnConfigurationValidator
Component@Subcomponent:	DrvTrans__baseCanxAsr@GenTool_GeneratorMsr
First affected version:	1.02.00
Fixed in versions:	3.03.01
Problem Description:	

ESCAN00095840**[Only in case of using a PN-CanTrcv]****ConsistencyRT00002 - Error at validator runtime:
CanTrcvPnConfigurationValidator**

What happens (symptoms):

After Import (CFG5 -> File -> Import -> using the "Import Mode" = "Replace") of a CanTrcv configuration the following error occurs within the Validation-view of CFG5:

[Error] ConsistencyRT00002 - Error at validator runtime

- Consistency encountered a validationResult with empty CE list [Error] CANTRCV90008 - Model access request failure

- The element was already removed. ElementInfo: MDFObject already removed. /ActiveEcuC/CanTrcv/CanTrcvConfigSet/CanTrcvChannel DefinitionRef: /MICROSAR/CanTrcv_Tja1145/CanTrcv/CanTrcvConfigSet/CanTrcvChannel.

This is not a configuration problem but an internal implementation error. Please contact Vector for support.

Validator-Class:

com.vector.cfg.gen.DrvTrans__baseCanxAsr.validation.CanTrcvPnConfigurationValidator Validator-Description: Validates the configuration for selective wakeup

Further runtime errors of this validator won't be reported in the UI.

We are sorry, but due to this internal error, code generation of /MICROSAR/CanTrcv_Tja1145/CanTrcv has to be blocked. As a workaround, you may try to restart DaVinci Configurator.

Otherwise, please call Vector for support.

Erroneous CEs:

[DefinitionRef: /MICROSAR/CanTrcv_Tja1145/CanTrcv]

CEs:

[DefinitionRef: /MICROSAR/CanTrcv_Tja1145/CanTrcv]

AutoSolvingAction: <none>

PreferredSolvingAction: <none>

SolvingActions: <none>

Variants: [General][UnfilteredInvariantProjectModelView]

When does this happen:

After Import (CFG5 -> File -> Import -> using the "Import Mode" = "Replace") of a CanTrcv configuration

In which configuration does this happen:

If a PN (== Partial Networking) CanTrcv is used e.g. Tja1145, E52013, Tle9255

AND

the "Import Mode" = "Replace" is used

OR

a "CanTrcvChannel" is just deleted

Resolution Description:

ESCAN00095840	[Only in case of using a PN-CanTrcv] ConsistencyRT00002 - Error at validator runtime: CanTrcvPnConfigurationValidator
<p>Workaround:</p> <p>-----</p> <p>This issue is not critical, please close the whole configuration including the CFG5 and open your configuration once again.</p> <p>Resolution:</p> <p>-----</p> <p>The described issue is corrected by modification of all affected work-products.</p>	
ESCAN00095854	Configuration tool reports Rte90005 java.lang.NullPointerException during validation phase
<p>Component@Subcomponent: Rte_Asr4@GenTool_GeneratorMsr</p> <p>First affected version: 4.14.00</p> <p>Fixed in versions: 4.16.00</p>	
<p>Problem Description:</p> <p>What happens (symptoms):</p> <p>-----</p> <p>Configuration tool reports Rte90005 java.lang.NullPointerException.</p> <p>When does this happen:</p> <p>-----</p> <p>This happens during validation phase.</p> <p>In which configuration does this happen:</p> <p>-----</p> <p>This happens for configuration with incomplete data prototype mappings that are defined but not used any more.</p>	
<p>Resolution Description:</p> <p>Workaround:</p> <p>-----</p> <p>Complete all data prototype mappings, even if they are not used. OR Delete all unused incomplete data prototype mappings in configuration.</p> <p>Resolution:</p> <p>-----</p> <p>The described issue is corrected by modification of all affected work-products.</p>	

ESCAN00095891	Auto-Configuration - SDLC: Wizard leads to validation errors in case of unconfigured channels
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	8.01.02
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

The Cfg5 shows the following error:	
AR-ECUC02008 Mandatory parameter ComMUserChannel is missing in	
SDLC_BswM_Backbone_<CHANNELNAME>.	
ComMUserChannel	
/ActiveEcuC/ComM/ComMConfigSet/DIAG_1_CAN/SDLC_BswM_Backbone_<CHANNELNAME>	
When does this happen:	

After execution of the configuration wizard of the 'Auto Configuration: Smart DLC'.	
In which configuration does this happen:	

Only if not all channels are selected in the BswM configuration wizard of the 'Auto Configuration: Smart DLC'.	
Resolution Description:	
Workaround:	

Remove the not needed container SDLC_BswM_Backbone_<CHANNELNAME> from the channel	
which causes this error.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095900 RTE Analyzer reports false out of bounds access	
Component@Subcomponent:	GenTool_IRAnalyzer@Application
First affected version:	0.07.00
Fixed in versions:	1.00.01
Problem Description:	
What happens (symptoms):	

RTE Analyzer reports Found 1 findings of category: 12002 Potential out of bounds write although the access is not out of bounds.	
When does this happen:	

During analysis.	
In which configuration does this happen:	

This happens when the generated code contains memcpy accesses that copy to byte arrays that are elements of other data structures and when the byte array is not the last element in the data structure.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095937 RTE49999 when a curve use a value data type with texttable compu method	
Component@Subcomponent:	Rte_Asr4@Generator
First affected version:	4.11.00
Fixed in versions:	4.16.00
Problem Description: What happens (symptoms): ----- RTE Generation aborts with an "RTE49999 Object reference not set to an instance of an object" error message. When does this happen: ----- During generation. In which configuration does this happen: ----- This happens when the configuration contains curves for which the value axis data type contains a texttable compu method.	
Resolution Description: Workaround: ----- Configure a linear compu method. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00095940 Missing check for UINT8_DYN signals when creating the Gw-Routing Key**Component@Subcomponent:** Il_AsrComCfg5@GenTool_GeneratorMsr**First affected version:** 12.00.00**Fixed in versions:** 13.03.01**Problem Description:**

What happens (symptoms):

In signal based gateway configuration, a compilation error may occur, if dynamic length signals/ group signals are used.

Further, memory (RAM) is wasted, as memory is reserved for UINT8_DYN signals/groupSignals in purpose of gateway routing, although routing of dynamic length signals / group signals is not supported.

When does this happen:

At compile time, a compilation error will occur in below mentioned case.

At run time, waste of RAM.

In which configuration does this happen:

Compilation error:

In all configurations which contain gateway mappings and no signals or groupSignals with appl type:

ComSignalType = UINT8_N

and at least one signal or groupSignal has an appl type configured to:

ComSignalType = UINT8_DYN

Waste of RAM:

In all configurations which contain gateway mappings and dynamic length signals/ group signals.

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095942 Undefined a2l data type in record layout	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.11.00
Fixed in versions:	1.16.00
Problem Description:	
What happens (symptoms):	

The record layout in Rte.a2l file looks like:	
/begin RECORD_LAYOUT RTE_RL_<name>	
FNC_VALUES 1 UNDEFINED ROW_DIR DIRECT	
STATIC_RECORD_LAYOUT	
/end RECORD_LAYOUT	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains curves mapped to array data types with native delcaration and when the	
native declaration of base type is not: uint8, uint16, uint32, uint64, sint8, sint16, sint32, sint64,	
float32, float64 float80, boolean or string.	
Resolution Description:	
Workaround:	

Set the native declaration to a platform type.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095944 Missing compu method in A2L**Component@Subcomponent:** Rte_Core@Implementation**First affected version:** 1.11.00**Fixed in versions:** 1.16.00**Problem Description:**

What happens (symptoms):

A characteristic object references a compu method that does not exist

When does this happen:

During generation.

In which configuration does this happen:

This happens when a map or curve is configured and no compu method is configured on application data type level but on implementation data type level.**Resolution Description:**

Workaround:

Manually define the missing compu method in the A2L that includes Rte.a2l.

Resolution:

The described issue is corrected by modification of all affected work-products.**ESCAN00096004 Compiler error: NON RTE USE CASE - Redeclared typedef Csm_ReturnType****Component@Subcomponent:** SysService_AsrCsm@Implementation**First affected version:** 2.02.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

Compiler warning/error may occur due to duplicated definition of Csm_ReturnType , e.g.:
Redeclared typedef Csm_ReturnType

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

In configuration without RTE.**Resolution Description:**

Workaround:

Define the macro Rte_TypeDef_Csm_ReturnType as well as the type definition within a global header.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096021 [Error] ConsistencyRT00002 - Error at validator runtime: CanIfTxBufferSupportValidator

Component@Subcomponent: If_AsrIfCan@GenTool_GeneratorMsr

First affected version: 3.04.00

Fixed in versions: 4.06.00

Problem Description:

What happens (symptoms):

After loading of configuration the following error occurs within the "Validation" view:

[Error] ConsistencyRT00002 - Error at validator runtime

- Consistency: an exception was caught while executing onModelEvent() of a validator.

Configuration inconsistencies couldn't be reported by this validator.ModelView:InvariantValuesModelView

This is not a configuration problem but an internal implementation error. Please contact Vector for support.

Validator-Class:

com.vector.cfg.gen.If_AsrIfCan.validation.TxValidators.TxBufferValidators.CanIfTxBufferSupportVali

Validator-Description:Setting control of features: "CanIfPublicTxBuffering,

"CanIfMultipleBasicCANTxObjects" and "CanIfCtrlDrvTxCancellation".

Further runtime errors of this validator won't be reported in the UI.

Ex: com.vector.cfg.model.exceptions.InvisibleVariantObjectFeatureException: Cannot read

invisible feature: Source: /ActiveEcuC/CanIf/CanIfCtrlDrvCfg/DUT_CH0{Variant0} (DefRef: /

MICROSAR/CanIf/CanIfCtrlDrvCfg/CanIfCtrlCfg) View: InvariantValuesModelView Feature: parent

We are sorry, but due to this internal error, code generation of /MICROSAR/CanIf/

CanIfCtrlDrvCfg, /MICROSAR/CanIf/CanIfInitCfg/CanIfBufferCfg, /MICROSAR/CanIf/

CanIfPrivateCfg, /MICROSAR/CanIf/CanIfInitCfg/CanIfTxPduCfg/CanIfTxPduBufferRef, /MICROSAR/

CanIf/CanIfInitCfg/CanIfBufferCfg/CanIfBufferHthRef, /[ANY]/Can/CanConfigSet/

CanHardwareObject/CanHandleType, /MICROSAR/CanIf/CanIfPublicCfg/CanIfPublicTxBuffering, /

MICROSAR/CanIf/CanIfPrivateCfg/CanIfMultipleBasicCANTxObjects, /MICROSAR/CanIf, /

MICROSAR/CanIf/CanIfPublicCfg, /MICROSAR/CanIf/CanIfCtrlDrvCfg/CanIfCtrlDrvTxCancellation, /

MICROSAR/CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg, /[ANY]/Can/CanConfigSet/

CanHardwareObject, /MICROSAR/CanIf/CanIfCtrlDrvCfg/CanIfCtrlCfg, /MICROSAR/CanIf/

CanIfInitCfg/CanIfTxPduCfg, /MICROSAR/CanIf/CanIfInitCfg/CanIfBufferCfg/

CanIfTxBufferHandlingType, /MICROSAR/CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg/

CanIfHthCanCtrlIdRef, /MICROSAR/CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg/

CanIfHthIdSymRef has to be blocked. As a workaround, you may try to restart DaVinci

Configurator. Otherwise, please call Vector for support

-> BUT: This error occurs also after restart of Configurator 5.

When does this happen:

After loading of configuration

In which configuration does this happen:

--> Please note: If you don't get such error message then your configuration is not affected by this issue at all and you may ignore this issue.

In case of a configuration with multiple (> 1) variants [identities] especially in CAN-cluster:

CanDrv, CanIf... etc.

-> Please note: In such case "Postbuild-Selectable Support" is enabled!

Resolution Description:

ESCAN00096021		[Error] ConsistencyRT00002 - Error at validator runtime: CanIfTxBufferSupportValidator	
Workaround:			

No workaround available.			
Resolution:			

The described issue is corrected by modification of all affected work-products.			
ESCAN00096024		Incorrect parameter in API Dem_PostRunRequested	
Component@Subcomponent:	Diag_Asr4Dem@Doc_TechRef		
First affected version:	1.00.00		
Fixed in versions:	7.05.00		
Problem Description:			
What happens (symptoms):			

In description of API Dem_PostRunRequested() input parameter 'IsRequested' is not marked as pointer.			
When does this happen:			

always			
In which configuration does this happen:			

all configurations			
Resolution Description:			
Workaround:			

No workaround available.			
Resolution:			

The described issue is corrected by modification of all affected work-products.			

ESCAN00096026 RteAnalyzer reports incompatible pointer type passing for inter-runnable variables with multi-dimensional arrays

Component@Subcomponent: Rte_Core@Implementation

First affected version: 1.14.00

Fixed in versions: 1.16.00

Problem Description:

What happens (symptoms):

Analysis with RTE Analyzer fails because the RTE is unable to generate stubs that contain the correct variable for inter-runnable variable name with multi-dimensional arrays.

When does this happen:

The warning is issued by the compiler during the compilation of the code in case the configuration is as described below.

In which configuration does this happen:

This happens when the configuration contains inter-runnable variables with multi-dimensional arrays.

Resolution Description:

Workaround:

Edit the analyzer stub source file so that it passes a pointer to the array base type.

Eg:

TSC_SUM_SSC_Rte_IrvRead_SUM_SSC_Rx_QueueProcess_Runnable_SUM_SSC_RxQueue_Counter

This should become:

TSC_SUM_SSC_Rte_IrvRead_SUM_SSC_Rx_QueueProcess_Runnable_SUM_SSC_RxQueue_Counter

Fix this for both inter-runnable variables read and write.

Please note that needs to be done before every analysis with RTE Analyzer as the file gets overwritten when the RTE is generated again.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096061 Auto-Configuration - SDLC: Support of CAN FD PDUs is not working	
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	10.01.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

The SDLC configuration wizard does not provide the possibility to configure routing paths which contain CAN FD PDUs.	
When does this happen:	

Always during configuration.	
In which configuration does this happen:	

In CAN FD configurations.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096126 Com_Init may lead to long interrupt locks	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	10.00.00
Fixed in versions:	13.03.01
Problem Description:	
What happens (symptoms):	

Depending on the amount of configured ComIPdus and ComSignals/ ComGroupSignals calling Com_Init may lead to a long duration where the interrupts are locked. This can lead to some inconsistencies for the Os. However, this issue depends on the used OS and the way how Exclusive Areas are locked.	
When does this happen:	

At runtime, when Com_Init() is called.	
In which configuration does this happen:	

Large amount of ComIPdus and ComSignals/ ComGroupSignals are configured.	
Resolution Description:	
Workaround:	

Manually implement the critical sections belonging to SchM_Enter_Com_COM_EXCLUSIVE_AREA_RX, SchM_Enter_Com_COM_EXCLUSIVE_AREA_TX in dependency to the initialization state of the Com.	
If Com is not initialized, the critical section shall do nothing, otherwise the critical section should operate in normal fashion.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096128 Generator Exception in case the PduRSrcPdu global Pdu is referenced by more than one other container	
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	11.01.00
Fixed in versions:	12.00.00, 11.01.02
Problem Description:	
What happens (symptoms):	

A java exception occurs while generating the PduR: PDUR90005 Exception in PduR generator during Validation encountered: java.lang.IllegalArgumentException: Handle ID of the src module is not unique (=java.util.stream.ReferencePipeline\$Head@67fe09f1)! /ActiveEcuC/PduR	
When does this happen:	

If you generate the PduR code.	
In which configuration does this happen:	

The exception occurs if the global Pdu of a PduRSrcPdu (PduRSrcPduRef) is referenced by more than one other module/container. E.g. if two different SoAdSocketRoutes reference this global Pdu.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products. Beware that there is still an exception implemented for PduRDestPdu global Pdu references which are referenced by more than one other container. This is a wrong configuration and not supported.	

ESCAN00096129 MEMMAP_SW_MINOR_VERSION / MEM_SW_MINOR_VERSION is not correct	
Component@Subcomponent:	CommonAsr__Common@Impl__MemMap
First affected version:	1.10.00
Fixed in versions:	1.11.00
Problem Description: What happens (symptoms): ----- The defines: MEMMAP_SW_MINOR_VERSION / MEM_SW_MINOR_VERSION [see file: MemMap.h] have an incorrect value "9" but the value must be "10". ----- When does this happen: ----- - Always in case of taking a look on version ----- In which configuration does this happen: ----- - In all configurations, is independent of configuration -----	
Resolution Description: Workaround: ----- Please see the history for correct version. ----- Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096177 RTE49999 when no trigger is assigned to a server operation**Component@Subcomponent:** Rte_Core@Implementation**First affected version:** 1.04.00**Fixed in versions:** 1.16.00**Problem Description:**

What happens (symptoms):

RTE generator prints a RTE49999 error message.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the configuration contains client-server calls and when the server provides no trigger and server runnable for the server operation.**Resolution Description:**

Workaround:

Configure a trigger and a server runnable.

Resolution:

The described issue is corrected by modification of all affected work-products.**ESCAN00096187 RTE49999 when the base type on network representation has no size****Component@Subcomponent:** Rte_Core@Implementation**First affected version:** 1.15.00**Fixed in versions:** 1.16.00**Problem Description:**

What happens (symptoms):

RTE generator reports a RTE49999 error message.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the network representation references no base type.**Resolution Description:**

Workaround:

Configure a valid base type and size on the network representation.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096299 **During generation of code the error occurs:
"CANTRCV01009 No valid wakeup source specified."****Component@Subcomponent:** DrvTrans__baseCanxAsr@GenTool_GeneratorMsr**First affected version:** 2.01.00**Fixed in versions:** 3.00.02, 3.03.01**Problem Description:**

What happens (symptoms):

During generation of configuration the following error occurs:

CANTRCV01009 No valid wakeup source specified. (1 message)

CANTRCV01009 Invalid wakeupsource configured for 'CanTrcvWakeupSourceRef' on Channel
'CanTrcvChannel_CAN02'/ActiveEcuC/CanTrcv/CanTrcvConfigSet/CanTrcvChannel_CAN02[CanTrcvWakeupSourceRef]
{CV_2}

When does this happen:

(1) Always and immediately -> during generation of code

In which configuration does this happen:

- Multiple CAN transceiver channels are configured [Container: CanTrcvChannel]

AND

- At least within one CAN transceiver channel the parameter "CanTrcvWakeupByBusUsed" is
enabled

AND

- but there is at least one CAN transceiver channel with disabled parameter
"CanTrcvWakeupByBusUsed" and there is no wakeup source configured for this channel
[parameter: CanTrcvWakeupSourceRef]

-> In such case for each such CAN transceiver the generation error [see above] occurs

Resolution Description:

Workaround:

Please create a dummy wakeup source and reference it via the parameter:

"CanTrcvWakeupSourceRef".

-> Afterwards please generate once again

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096311 Validation message "CANIF30001 At least one BasicCAN Tx-PDU is configured. Hence it is recommended to enable the Tx-buffer..." leads to invalid configuration

Component@Subcomponent: If_AsrIfCan@GenTool_GeneratorMsr

First affected version: 4.00.00

Fixed in versions: 4.06.01

Problem Description:

What happens (symptoms):

The following validation message leads to invalid configuration:

CANIF30001 Activation of feature is recommended. (1 message)

CANIF30001 At least one BasicCAN Tx-PDU is configured. Hence it is recommended to enable the Tx-buffer, in order to avoid re-trying of Tx-requests by CAN upper layers.

Enable the Tx-buffer: Enable parameter "CanIfPublicTxBuffering".

/ActiveEcuC/Can/CanConfigSet/ABS_CHHS_4dfd5980_Tx_Std[CanHandleType]

/ActiveEcuC/CanIf/CanIfPublicCfg[CanIfPublicTxBuffering]

especially in case of implementation variant: VARIANT-PRE-COMPILE a

CANIF90005 Generator (MICROSAR Can Interface Generator) failure, because of an exception (1 message)

CANIF90005 Exception in CanIf generator during Validation encountered:

java.util.NoSuchElementException

/ActiveEcuC/CanIf

occurs in case of no Tx-PDU is mapped to configured Tx-BasicCAN-HW-Objects [A "Tx-BasicCAN-HW-Object" is from the configuration point of view: a "CanHardwareObject"-object of type "CanObjectType == TRANSMIT" and "CanHandleType == BASIC"]

When does this happen:

- During the configuration using the Configurator 5 / generation of code

In which configuration does this happen:

In case of in your configuration no Tx-PDUs are mapped to the Tx-BasicCAN-HW-Objects [see above]. In such case you get in addition the following validation message [e.g.]:

CANIF20004 Unused CAN hardware Tx-object is configured. (1 message)

CANIF20004 No Tx-PDUs are mapped to object: "CanIfBufferCfg_Tx_Std_ad992627". Hence the configured Tx-hardware object(s): "CanIfBufferCfg_Tx_Std_ad992627",

"ABS_CHHS_4dfd5980_Tx_Std", "ABS_CHHS_4dfd5980_Tx_Std" are NOT needed and you may remove them from configuration in order to reduce RAM and ROM consumption!

Delete the unneeded Tx-hardware object(s).

/ActiveEcuC/Can/CanConfigSet/ABS_CHHS_4dfd5980_Tx_Std

/ActiveEcuC/CanIf/CanIfInitCfg/CanIfBufferCfg_Tx_Std_ad992627

/ActiveEcuC/CanIf/CanIfInitCfg/CanIfInitHohCfg/ABS_CHHS_4dfd5980_Tx_Std

Resolution Description:

ESCAN00096311	Validation message "CANIF30001 At least one BasicCAN Tx-PDU is configured. Hence it is recommended to enable the Tx-buffer..." leads to invalid configuration
----------------------	--

Workaround:

As already mentioned on "Main" page: In such case you get in addition the following validation message [e.g.]:

CANIF20004 Unused CAN hardware Tx-object is configured. (1 message)

CANIF20004 No Tx-PDUs are mapped to object: "CanIfBufferCfg_Tx_Std_ad992627". Hence the configured Tx-hardware object(s): "CanIfBufferCfg_Tx_Std_ad992627", "ABS_CHHS_4dfd5980_Tx_Std", "ABS_CHHS_4dfd5980_Tx_Std" are NOT needed and you may remove them from configuration in order to reduce RAM and ROM consumption!

Delete the unneeded Tx-hardware object(s).

/ActiveEcuC/Can/CanConfigSet/ABS_CHHS_4dfd5980_Tx_Std

/ActiveEcuC/CanIf/CanIfInitCfg/CanIfBufferCfg_Tx_Std_ad992627

/ActiveEcuC/CanIf/CanIfInitCfg/CanIfInitHohCfg/ABS_CHHS_4dfd5980_Tx_Std

Hence you have multiple workaround options:

either

- Please execute the solving action of validation message mentioned above [CANIF20004 delete the obsolete objects] and do not enable the parameter "CanIfPublicTxBuffering" [see above]]

or

- Map at least one Tx-PDU to the Tx-BasicCAN-HW-Object

or

- Ignore the validation message to enable the parameter "CanIfPublicTxBuffering", set this parameter to "false" and set this parameter to "user defined".

Resolution:

The described issue is corrected by modification of all affected work-products.

**ESCAN00096335 Compiler error: Unknown identifier
Dcm_DemApiNrcMapSelectDTC****Component@Subcomponent:** Diag_Asr4Dcm@Implementation**First affected version:** 7.02.00**Fixed in versions:** 8.05.00**Problem Description:**

What happens (symptoms):

During Dcm.c compilation an error of kind: unknown identifier Dcm_DemApiNrcMapSelectDTC occurs.

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

-
- DCM is configured to interact with a DEM API 4.3.0 (in Dcm_Cfg.h: #define DCM_DEM_API_430_ENABLED == STD_ON)
 - AND
 - DCM does internally support service 0x19
 - AND
 - None of the following sub-functions of SID 0x19 is supported (at all or in DCM): 0x01, 0x02, 0x03, 0x07, 0x08, 0x0A, 0x0F, 0x11, 0x12, 0x13, 0x14, 0x15, 0x17, 0x42;
 - OR
 - DCM does not support (internally) OBD services 0x03, 0x07 and 0x0A.

Resolution Description:

Workaround:

A configuration without sub-functions 0x02 (ReportDTCByStatusMask) or 0x0A (ReportAllSupportedDTCs) is highly unusual. Please check whether the configuration of SID 0x19 is complete and add missing sub-functions.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096391	Compiler error: function "CanHL_WakeupProcessed" / "CanHL_SleepProcessed" was referenced but not defined
Component@Subcomponent:	DrvCan__coreAsr@Implementation
First affected version:	5.02.00
Fixed in versions:	5.07.01, 5.04.05, 5.03.05, 5.05.06, 5.06.01
Problem Description:	
What happens (symptoms):	

Compiler error:	
function "CanHL_WakeupProcessed" / "CanHL_SleepProcessed" was referenced but not defined	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as	
described below.	
In which configuration does this happen:	

Wakeup over CAN is supported by CAN driver	
and	
CanWakeupSupport is not active	
and	
Compiler/Compiler option does not accept Declaration without Definition	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096422 Validation rule inhibits generation in an valid (special) use case

Component@Subcomponent: SysService_Asr4WdM@GenTool_GeneratorMsr

First affected version: 2.00.00

Fixed in versions: 2.01.01

Problem Description:

What happens (symptoms):

A validation rule inhibits the generation of the component if the WdgM is intended to be used on a core with an ID (WdgMModeCoreAssignment) which is greater or equal as the value for the maximum number of cores configured in the OS (OsNumberOfCores).

This use case can be a valid one if not all cores are used as AUTOSAR cores. Then the number of maximum number of cores in OS (OsNumberOfCores) is maybe "1", but 'WdgMModeCoreAssignment' is configured with a value greater than "0". In this configuration core 0 is a Non-AUTOSAR core and core 1 is an AUTOSAR core. This is a valid use case and the generator shall not inhibit the generation, but shall give a warning to double check this configuration if it is really intended.

When does this happen:

If different cores are used at the controller and at least one core is a Non-AUTOSAR core.

This ESCAN applies not if the Non-AUTOSAR core is the core with the highest core id.

In which configuration does this happen:

If parameter "/MICROSAR/WdgM/WdgMConfigSet/WdgMMode/WdgMModeCoreAssignment" is greater or equal than "AUTOSAR/Os/OsOS/OsNumberOfCores".

Resolution Description:

Workaround:

Set the parameter "/MICROSAR/WdgM/WdgMConfigSet/WdgMMode/WdgMModeCoreAssignment" to user defined.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096516 Compiler error: Wrong generated Rte_IocSend calls for queued communication	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.03.00
Fixed in versions:	1.17.00
Problem Description: What happens (symptoms): ----- Compiler reports unresolved symbols, like Rte_IocSend_<Identifier>.	
When does this happen: ----- The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen: ----- This happens for configurations with queued sender/receiver N:1 communication over partition boundaries, when all senders are mapped to the same OsApplication an "Enforce IOC" (Microsar Parameter) is not set.	
Resolution Description: Workaround: ----- Add an addition sender which is mapped to another OsApplication than the other senders. OR Use a PR-Port instead of a R-Port on receiving side. OR Activate "Enforce IOC" parameter.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096521 Compile error on Unix systems due to upper case in include file	
Component@Subcomponent:	SysService_CryptoCv@Impl_actCLib
First affected version:	2.04.00
Fixed in versions:	2.09.00
Problem Description:	
What happens (symptoms):	

In "actEd25519.h", this include is used	
#include "actIED25519.h"	
But the header has the file name "actIEd25519.h"	
This causes a compile error on system with case sensitive file names.	
When does this happen:	

compile time	
In which configuration does this happen:	

compilation is done on a system with case sensitive file names	
Resolution Description:	
Workaround:	

create a header file "actIED25519.h" with this content	
#include "actIEd25519.h"	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

**ESCAN00096532 Compiler error: undeclared identifier
"WDGM_GLOBAL_STATUS_xxx"****Component@Subcomponent:** SysService_Asr4WdM@GenTool_GeneratorMsr**First affected version:** 2.01.00**Fixed in versions:** 2.01.01**Problem Description:**

What happens (symptoms):

Compiler issues an error during compilation of WdgM.c: undeclared identifier
"WDGM_GLOBAL_STATUS_xxx" where xxx can be "STOPPED", "OK", "FAILED" or "DEACTIVATED".

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

-
- If WdgM is configured for single core usage
 - The application referenced in "WdgMGlobalMemoryAppTaskRef" is not referenced at a WdgMAppTaskRef parameter of a supervised entity.

Resolution Description:

Workaround:

Create an additional supervised entity which references the same OS application as "WdgMGlobalMemoryAppTaskRef". Create a WdgMLocalStatusParams for this new supervised entity and set "WdgMSupervisedEntityInitialMode" to "WDGM_LOCAL_STATUS_DEACTIVATED"

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096559 Wrong signal type for data conversion	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.12.00
Fixed in versions:	1.17.00
Problem Description:	
What happens (symptoms):	

The COM module reports that the signal data type is invalid.	
When does this happen:	

After the RTE generator run in the calculation phase.	
In which configuration does this happen:	

This happens when data conversion for COM signals is configured.	
Resolution Description:	
Workaround:	

Set the signal type parameter to user defined and select the value that is expected by COM.	
In case the RTE generator selected SINT24 or UINT24 create type reference data types in the configuration	
uint24 => uint32	
sint24 => sint32	
and select SINT32 or UINT32 in the COM configuration.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096581	PduRTxBuffer references are incorrectly validated for transport protocol 1:N/N:1 routing paths with API forwarding PduRDestPdu
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	11.01.00
Fixed in versions:	13.00.00, 12.01.00
Problem Description:	
What happens (symptoms):	
<p>-----</p> <p>A validation error PDUR10501 is shown incorrectly for PduRDestPdus which are API Forwardings.: PDUR10501 All PduRDestTxBufferRefs of gateway PduRDestPdu have to reference the same PduRTxBuffers for a 1:N/N:1 transport protocol routing path.</p>	
When does this happen:	
<p>-----</p> <p>The error message is always shown for PduRDestPdus mentioned below. The message is shown during live validation in the DaVinci Configurator 5.</p>	
In which configuration does this happen:	
<p>-----</p> <p>The incorrect validation message is shown for either:</p> <ul style="list-style-type: none"> - 1:N transport protocol routings with a PduRDestPdu which is a API Forwarding - N:1 transport protocol routings with a PduRDestPdu which is a API Forwarding 	
<p>The message is only shown if the API Forwarding does not reference the same PduRTxBuffers like the Gateway PduRDestPdu (which is the correct configuration).</p>	
Resolution Description:	
Workaround:	
<p>-----</p> <p>Set the same PduRTxBuffers for the API Forwarding PduRDestPdu as for the Gateway PduRDestPdu.</p>	
<p>Another validation error 'PDUR11500 The PduRDestTxBufferRef PduRDestTxBufferRef(value=XXXXX) must only be configured for gateway routing paths.' is shown. Set the PduRDestTxBufferRef parameter for the API Forwarding PduRDestPdu to user defined. The error will be reduced to a warning. It does not affect the generation of the PduR.</p>	
Resolution:	
<p>-----</p> <p>The described issue is corrected by modification of all affected work-products.</p>	

ESCAN00096582 Non-interactive Mode fails**Component@Subcomponent:** _3rdParty_McalIntegration_Helper@VectorIntegration**First affected version:** 2.02.01**Fixed in versions:** 2.02.03**Problem Description:**

What happens (symptoms):

The non-interactive mode doesn't work: Running the batch file Script_MCAL_Prepare.bat (starts the 3rdParty MCAL Integration Helper tool) with the parameter --auto (providing a derivative) it is expected to finish the tool without user action. Instead a GUI window opens expecting a user action / dialog.

When does this happen:

Starting the 3rdParty MCAL Integration Helper tool via Script_MCAL_Prepare.bat.

In which configuration does this happen:

Any.

Resolution Description:

Workaround:

No workaround available.

ESCAN00096594 C Standard Library is used**Component@Subcomponent:** SysService_CryptoCv@Impl_actCLib**First affected version:** 1.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

limits.h from C Standard Library is used
(in actPlatformTypes.h)

When does this happen:

always

In which configuration does this happen:

always

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096615 RTE Analyzer fails due to duplicated runnable functions	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.17.00
Problem Description:	
What happens (symptoms): ----- Rte Analyzer aborts with error message: ERROR: Linking globals named 'Dem_GetEventUdsStatus': symbol multiply defined! When does this happen: ----- The error is issued by RTE Analyzer during compilation of the code in case the configuration is as described below. In which configuration does this happen: ----- This happens when the configuration contains multiple service components with different names and the same runnable entity symbol. Moreover one service component needs to contain different runnables with the same symbol.	
Resolution Description:	
Workaround: ----- Remove the implementation of the regarded service component runnable from all but one RteAnalyzer stubs. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00096629 Linker error: unresolved symbol error for not existing callout function referenced in Det_Cfg.o in case of disabled DET**Component@Subcomponent:** SysService_AsrDet@GenTool_GeneratorMsr**First affected version:** 10.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

If the DET is disabled and a callout function is configured which does not exist an unresolved symbol error is thrown by the linker.

When does this happen:

The error is issued by the linker during linking of the code in case the configuration is as described below.

In which configuration does this happen:

The issue occurs if all of the following conditions apply:

1) The DET is disabled by setting DetEnableDet = false

AND

2) One or more callout functions are configured, e.g. DetErrorHook, DetReportRuntimeErrorCallout or DetReportTransientFaultCallout

AND

3) At least one of the configured callout functions does not exist

Resolution Description:

Workaround:

The following workarounds are possible:

1) In order to disable the DET remove it from the configuration.

2) Don't link Det_Cfg.o in case DET is disabled in your configuration.

3) Provide the configured callout functions also for a disabled DET.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096748 Nonexistent "TimerSTMin" struct member in a2l file**Component@Subcomponent:** Tp_Asr4TpCan@GenTool_GeneratorMsr**First affected version:** 3.01.00**Fixed in versions:** 4.01.02**Problem Description:**What happens (symptoms):

In the generated a2l file, the following symbol name can be found:

CanTp_TxState[0].TimerSTMin

When performing an update using the generated a2l file, a warning is issued stating that the symbol address couldn't be updated.

The CanTp continues to work properly. Only the debugging information related to the STmin is not available.

When does this happen:

Always and immediately.

In which configuration does this happen:

This happens only in configurations where the version 2.01.01 or higher of the CanTp implementation is used. (The implementation version can be found in the CanTp.h file)

Resolution Description:Workaround:

Manually edit the generated a2l file and change all occurrences of "TimerSTMin" to "STminTimer".

Resolution:

A resolution is not yet available.

ESCAN00096774 Compiler error: Duplicated variable definitions in analyzer stubs	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.17.00
Problem Description:	
What happens (symptoms):	

Compilation in RTE Analyzer fails because the analyzer stubs contain multiple variable declarations with the same name.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when the indirect API is configured.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096900 Compiler error: identifier EcuM_Get<***> not declared

Component@Subcomponent: SysService_Asr4EcuM@GenTool_GeneratorMsr

First affected version: 8.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

The compiler throws one of the following errors:

Compiler error: identifier EcuM_GetValidationTimeoutTable not declared
 Compiler error: identifier EcuM_DecValidationTimeoutTable not declared
 Compiler error: identifier EcuM_SetValidationTimeoutTable not declared
 Compiler error: identifier EcuM_GetReasonOfWakeupSourceList not declared
 Compiler error: identifier EcuM_GetChannelOfWakeupSourceList not declared

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

In variant configurations

AND

At least one variant don't use parameter EcuMValidationTimeout | EcuMComMChannelRef | EcuMResetReasonRef (Value = 0 or not existent)

AND

Another variant use parameter EcuMValidationTimeout | EcuMComMChannelRef | EcuMResetReasonRef

Resolution Description:

Workaround:

Ensure that the parameters EcuMValidationTimeout | EcuMComMChannelRef | EcuMResetReasonRef are existent in all variants OR are not existent in all variants. But the existence for each of them has to be consistent over all variants.

It is sufficient to configure a dummy wakeup source which is not used by the code to ensure this.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096969 Misleading function return code description for API Dcm_GetTesterSourceAddress	
Component@Subcomponent:	Diag_Asr4Dcm@Doc_TechRef
First affected version:	7.02.00
Fixed in versions:	9.03.00
Problem Description:	
What happens (symptoms):	

API Dcm_GetTesterSourceAddress always returns E_OK, even if the TesterSourceAddress has an invalid value.	
When does this happen:	

At runtime, if invalid function arguments are passed (e.g. invalid DcmRxPduId, NULL pointer for the return value etc.)	
In which configuration does this happen:	

This happens in all configurations where	
- Dev error detect is disabled with Dcm/DcmConfigSet/DcmGeneral/DcmDevErrorDetect (DCM_DEV_ERROR_DETECT == STD_OFF)	
Resolution Description:	
Workaround:	

Consider the E_NOT_OK return value to be returned only under following condition DCM_DEV_ERROR_DETECT == STD_ON.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096982 AssertionError: The getMaxUnsignedValueForNumBytes utility allows only up to 4 bytes!**Component@Subcomponent:** Diag_Asr4Dcm@GenTool_GeneratorMsr**First affected version:** 1.02.00**Fixed in versions:** 9.00.00**Problem Description:**

What happens (symptoms):

The DCM generator invokes assertion with message: "The getMaxUnsignedValueForNumBytes utility allows only up to 4 bytes!"

When does this happen:

Each and every time DCM configuration shall be generated.

In which configuration does this happen:

- One of the following diagnostic services is to be handled within DCM: 0x23 (ReadMemoryByAddress), 0x2C 0x02 (DynamicallyDefineIdentifier by MemoryAddress) or 0x3D (WriteMemoryByAddress) (in Dcm_Cfg.h: DCM_SVC_XX_SUPPORT_ENABLED == STD_ON for any XX = {23, 2C, 3D})

AND

- The memory layout shall not be capable of supporting MID (MemoryID) (ECUC parameter: /Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspUseMemoryId == FALSE)

AND

- The user has defined a custom ALFID (AddressAndLengthFormatIdentifier) values, with more than four byte address (ECUC parameter: /Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryAddressAndLengthFormatIdentifier/DcmDspMemorySupportedAddressAndLengthFormatIdentifier has any values of the kind: 0x25, 0x45 or any 0xX5).

Note: The affected configuration is invalid and shall be rejected by a validation followed by a meaningful explanation.

Resolution Description:

Workaround:

Since the affected configuration is invalid i.e. without MID the maximum address size cannot exceed four bytes, either enable MID support or reduce the address size.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097053 Compiler error: Empty struct DCM_DIDMGROPTYPE_READCONTEXTTYPE_TAG	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	4.01.00
Fixed in versions:	9.01.00
Problem Description: What happens (symptoms): ----- During Dcm.c compilation with VC compiler following error occurs: error C2016: C requires that a struct or union has at least one member When does this happen: ----- The error is issued by the compiler during compilation of the code in case the configuration is as described below. In which configuration does this happen: ----- - Service 0x22 (ReadDataByIdentifier) is active and handled by DCM AND - DCM is configured to support DIDs with multiple signals (in Dcm_Cfg.h: #define DCM_DIDMGR_MULTISIGNAL_ENABLED == STD_ON) AND - None of the multi signal DIDs support read operation (in Dcm_Cfg.h: #define DCM_DIDMGR_MSIG_OPTYPE_READ_ENABLED == STD_OFF) AND - None of single signal DIDs does support paged read access (in Dcm_Cfg.h: #define DCM_DIDMGR_OPCLS_READ_PAGED_ENABLED == STD_OFF)	
Resolution Description: Workaround: ----- Split a DID with a read access having a single signal into two data signals. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097056 Compiler error: 'Offset' : is not a member of 'DCM_DIDMGROPTYPE_READCONTEXTTYPE_TAG'	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	4.01.00
Fixed in versions:	9.01.00
Problem Description: What happens (symptoms): ----- During Dcm.c compilation with VC compiler following error occurs: error C2039: 'Offset' : is not a member of 'DCM_DIDMGROPTYPE_READCONTEXTTYPE_TAG' When does this happen: ----- The error is issued by the compiler during compilation of the code in case the configuration is as described below. In which configuration does this happen: ----- - Service 0x22 (ReadDataByIdentifier) is active and handled by DCM AND - DCM is configured to support DIDs with multiple signals (in Dcm_Cfg.h: #define DCM_DIDMGR_MULTISIGNAL_ENABLED == STD_ON) AND - None of the multi signal DIDs support read operation (in Dcm_Cfg.h: #define DCM_DIDMGR_MSIG_OPTYPE_READ_ENABLED == STD_OFF) AND - At least one of DIDs does support paged read access (in Dcm_Cfg.h: #define DCM_DIDMGR_OPCLS_READ_PAGED_ENABLED == STD_ON)	
Resolution Description: Workaround: ----- Split a DID with a read access having a single signal into two data signals. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097073 Sub-chapter "ResponseOnEvent Specifics" seems to be part of chapter "Handling with DID Ranges"	
Component@Subcomponent:	Diag_Asr4Dcm@Doc_TechRef
First affected version:	5.01.00
Fixed in versions:	9.03.00
Problem Description:	
What happens (symptoms): ----- Sub-chapter "ResponseOnEvent Specifics" seems to be part of chapter "Handling with DID Ranges". Actually there is missing the corresponding parent chapter "DCM AR Version Specific Features" the RoE specifics belongs to.	
When does this happen: ----- Reading the technical reference.	
In which configuration does this happen: ----- N/A	
Resolution Description:	
Workaround: ----- Consider this chapter as a stand alone.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097086 Compiler error: Undefined reference to `Com_GetTxFilterInitValueArrayBasedFilterInitValue`	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	9.00.00
Fixed in versions:	14.00.00
Problem Description:	
What happens (symptoms):	

Compile error occurs in Function Com_TxSignal_EvaluateFilter: Undefined reference to `Com_GetTxFilterInitValueArrayBasedFilterInitValueLengthOfTxSigInfo'	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This error will occur in configuration, where	
- Rx UINT8_N Signal/ Group Signal with a Filter exists (NEVER MASKED_NEW_DIFFERS_MASKED_OLD)	
AND	
- At least one Tx Signal /Group Signal with (Application Type != UINT8_N) with ((Filter != ALWAYS) or (transferProperty == TRIGGERED_ON_CHANGE TRIGGERED_ON_CHANGE_WITHOUT_REPETITION)) exists	
AND	
- all Tx Signal /Group Signals with (Application Type == UINT8_N) are configured without any Filter and without any OnChange-TransferProperty	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097087 Null pointer exception when a data element is missing	
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.08.00
Fixed in versions:	4.17.00
Problem Description:	
What happens (symptoms): ----- RTE generation aborts with a null pointer exception when a connection connects two ports with incompatible interfaces and when no port interface mapping is configured.	
When does this happen: ----- During generation.	
In which configuration does this happen: ----- This happens when	
Resolution Description:	
Workaround: ----- Rename the data element or configure a port interface mapping.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097092 Compiler error: Identifier "sint"<X> is undefined**Component@Subcomponent:** Il_AsrComCfg5@Implementation**First affected version:** 11.00.00**Fixed in versions:** 13.03.02, 14.00.00**Problem Description:**

What happens (symptoms):

Compiler error may occur, if the platform (compiler and/or controller) does not support signed integer data types {sint8, sint16,sint32, sint64}, i.e. the corresponding data types are not available in Platform_Types.h}:

"Identifier "sint"<X> is undefined."

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

Any configuration in combination with a compiler / compiler option which results in not supporting any of the above mention signed types.

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097096 Compiler error: incompatible pointer type / loss of volatile qualifier in pointer cast	
Component@Subcomponent:	DrvCan__coreAsr@Implementation
First affected version:	5.02.00
Fixed in versions:	6.00.00, 5.07.02
Problem Description:	
What happens (symptoms):	

loss of volatile qualifier by using VStdLib API VStdMemCpy().	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

MicroSar4 only: (not relevant for MSR3x)	
and	
CAN_FD_SUPPORT == CAN_FULL	
and	
CAN_RX_QUEUE == STD_ON	
and	
Compiler throw this as error (some will throw this as warning)	
Resolution Description:	
Workaround:	

Deactivate RX_QUEUE	
or	
(reduce warning/error level of compiler / deactivate this check --> no real problem!)	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097143 WdgMLocalStateChangeCbK not created for all Supervised Entities	
Component@Subcomponent:	SysService_Asr4WdM@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	2.01.01
Problem Description:	
What happens (symptoms): ----- After creating a LocalStatusChangeCbK for a Supervised Entity, this one doesn't appear as service Port under Service Components - Prot Prototypes. When does this happen: ----- If more than one Supervised Entity is created on the same Core and if they references different OsApplications. Only those LocalStatusChangeCbK are added to Service Components which have the same OsApplication referenced as referenced from corresponding "WdgMMode/WdgMGlobalMemoryAppTaskRef". In which configuration does this happen: ----- All configuration with more than one SE on the same Core with different referenced OsApplication as referenced from corresponding "WdgMMode/WdgMGlobalMemoryAppTaskRef"	
Resolution Description:	
Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

**ESCAN00097148 WdgMGlobalStateChangeCbK /
WdgMLocalStateChangeCbK function prototype
generated with incompatible signature compared to
RTE****Component@Subcomponent:** SysService_Asr4WdM@GenTool_GeneratorMsr**First affected version:** 2.00.00**Fixed in versions:** 2.02.00**Problem Description:**

What happens (symptoms):

The WdgMGlobalStateChangeCbK / WdgMLocalStateChangeCbK function prototype gets generated by the WdgM generator (in WdgM_PBcfg.h) even if the Service Port is connected and the Server Runnable is created and the prototype is already generated by Rte (in Rte_WdgM_<Application name referenced from WdgMGlobalMemoryAppTaskRef>.h).

In addition the signature of the prototypes of WdgM does not match the prototypes defined by the Swc template. The Rte generates Std_ReturnType as return value, the WdgM void. This is because the swc-files contains the erroneous return value (E_NOT_OK) instead of no return value.

When does this happen:

At compiling time.

In which configuration does this happen:

When WdgMGlobalStateChangeCbK / WdgMLocalStateChangeCbK is configured and Rte is used (WdgMUseRte==true).

Resolution Description:

ESCAN00097148 WdgMGlobalStateChangeCbK / WdgMLocalStateChangeCbK function prototype generated with incompatible signature compared to RTE

Workaround:

Follow the description of parameter /MICROSAR/WdgM/WdgMConfigSet/WdgMMode/WdgMGlobalStateChangeCbK and /MICROSAR/WdgM/WdgMGeneral/WdgMSupervisedEntity/WdgMLocalStateChangeCbK of BSWMD version 6.01.01.

The following description are in this version:

Callback function for notifying Watchdog Manager global status change.

Note:

If WdgMUseRte is configured as true and thus the receiver of the callback is an application above the RTE, the the user has to configure this parameter with the name of a C function which shall be called by WdgM. Within a separate file where this function is defined the user has to include the corresponding Rte application header file and call the Rte function. The Rte function is named according the following convention:

Rte_Call_WdgM_<OsApplicationName>_globalStateChangeCbK_Core<CoreAssignment>_GlobalStat

where OsApplicationName is entry in WdgMGlobalMemoryAppTaskRef and CoreAssignment is the entry in WdgMModeCoreAssignment.

If WdgMGlobalMemoryAppTaskRef is not configured, the following pattern has to be applied:

Rte_Call_WdgM_globalStateChangeCbK_Core<CoreAssignment>_GlobalStatusCallback where CoreAssignment is the entry in WdgMModeCoreAssignment.

Only if this naming convention is followed, the API of the RTE can be used within the callback.

Callback function for notifying the supervised entity's status change.

Note:

If WdgMUseRte is configured as true and thus the receiver of the callback is an application above the RTE, the the user has to configure this parameter with the name of a C function which shall be called by WdgM. Within a separate file where this function is defined the user has to include the corresponding Rte application header file and call the Rte function. The Rte function is named according the following convention:

Rte_Call_WdgM_<OsApplicationName>_localStateChangeCbK_<SupervisedEntityName>_LocalStatu

where OsApplicationName is entry in WdgMGlobalMemoryAppTaskRef of the relating WdgMMode and SupervisedEntityName is the entry in WdgMSupervisedEntity.

If WdgMAppTaskRef is not configured, the following pattern has to be applied:

Rte_Call_WdgM_localStateChangeCbK_<SupervisedEntityName>_LocalStatusCallback where SupervisedEntityName is the entry in WdgMSupervisedEntity.

Only if this naming convention is followed, the API of the RTE can be used within the callback.

"

Resolution:

ESCAN00097148 WdgMGlobalStateChangeCbK / WdgMLocalStateChangeCbK function prototype generated with incompatible signature compared to RTE

The described issue is corrected by modification of all affected work-products.

For this ESCAN a STORYC was created. The implementation takes more time than a BugFix.

ESCAN00097151 Incomplete Mirror Data

Component@Subcomponent: DrvCan_Mpc5700McanLI@Implementation

First affected version: 2.09.00

Fixed in versions: 2.20.00

Problem Description:

What happens (symptoms):

Not all data bytes are provided to the gateway.

When does this happen:

At runtime

In which configuration does this happen:

Only for AutoSar
AND using the Mirror Mode
AND using CAN-FD with more than 12 data bytes.

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097168 EcuM debug data cannot be found in the map file	
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	1.01.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

During A2L update several symbols of EcuM (that the EcuM generator actually registers through the CFG5 McData Service Interface) cannot be found in the map file.	
EcuM_ExpiredWakeup	
EcuM_PendingCheckWakeup	
EcuM_PendingWakeup	
When does this happen:	

After compilation when the A2L / calibration workflow is used to generate a complete A2L file with addresses of the target.	
In which configuration does this happen:	

Whenever generation of Debug Data is enabled in DaVinci Configurator and EcuM is used.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097174 WdgM_UpdateTickCount is declared not as "void" in SWC file**Component@Subcomponent:** SysService_Asr4WdM@GenTool_GeneratorMsr**First affected version:** 2.00.00**Fixed in versions:** 2.01.01**Problem Description:**

What happens (symptoms):

The Rte generates the prototype of function WdgM_UpdateTickCount as following:
FUNC(Std_ReturnType, WdgM_<ApplicationName>_CODE) WdgM_UpdateTickCount(void);

In fact this function is a void-function has no return value. Therefore this can result in a compiler error / warning.

When does this happen:

This issue happens if the the Rte is used and the timebase is configured as external tick source.

In which configuration does this happen:

/MICROSAR/WdgM/WdgMGeneral/WdgMUseRte == true
AND
/MICROSAR/WdgM/WdgMGeneral/WdgMTimebaseSource == WDG_M_EXTERNAL_TICK

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097203 Compiler error: "conversion of data types, possible loss of data" in case of large buffers

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 1.03.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Following compiler errors might be reported:

Error: Dcm.c, line 36235: error C4244: '=' : conversion from 'Dcm_CfgNetBufferSizeMemType' to 'Dcm_DidMgrDidLengthType', possible loss of data

Error: Dcm.c, line 13586: error C4244: '=' : conversion from 'const Dcm_CfgDidMgrOptimizedDidLengthType' to 'uint16', possible loss of data

Error: Dcm.c, line 25946: error C4244: '=' : conversion from 'const Dcm_CfgDidMgrOptimizedDidLengthType' to 'Dcm_DidMgrDidLengthType', possible loss of data

In many cases (dependent on compiler option settings) this might be reported as warning.

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

- At least one DCM buffer is larger than 65535bytes (ECUC parameter DcmDslBufferSize > 65535)
AND

- Service 0x22 (ReadDataByIdentifier) is handled within DCM (in Dcm_Cfg.h #define DCM_SVC_22_SUPPORT_ENABLED == STD_ON)

OR

- Service 0x2A (ReadDataByPeriodicIdentifier) is handled within DCM (in Dcm_Cfg.h #define DCM_SVC_2A_SUPPORT_ENABLED == STD_ON)

OR

- Service 0x2C (DynamicallyDefineDataByIdentifier) is handled within DCM (in Dcm_Cfg.h #define DCM_SVC_2C_SUPPORT_ENABLED == STD_ON)

OR

- Service 0x2F (InputOutputControlByIdentifier) is handled within DCM (in Dcm_Cfg.h #define DCM_SVC_2F_SUPPORT_ENABLED == STD_ON)

Note: Such buffer sizes are typically used in case of Bootloader applications.

Resolution Description:

Workaround:

Ignore the warning since no DID can have size of more than 16KB, since largest DcmDspDidDataPos can accept only up to 8KB and the largest DID signal can have only up to 8KB. So the largest DID can be a DID with a signal starting at position 8191 and having 8192 bytes.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097240 CanIf debug data cannot be found in the map file	
Component@Subcomponent:	If_AsrIfCan@GenTool_GeneratorMsr
First affected version:	3.07.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- During A2L update some symbols of CanIf (that the CanIf generator actually registers through the CFG5 McData Service Interface) cannot be found in the map file. CanIf_CtrlStates.CtrlModeOfCtrlStates CanIf_CtrlStates.PduModeOfCtrlStates When does this happen: ----- After compilation when the A2L / calibration workflow is used to generate a complete A2L file with addresses of the target. In which configuration does this happen: ----- Whenever generation of Debug Data is enabled in DaVinci Configurator and CanIf is used.	
Resolution Description: Workaround: ----- Fix the generated symbols in the A2L file manually before proceeding with the A2L workflow. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097257 Compiler error: error C2065: 'CanNm_CancelTransmit' : undeclared identifier

Component@Subcomponent: Nm_Asr4NmCan@Implementation

First affected version: 1.00.00

Fixed in versions: 7.00.00

Problem Description:

What happens (symptoms):

The following error is issued by the compile:

error C2065: 'CanNm_CancelTransmit' : undeclared identifier

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

- - /MICROSAR/PduR/PduRBswModules/PduRBswModuleRef is "CanNm"
AND
- /MICROSAR/PduR/PduRBswModules/PduRCancelTransmit is ON

Resolution Description:

Workaround:

- - Provide a User Config file with following content:

```
extern Std_ReturnType CanNm_CancelTransmit(PduIdType TxPduId);
```

and add it's path to the following parameter in DaVinci Configurator 5

/MICROSAR/CanNm/CanNmGlobalConfig/CanNmUserConfigFile

- Generate CanNm component in DaVinci Configurator 5.

Provide the following code in an application file:

```
#include "CanNm.h"
```

```
Std_ReturnType CanNm_CancelTransmit(PduIdType TxPduId)
{
    return E_OK;
}
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097274 Compiler error: Incompatible Rte_MemCpy prototypes	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.13.00
Fixed in versions:	1.17.00
Problem Description: What happens (symptoms): ----- Compilation fails because the prototype for Rte_MemCpy or Rte_MemCpy32 is incompatible to the prototype in the RTE implementation. When does this happen: ----- The error is issued by the compiler during compilation of the code in case the configuration is as described below. In which configuration does this happen: ----- This happens when the platform does not define uint16_least and uint32_least to the same types and when the configuration contains a component with sender-receiver communication or inter-runnable variables with arrays.	
Resolution Description: Workaround: ----- Typedef uint16_least and uint32_least to the same type. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097291 Compiler error: Use of undeclared identifier Rte_Appl_AckFlags	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.02.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

Compilation fails because a Rte_Feedback API accesses Rte_Appl_AckFlags that do not exist.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when a signal is sent from a different partition than the partition that contains the COM and when transmission acknowledgement is configured.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097298 Mismatching data type Dem_DTCOriginType**Component@Subcomponent:** Diag_Asr4Dcm@GenTool_GeneratorMsr**First affected version:** 7.02.00**Fixed in versions:** 9.01.00**Problem Description:**

What happens (symptoms):

RTE validation issues the following message:

RTE12025 - Multiple enumeration constants defined with the same name.

RTE12025 - Enumeration constant <DEM_DTC_ORIGIN_SECONDARY_MEMORY> of data type <Dem_DTCOriginType> already defined for data type <Dem_DTCOriginType>. Both are used within ComponentType <Name> but have difference values <5> and <4>.

Help:

- ensure uniqueness for all values of enumeration constants in the context of one ComponentType.

When does this happen:

At RTE generation time.

In which configuration does this happen:

- DEM Api Version according to ASR 4.3.0 is used (in DCM ECUC: "/Dcm/DcmConfigSet/DcmGeneral/DcmDemApiVersion" == DCM_DEM_API_4_03_00);

AND

- Sub-Service 0x1904 is supported (in Dcm_Cfg.h: #define DCM_SVC_19_04_SUPPORT_ENABLED == STD_ON);

OR

- Sub-Service 0x1906 is supported (in Dcm_Cfg.h: #define DCM_SVC_19_06_SUPPORT_ENABLED == STD_ON);

OR

- Sub-Service 0x1910 is supported (in Dcm_Cfg.h: #define DCM_SVC_19_10_SUPPORT_ENABLED == STD_ON);

OR

- Sub-Service 0x1914 is supported (in Dcm_Cfg.h: #define DCM_SVC_19_14_SUPPORT_ENABLED == STD_ON);

OR

- Sub-Service 0x1918 is supported (in Dcm_Cfg.h: #define DCM_SVC_19_18_SUPPORT_ENABLED == STD_ON);

OR

- Sub-Service 0x1919 is supported (in Dcm_Cfg.h: #define DCM_SVC_19_19_SUPPORT_ENABLED == STD_ON);

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097303 Compiler error: Call to job finished runnable misses parameters	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.08.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

Compilation fails because a task calls a notify job finished runnable without arguments.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when an NvBlock SWC is configured and when the NvM_MainFunction and the job finished runnable are mapped to different tasks.	
Resolution Description:	
Workaround:	

Do not map the job finished runnable to a task.	
It will then be executed in the context of the task that schedules NvM_MainFunction.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097341 Mode Transition Value of Mode Declaration Group WdgM_Mode is set to 255	
Component@Subcomponent:	SysService_Asr4WdM@GenTool_GeneratorMsr
First affected version:	2.01.00
Fixed in versions:	2.01.01
Problem Description:	
What happens (symptoms): ----- The following error is raised by CFG5: RTE13068 The Mode Transition Value of Mode Declaration Group <WdgM_Mode> is set to <255>. This value can not be represented by <WdgMMode> data type. Help: - Use a value in range of <WdgMMode> data type. - Map Mode Declaration Group to a sufficient data type. When does this happen: ----- This issue occurs during generation phase of the RTE generator only if Status Reporting Mechanism is set to WDGM_USE_MODE_SWITCH_PORTS. In which configuration does this happen: ----- /MICROSAR/WdgM/WdgMGeneral/WdgMStatusReportingMechanism == WDGM_USE_MODE_SWITCH_PORTS	
Resolution Description:	
Workaround: ----- Do not use Status Reporting Mechanism WDGM_USE_MODE_SWITCH_PORTS. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097355	Auto-Configuration Ecu State Handling: Self run request timeout value is not shown correct in case of 0
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	11.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	
----- The overview page of the Auto-configuration Ecu State handling does not show the correct value for the self run request timeout. Instead it shows the default value (0.1).	
When does this happen:	
----- Always if the value is set to 0.	
In which configuration does this happen:	
----- In all configurations with Auto-Configuration Ecu State Handling configured	
AND	
Value of self run request timeout is set to 0.	
Resolution Description:	
Workaround:	
----- No workaround available.	
Resolution:	
----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097457 Matrix dimensions are swapped for two dimensional arrays in A2L file	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

A measurement tool displays rows as columns and columns as rows.	
When does this happen:	

During measurement and calibration.	
In which configuration does this happen:	

This happens when the configuration contains arrays with two dimensions.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097476 RTE01004 error during contract phase generation (Could not read back DVCfgRteGen data)	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.12.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms): ----- RTE generation is incorrectly aborted with RTE01004 'Could not read back DVCfgRteGen data' error message.	
When does this happen: ----- During contract phase header generation.	
In which configuration does this happen: ----- This happens for all configuration when 'Contract phase header' is selected as 'Generation Mode' inside Cfg5 project settings.	
Resolution Description:	
Workaround: ----- Change Cfg5 project settings so that 'Template file and contract phase header' is selected as 'Generation Mode'.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097477		Code generation is not possible due to error RTE13068 - Insufficient data type to represent mode value
Component@Subcomponent:	Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr	
First affected version:	7.00.00	
Fixed in versions:	9.00.01	
Problem Description:		
What happens (symptoms):		

DaVinci Configurator reports the following error		
[Error] RTE13068 - Insufficient data type to represent mode value		
- The Mode Transition Value of Mode Declaration Group <ComMMode> is set to <3>.		
This value can not be represented by <ComM_ModeType> data type.		
When does this happen:		

The error is issued by the RTE generator during calculation phase.		
In which configuration does this happen:		

This issue is reported as a preventive measure. We assume that the issue does not occur.		
It was detected in a DaVinci Configurator Service Pack, which is not distributed anymore.		
Resolution Description:		
Workaround:		

No workaround available.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00097525 RTE49999 when transformers are not configured for all fan-out signals	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.04.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with an RTE49999 error.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when a data element is mapped to multiple signals and when not all the signals are configured to use transformers.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097531		a2l: In Variant use case only the first variant is generated for the bus specific a2l files	
Component@Subcomponent:		Cp_Asr4Xcp@GenTool_GeneratorMsr	
First affected version:		1.00.00	
Fixed in versions:		3.00.01	
Problem Description:			
What happens (symptoms):			

The XCP Generator creates bus specific a2l fragments (e.g. CanXcp_....a2l). Those fragments shall be present for each channel and each variant. The current generator only generates the channels of the first variant. A2l files for additional variants are not generated.			
As a result the a2l fragments for additional variants must be created manually.			
When does this happen:			

Always and immediately			
In which configuration does this happen:			

When multiple variants are used.			
Resolution Description:			
Workaround:			

A2l files for additional variants can be created manually.			
Resolution:			

The described issue is corrected by modification of all affected work-products.			

ESCAN00097649 Compiler error: Rte_Write writes a variable that does not exist	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

The compiler states: undeclared identifier	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

<ul style="list-style-type: none"> - LdCom is used with external Signal - Write from non bsw partition - Data transformation is disabled 	
Resolution Description:	
Workaround:	

Create another receiver port in the same component as the LdCom sender port and connect it the LdCom sender port.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097683 A generated value is not in range of the specified datatype	
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

An error is reported in the configurator with following error message: COM90500 The value 122040 with comment () is not in the range of the specified datatype UINT_16.	
When does this happen:	

During generation of COM	
In which configuration does this happen:	

In configurations in which any generated table has more than 65535 entries AND /PduR/PduRBswModules/PduRTransportProtocol for COM is set to FALSE	
Resolution Description:	
Workaround:	

1) Use LdCom instead of COM for large PDUs	
or	
2) Enable /ActiveEcuC/PduR/Com[1:PduRTransportProtocol] and configure one Dummy TP PDU	
or	
3) set /MICROSAR/Com/ComConfig/ComIPdu/ComIPduSignalProcessing to IMMEDIATE for all PDUs which are greater than 65535.	
Resolution:	

No modification of code as it would cause performance problems and use a lot of RAM, especially in Post-Build configurations.	

ESCAN00097684 Warning RTE49999 when XcpEvent support is enabled	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms): ----- RTE generator prints a warning RTE49999 AST property is undefined. Compilation fails because a task calls the XcpEvent API with the undfined identifier UNDEFINED.	
When does this happen: ----- During generation.	
In which configuration does this happen: ----- This happens when XcpEvent support is enabled and when a task contain only server runnables as executable entities.	
Resolution Description:	
Workaround: ----- Map the server runnables to another task that also contains runnables with different trigger.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097802 Activation reason data type uses bit access**Component@Subcomponent:** Rte_Core@Implementation**First affected version:** 1.09.00**Fixed in versions:** 1.18.00**Problem Description:**

What happens (symptoms):

In the struct Rte_ActivatingEvent_<ComponentTypeName>_Type in Rte.c, an activating event for a runnable specifies a bit field length. This leads to a Misra Warning "The behavior of bit fields not of type int is undefined".

When does this happen:

Always and immediately

In which configuration does this happen:

It happens when all of the following conditions are fulfilled:

- a runnable has an activation reason AND
- the runnable is not triggered by "On Operation Called", "On Mode Entry", "On Mode Exit" or "On Transition" AND
- RTE's optimization mode equals MEMORY

Resolution Description:

Workaround:

Set Rte's optimization mode to RUNTIME.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00097876 Generated data streams toggle with each code generation if <MSN>ReduceDataByStreaming is enabled**Component@Subcomponent:** SysService_Asr4BswMCfg5@GenTool_GeneratorMsr**First affected version:** 2.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

Generated Code has to be recompiled or added again to the Users CMS because the order in streamed CONST arrays is not deterministic and changes by chance with each code generation.

When does this happen:

At generation time.

In which configuration does this happen:

Any configuration where <MSN>ReduceDataByStreaming returns true.

Resolution Description:

ESCAN00097910 Dcm_Swc.arxml: Missing values of Mode-Declarations in Mode-Declaration-Groups	
Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	7.01.01, 9.04.00
Problem Description:	
What happens (symptoms):	

RTE validation issues the following message:	
RTE13077 - ModeDeclarationGroup <NameMDG> with explicit order does not specify a value for mode <NameMode>.	
When does this happen:	

At RTE generation time.	
In which configuration does this happen:	

- At least one "/Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule" exists;	
AND	
- This ModeRule has at least one DcmArgumentRef to a "/Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition";	
AND	
- This ModeCondition has configured DcmBswModeRef:	
- DcmContextModeDeclarationGroupPrototype is set to a DCM ModeDeclarationGroup prototype;	
AND	
- DcmTargetModeDeclaration is set to a valid ModeDeclaration;	
Resolution Description:	
Workaround:	

Use a DaVinci Cfg5 version smaller than 5.14.8.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097950 Compiler error: 'CanTp_GetRxSduCfgInd' undefined	
Component@Subcomponent:	Tp_Asr4TpCan@Implementation
First affected version:	3.01.00
Fixed in versions:	3.03.02
Problem Description:	
What happens (symptoms):	

The following errors are reported during compilation:	
'CanTp_GetRxSduCfgInd' undefined; assuming extern returning int	
'CanTp_GetRxSduCfgIndStartIdxOfRxPduMap' undefined; assuming extern returning int	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

The errors are reported in configurations fulfilling the following conditions:	
- Configurations where no Rx N-SDU is present (not a single CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu container is present)	
AND	
- Normal addressing is used (at least one CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxAddressingFormat is set to CANTP_STANDARD) OR multiple addressing is used (at least two CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxAddressingFormat parameters are set to different values)	
Resolution Description:	
Workaround:	

In order to compile, the missing functions could be defined as follows within a user config file (CanTp/CanTpGeneral/CanTpUserConfigFile):	
#define CanTp_GetRxSduCfgInd(Index) 0	
#define CanTp_GetRxSduCfgIndStartIdxOfRxPduMap(Index) 0	
NOTE: The macros proposed in the workaround could trigger some warnings during compilation (e.g., potential divide by 0 in the function CanTp_RxTransmitFrame). If no Rx SDUs are configured, the lines where the warnings are issued won't ever be reached and ignoring those warnings would be safe.	
Resolution:	

Not yet available.	

ESCAN00097971 RTE49999: Mismatching constant values	
Component@Subcomponent:	Rte_Asr4@Generator
First affected version:	4.07.00
Fixed in versions:	4.18.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with an error message RTE49999 Mismatching constant values: <cfull> and <celement> although the elements use the same value.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains sender-receiver communication where a sender only receives a subset of the record elements of the sender. The problem occurs when the receiving component does not have a port access to the receiver port.	
Resolution Description:	
Workaround:	

Configure port accesses for the receivers.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098057 Generated data streams toggle with each code generation if <MSN>ReduceDataByStreaming is enabled	
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	15.00.00
Problem Description:	
What happens (symptoms): ----- Generated Code has to be recompiled or added again to the Users CMS because the order in streamed CONST arrays is not deterministic and changes by chance with each code generation.	
When does this happen: ----- At generation time.	
In which configuration does this happen: ----- Any configuration where /ComReduceDataByStreaming returns true.	
Resolution Description:	
Workaround: ----- Configure ComReduceDataByStreaming to false if available.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098062 RTE49999: When InitializeImplicitBuffers is configured for implicit connections to NvBlock SWCs	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

RTE Generator aborts with a RTE49999 error.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when all of the following conditions are true:	
- /MICROSAR/Rte/RteGeneration/RteInitializeImplicitBuffers is configured to true	
- the access to the data elements is implicit	
- the configuration contains a sender-receiver connection to a NvBlock SWC and the NvBlock has no RAM block init value	
or	
the nv port is not connected	
Resolution Description:	
Workaround:	

Connect all Nv ports with implicit accesses.	
Configure RAM block init values for the NvBlocks.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098068 Null pointer exception when a service dependency contains an invalid pim reference	
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	4.18.00
Problem Description:	
What happens (symptoms):	

RTE generator aborts with a null pointer exception.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains incomplete service dependencies e.g.	
<ROLE-BASED-DATA-ASSIGNMENT>	
<ROLE>ramBlock</ROLE>	
<USED-DATA-ELEMENT/>	
</ROLE-BASED-DATA-ASSIGNMENT>	
instead of	
<ROLE-BASED-DATA-ASSIGNMENT>	
<ROLE>ramBlock</ROLE>	
<USED-DATA-ELEMENT>	
<LOCAL-VARIABLE-REF DEST="VARIABLE-DATA-PROTOTYPE">/path/to/pim</LOCAL-VARIABLE-REF>	
</USED-DATA-ELEMENT>	
</ROLE-BASED-DATA-ASSIGNMENT>	
Resolution Description:	
Workaround:	

Configure a valid reference.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098104 RTE Analyzer reports false out of bound accesses	
Component@Subcomponent:	GenTool_IRAnalyzer@Application
First affected version:	0.09.00
Fixed in versions:	1.01.00
Problem Description:	
What happens (symptoms):	

RTE generator reports an out of bound access although all write accesses are within the bounds.	
When does this happen:	

This happens during analysis of write accesses with multiple possible pointer targets. This means write accesses to array elements or write accesses to e.g. function parameters that get passed different values in different call sites. This only happens when the write access does not write the entire object. E.g. when an element in a structure is written. This only happens when the write access is an assignment, not a memcpy or memset.	
In which configuration does this happen:	

This happens in all configurations. See description above for the code structure of the analyzed code that leads to the ESCAN.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098155 Inconsistencies of Technical Reference regarding Dem usage	
Component@Subcomponent:	SysService_Asr4WdM@Doc_TechRef
First affected version:	1.02.00
Fixed in versions:	1.02.02
Problem Description:	
What happens (symptoms):	

The technical reference was updated. In an earlier version of the WdgM the Dem was not directly connected to the WdgM. A wrapper was needed to handle the Dem. Now the documentation was updated, because there was a description how to implement this wrapper on several passage.	
This description confuses the customer and is fixed.	
When does this happen:	

Missing updated documentation after further development of the component.	
In which configuration does this happen:	

Always	
Resolution Description:	
Workaround:	

No workaround available.	
This is only an inconsistency in user documentation and therefore a workaround is not applicable. A "textual" workaround is that a wrapper is no more needed for Dem usage.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098167 RTE01081 Model object </MICROSAR/ IoHwAb_swc/ComponentTypes/IoHwAb> of command line parameter -m is invalid.	
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	4.18.00
Problem Description:	
What happens (symptoms): ----- RTE SWC Template generation in CFG5 aborts with an error message RTE01081 Model object </MICROSAR/IoHwAb_swc/ComponentTypes/IoHwAb> of command line parameter -m is invalid.	
When does this happen: ----- During SWC template generation.	
In which configuration does this happen: ----- This happens in configuration that contain the IoHwAb component.	
Resolution Description:	
Workaround: ----- No workaround available.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098187 RTE generator generates wrong compu method in case data type names are not unique	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.18.00
Problem Description: What happens (symptoms): ----- RTE generator generates MEASUREMENT, CHARACTERISTIC or axis objects that reference invalid COMPU_METHOD objects. When does this happen: ----- During generation. In which configuration does this happen: ----- This happens when the configuration contains multiple data types with the same name that reference different compu methods. This is the case when the configuration contains an implementation data type and one or multiple application data types with the same name and when the data types use different compu methods or no compu method at all.	
Resolution Description: Workaround: ----- Use distinct data type names. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098204	RTE01060: Could not read Rte_Needs.ecuc.arxml when exclusive areas with implementation method resource are accessed from multiple partitions
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with an error message RTE01060: Could not read Rte_Needs.ecuc.arxml when exclusive areas with implementation method resource are accessed from multiple partitions	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains exclusive areas with implementation method OS_RESOURCE and when the exclusive area is accessed from multiple partitions, e.g. when it is accessed from unmapped server runnables that are called from multiple partitions.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098260 Erroneous validation message "CanIfMultipleBasicCANTxObjects is not required"	
Component@Subcomponent:	If_AsrIfCan@GenTool_GeneratorMsr
First affected version:	4.02.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- Erroneous validation message CANIF10027 (None CAN-channel has multiple BasicCAN Tx-objects. Hence the feature ""CanIfMultipleBasicCANTxObjects" is not required in current configuration and must be disabled.) shows up in CFG5 and cannot be solved. When does this happen: ----- During configuration. In which configuration does this happen: ----- Multiple CAN drivers are used AND There is at least one CAN channel with != 1 BasicCAN Tx-hardware object ("CanHardwareObject" with "CanHandleType" == BASIC and "CanObjectType" == TRANSMIT) for one of the drivers.	
Resolution Description: Workaround: ----- Set the parameter "CanIfMultipleBasicCANTxObjects" to user defined and keep it enabled. [Error] CANIF10027 is then demoted to a warning that can be ignored. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098424	a2I: OPTIONAL_CMD GET_DAQ_EVENT_INFO generated unconditionally.
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	3.00.02
Problem Description:	
What happens (symptoms):	

The statement "OPTIONAL_CMD GET_DAQ_EVENT_INFO" is generated unconditionally in the file Xcp.a2I.	
The Configuration Option XcpGetDAQEventInfo has no influence.	
As a result the Master Tool tries to use this command event though the Option is inactive and the command is therefore not present. This will result in a negative response.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

All configurations.	
Resolution Description:	
Workaround:	

The generated Xcp.a2I can be modified manually to remove the generated OPTIONAL_CMD.	
For CANape this is not required as CANape detects the missing command and removes this line automatically from the a2I after the first failed try.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098469 Unused Interrupt enabled	
Component@Subcomponent:	DrvCan_Mpc5700McanLI@Implementation
First affected version:	2.09.00
Fixed in versions:	3.01.00
Problem Description:	
What happens (symptoms):	

Transmission Cancellation Finished Interrupt Enable (TCFE) is active but not used. The interrupt is not handled by the driver, therefore, there are no unexpected consequences and the ECU will continue to function correctly.	
When does this happen:	

At runtime, during initialization.	
In which configuration does this happen:	

TX Interrupt enabled, "CAN_TX_PROCESSING == CAN_INTERRUPT" OR Individual polling is configured, "CAN_INDIVIDUAL_PROCESSING == STD_ON"	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098487 PDUR_EXCLUSIVE_AREA_1 is created by tool but not used in embedded code	
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	12.00.00
Problem Description:	
What happens (symptoms):	

PDUR_EXCLUSIVE_AREA_1 is created by the tool, but never used in the embedded code of the PduR.	
When does this happen:	

Always.	
In which configuration does this happen:	

All configurations.	
Resolution Description:	
Workaround:	

Ignore PDUR_EXCLUSIVE_AREA_1.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098523 GeneralDiagnosticInfo interface is named "GeneralEventInfo" instead of "GeneralEvtInfo"	
Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	13.00.00
Problem Description: What happens (symptoms): ----- The port name of the GeneralDiagnosticInfo interface is named "GeneralEventInfo" instead of "GeneralEvtInfo". This is visible when connecting this DEM provided port with an application port in the RTE, but has no implication on the functional operation of the modules. When does this happen: ----- Always and immediately In which configuration does this happen: ----- Parameter /Dem/DemGeneral/DemGeneralDiagnosticInfoSupport==TRUE enables the GeneralDiagnosticInfo Interface	
Resolution Description: Workaround: ----- Connect the port named "GeneralEventInfo" instead of "GeneralEvtInfo" with the related application. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098583	Generator Error Message ""XCP90110 Undefined DefinitionRef for Parameter." - misleading problem indication
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	-----
Generator Error Message:	
"XCP90110 Undefined DefinitionRef for Parameter. Element def.: /MICROSAR/SoAd/SoAdConfig/	SoAdSocketConnectionGroup/SoAdSocketId Parent: SoAdSocketConnectionGroup_XY"
indicates a problem with the Bswmd Files whereas the solution is to use Socket Connections	instead of Socket Connection Groups to configure the Xcp Ethernet Pdus in SoAd.
When does this happen:	-----
Xcp On Ethernet Tx Pdu configured in SoAd with a reference to a Socket Connection Group	
In which configuration does this happen:	-----
Xcp On Ethernet Tx Pdu configured in SoAd with a reference to a Socket Connection Group instead	of a reference to a socket connection.
Resolution Description:	
Workaround:	-----
Select connection instead of connection group	
Resolution:	-----
The described issue is corrected by modification of all affected work-products.	

ESCAN00098584	NvM NVM01036 validation does not clearly describe the problem
Component@Subcomponent:	MemService_AsrNvM@GenTool_GeneratorMsr
First affected version:	4.02.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	
----- DaVinci Cfg5 shows the NvM error NVM01036 NVM01036 "NvMCalcRamBlockCrc requires configured NvMBlockCrcType, NvMSelectBlockForReadAll, NvMRamBlockDataAddress and disabled" for NvMBlockDescriptors derived from NvBlockNeeds. Resolving the problem via provided solving action leads to other validation errors in e.g. RTE.	
Since the NvMBlockDescriptor is derived from NvBlockNeeds, the error cannot be fixed within the DaVinci Cfg5.	
When does this happen:	
----- NvMBlockDescriptor derived from NvBlockNeeds. For other blocks the error shall be clear and resolvable within the Cfg5.	
In which configuration does this happen:	
----- NvBlockNeeds with calcRamBlockCrc true, reliability != NO and enabled explicit synchronization leads to NvMBlockDescriptor with NvMBlockUseCrc true, NvMBlockCrcType != NoCrc and NvMBlockUseSyncMechanism.	
Resolution Description:	
Workaround:	
----- Correct the NvMBlockDescriptor preconditions directly within the DaVinci Developer -> ensure the configuration matches the preconditions described in error message NVM01036. Do not use the provided solving action!	
Resolution:	
----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098599 RTE49999 when a data element without init value is connected to a data element without port accesses	
Component@Subcomponent:	Rte_Asr4@Generator
First affected version:	4.06.00
Fixed in versions:	4.18.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with an RTE49999 error message.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the following conditions evaluate to true:	
<ul style="list-style-type: none"> - a r-port data element of a nonqueued sender-receiver communication does not specify an initial value - the connected p-port data element specifies a textual initial value - the r-port data element uses a data type without compu method or with a compu method that does not contain an entry for the textual initial value - the sender runnable does not have any port accesses to the data element 	
Resolution Description:	
Workaround:	

Add port accesses.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098646 RTE generator creates NvMRomBlockDataAddress with ROM variables that do not exist**Component@Subcomponent:** Rte_Core@Implementation**First affected version:** 1.00.00**Fixed in versions:** 1.18.00**Problem Description:**

What happens (symptoms):

Compilation of NVM fails because its data structures reference a constant that does not exist. The according parameter /MICROSAR/NvM/NvMBlockDescriptor/NvMRomBlockDataAddress with the wrong value is created by the RTE generator.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the online calibration method is DOUBLE_POINTERED and when a PerInstanceMemory is mapped to a NV Block and uses a calibration parameter as ROM block default.

Resolution Description:

Workaround:

Create the missing constant in the application or set the parameter /MICROSAR/NvM/NvMBlockDescriptor/NvMRomBlockDataAddress to user defined and set it to the Rte_Calprm constant in Rte.c that is used to initialize RteParameterRefTab.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098679 Compiler error: incompatible declaration of ComM_ConfigPtr

Component@Subcomponent: Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr

First affected version: 3.00.00

Fixed in versions: 9.00.03

Problem Description:

What happens (symptoms):

Compiler reports an error in the following line

P2CONST(ComM_ConfigType, AUTOMATIC, COMM_INIT_DATA) ComM_ConfigPtr = NULL_PTR;

\external\BSW\ComM.c:

declaration is incompatible with "ComM_ConfigType const *ComM_ConfigPtr
@ .OS_OsApplication_NonTrusted_VAR_NOINIT" declared at line 672 of gendata/
ComM_Private_Cfg.h

Note: the reason is that the variable ComM_ConfigPtr is mapped to two different memory sections
COMM_START_SEC_VAR_ZERO_INIT_UNSPECIFIED (as defined in ComM.c) and
COMM_START_SEC_VAR_NOINIT_UNSPECIFIED (as declared in ComM_Private_Cfg.h).

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as
described below.

In which configuration does this happen:

- Implementation Variant of ComM module is VARIANT-POST-BUILD-LOADABLE or Postbuild-
Selectable Support is enabled

AND

- ComM_ConfigPtr symbol is used: # define COMM_USE_INIT_POINTER STD_ON can be found in
ComM_Cfg.h or ComM_PBCfg.h

AND

- Memory sections COMM_START_SEC_VAR_ZERO_INIT_UNSPECIFIED and
COMM_START_SEC_VAR_NOINIT_UNSPECIFIED are mapped to different memory areas.

Resolution Description:

Workaround:

Map the memory section COMM_START_SEC_VAR_ZERO_INIT_UNSPECIFIED to the same memory
area as COMM_START_SEC_VAR_NOINIT_UNSPECIFIED.

This has no side-effects because ComM_ConfigPtr is always assigned in ComM_Init().

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098712	Linker error: DemTriggerEventDataChangedCallback==FALSE and DemTriggerEventStatusChangedCallback==FALSE will only suppress the creation/usage of SWC port interfaces, but not the underlying function calls
Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	13.00.00
Problem Description:	

ESCAN00098712

**Linker error:
DemTriggerEventDataChangedCallback==FALSE
and
DemTriggerEventStatusChangedCallback==FALSE
will only suppress the creation/usage of SWC port
interfaces, but not the underlying function calls**

What happens (symptoms):

a) With (the Vector specific) configuration parameter DemTriggerEventDataChangedCallback==FALSE, the existence of /Dem/DemConfigSet/DemEventParameter/DemCallbackEventDataChanged configuration containers shall be ignored, and the resulting callouts shall not be generated. This option is typically used during development, when the related code in the application of the ECU is not available yet.

When using DemTriggerEventDataChangedCallback==FALSE, the DEM (correctly) does not call any such callout function.

Nevertheless the callout functions names are generated and therefore a linker error occurs, when you don't implement a dummy function stub for them.

The name of the callout function is defined in following way:

- If an additional parameter /Dem/DemConfigSet/DemEventParameter/DemCallbackEventDataChanged/DemCallbackEventDataChangedFnc was configured, its value is the function name.
- Otherwise when using the RTE, the generated function name is Rte_Call_CBDataEvt_{shortname of DemEventParameter}_EventDataChanged,
- or when using no RTE, the generated function name is Appl_Dem_CBDataEvt_{shortname of DemEventParameter}_EventDataChanged.

b) A similar issue exists with the EventStatusChanged configuration:

With (the Vector specific) configuration parameter DemTriggerEventStatusChangedCallback==FALSE, the existence of /Dem/DemConfigSet/DemEventParameter/DemCallbackEventStatusChanged configuration containers shall be ignored, and the resulting callouts shall not be generated. This option is typically used during development, when the related code in the application of the ECU is not available yet.

When using DemTriggerEventStatusChangedCallback==FALSE, the DEM (correctly) does not call any such callout function.

Nevertheless the callout functions names are generated and therefore a linker error occurs, when you don't implement a dummy function stub for them.

The name of the callout function is defined in following way:

- If an additional parameter /Dem/DemConfigSet/DemEventParameter/DemCallbackEventStatusChanged/DemCallbackEventStatusChangedFnc was configured, its value is the function name.
- Otherwise when using the RTE, the generated function name is Rte_Call_CBEvtUdsStatusChanged_{shortname of DemEventParameter}_{shortname of DemCallbackEventStatusChanged}_CallbackEventUdsStatusChanged,
- or when using no RTE, the generated function name is Appl_Dem_CBEvtUdsStatusChanged_{shortname of DemEventParameter}_{shortname of DemCallbackEventStatusChanged}_CallbackEventUdsStatusChanged.

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

ESCAN00098712

**Linker error:
DemTriggerEventDataChangedCallback==FALSE
and
DemTriggerEventStatusChangedCallback==FALSE
will only suppress the creation/usage of SWC port
interfaces, but not the underlying function calls**

In which configuration does this happen:

Case a)

/Dem/DemGeneral/DemTriggerEventDataChangedCallback == FALSE

The name(s) of the function(s) are generated, when a /Dem/DemConfigSet/DemEventParameter/DemCallbackEventDataChanged container exists and are derived from the parent DemEventParameter shortname.

This generated name is replaced by a user-defined function name, when the container has a sub-parameter ~DemCallbackEventDataChanged/DemCallbackEventDataChangedFnc.

Case b)

/Dem/DemGeneral/DemTriggerEventStatusChangedCallback == FALSE

The name(s) of the function(s) are generated, when a /Dem/DemConfigSet/DemEventParameter/DemCallbackEventStatusChanged container exists and are derived from the parent DemEventParameter and the DemCallbackEventStatusChanged shortname.

This generated name is replaced by a user-defined function name, when the container has a sub-parameter ~DemCallbackEventStatusChanged/DemCallbackEventStatusChangedFnc.

The RTE support is controlled by /Dem/DemGeneral/DemUseRte.

Resolution Description:

Workaround:

Either delete the (currently unimplemented) callbacks from your configuration (containers DemEventParameter/DemCallbackEventDataChanged resp. DemEventParameter/DemCallbackEventStatusChanged), and re-add them later again, when these functions are available.

Or delete the configuration parameters DemTriggerEventDataChangedCallback and DemTriggerEventStatusChangedCallback (or set them to TRUE)

AND

implement the missing functions (as stub routines).

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098820 RTE generator reports unexpected exit code	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.08.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

RTE generator reports an error message RTE01002 Unexpected Exit Code. MicrosarRteGen.exe crashes.	
When does this happen:	

During generation.	
In which configuration does this happen:	

In all configurations when the RTE generator is executed on a system with Windows10 with latest patches.	
The problem was reported to occur after applying KB4074592.	
Resolution Description:	
Workaround:	

No workaround available.	
Sometimes the problem disappears after the restart of the computer.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098822 RTE Generator reports unexpected exit code	
Component@Subcomponent:	Rte_Analyzer@Application
First affected version:	0.05.00
Fixed in versions:	1.01.00
Problem Description:	
What happens (symptoms): ----- RTE generator reports an error message RTE01002 Unexpected Exit Code. MicrosarRteAnalyzerCfgGen.exe or MicrosarRteAnalyzer.exe crashes. When does this happen: ----- During generation/analysis. In which configuration does this happen: ----- In all configurations when the RTE generator is executed on a system with Windows10 with latest patches. The problem was reported to occur after applying KB4074592.	
Resolution Description:	
Workaround: ----- No workaround available. Sometime the problem disappears when the system is restarted. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00098865 RTE49999 when sender and receiver use different init value constants with the same values	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with an RTE49999 error message.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains sender-receiver communication where sender and receiver use different init values and when the receiver init value is also used on another location.	
Resolution Description:	
Workaround:	

Use distinct init values for all elements.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098866 Service 0x3E: Misleading caution box regarding external service implementation	
Component@Subcomponent:	Diag_Asr4Dcm@Doc_TechRef
First affected version:	5.00.00
Fixed in versions:	10.00.00
Problem Description:	
What happens (symptoms):	

There is a misleading caution box in chapter "5.25.4 Configuration Aspects" stating "This service is mandatory and therefore may not be missing in the configuration and cannot be overridden by an application implementation. "	
But this service can be configured to be implemented by the application.	
When does this happen:	

Reading the technical reference.	
In which configuration does this happen:	

N/A	
Resolution Description:	
Workaround:	

Consider the service 0x3E to be implemented by the application as well.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098883 /MICROSAR/Xcp/XcpGeneral/XcpTimestampTicks limited in range to uint8	
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	3.00.02
Problem Description:	
What happens (symptoms):	

The parameter /MICROSAR/Xcp/XcpGeneral/XcpTimestampTicks has a range of uint16 in the bswmd/GUI but the generator treats this value as uint8.	
As a result, code generation will fail with an exception should this parameter have a value > 255.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

All configurations	
Resolution Description:	
Workaround:	

Use a user.cfg file to set the resulting define manually, e.g.:	
user.cfg:	
#undef XCP_DAQ_TIMESTAMP_TICKS_PER_UNIT	
#define XCP_DAQ_TIMESTAMP_TICKS_PER_UNIT 1000	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098887 Wrong linker section used	
Component@Subcomponent:	DrvCan_Mpc5700McanLI@Implementation
First affected version:	2.09.00
Fixed in versions:	3.02.00
Problem Description:	
What happens (symptoms):	

The variable "mirrorData" is placed in the wrong linker section, the default segment (bss) is used erroneously.	
When does this happen:	

At compile/link time.	
In which configuration does this happen:	

For all configurations using the Mirror Mode (CAN_MIRROR_MODE == STD_ON).	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098893		Compiler error: Missing Rte_MemClr when activation reason is used in a system with OS applications
Component@Subcomponent:	Rte_Core@Implementation	
First affected version:	1.15.00	
Fixed in versions:	1.18.00	
Problem Description:		
What happens (symptoms):		

Compilation fails because Rte_MemClr is missing.		
When does this happen:		

The error is issued by the compiler during compilation of the code in case the configuration is as described below.		
In which configuration does this happen:		

This happens when the configuration has configured activation reasons and when no minimum start interval, mode disablings, update flags, never received flags, acknowledge flags, alive timeout and dirty flags are configured.		
Resolution Description:		
Workaround:		

Configure never received handling for a receiver port.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00098918 Compiler error: Duplicated implicit APIs in analyzer stubs when the same port access is declared twice	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.14.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

Analysis with RTE Analyzer fails because TSC Rte_IWrite, Rte_IWriteRef or Rte_IRead stubs are duplicated.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when multiple implicit accesses for the same port and data element are configured for the same runnable.	
Resolution Description:	
Workaround:	

Delete duplicated implicit accesses.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098922 NmStateChangeCallback is not called if same states are passed in Nm_StateChangeNotification

Component@Subcomponent: Nm_Asr4NmCan@Implementation

First affected version: 3.00.00

Fixed in versions: 6.03.01, 7.02.01

Problem Description:

What happens (symptoms):

Nm calls Det_ReportError with the following parameters:

ModuleId: 0x27

InstanceId: 0x00

ApiId: 0x16

ErrorId: 0x23 (NM_E_SAME_STATE)

If '/MICROSAR/Nm/NmGlobalConfig/NmGlobalProperties/NmDevErrorDetect' is ON.

In any case '/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmStateChangeIndCallback' is not called.

When does this happen:

During runtime in case a channel is requested, released and requested again within
'/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmRepeatMessageTime'

In which configuration does this happen:

On channels with:

- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmPnEnabled' = ON
AND

- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/
CanNmPnHandleMultipleNetworkRequests' = ON
AND

- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmStateChangeIndEnabled' = ON
AND

- '/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmStateChangeIndEnabled' = ON
AND

- '/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmStateChangeIndCallback' is
implemented in application
AND

- '/MICROSAR/Nm/NmGlobalConfig/NmGlobalProperties/NmDevErrorDetect' = ON

Resolution Description:

ESCAN00098922 NmStateChangeCallback is not called if same states are passed in Nm_StateChangeNotification

Workaround:

Create a user configuration file with the following content:

```
#if defined CANNM_SOURCE
# undef Nm_StateChangeNotification
# define Nm_StateChangeNotification Appl_Nm_StateChangeNotification
#endif
```

and add this file in Davinci Configurator '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmUserConfigFile' parameter.

Create or use an application file and add the following content:

```
#include "Nm_Cbk.h"

#if( NM_STATE_CHANGE_IND_ENABLED == STD_ON )
FUNC( void, NM_CODE ) Appl_Nm_StateChangeNotification( CONST( NetworkHandleType,
AUTOMATIC ) nmChannelHandle,
CONST( Nm_StateType, AUTOMATIC) nmPreviousState,
CONST( Nm_StateType, AUTOMATIC ) nmCurrentState )
{
    if( nmPreviousState != nmCurrentState )
    {
        Nm_StateChangeNotification(nmChannelHandle, nmPreviousState, nmCurrentState);
    }
}
#endif
```

This workaround filters all calls of CanNm to Nm in case nmPreviousState and nmCurrentState have the same value.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00098939 RTE49999 when application data type with texttable compu method is mapped to float implementation data type	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with an error RTE49999.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when an application data type with texttable compu method is mapped to a float data type and when the float data type is on another data element that is mapped to a signal.	
Resolution Description:	
Workaround:	

Create a separate type reference implementation data type and map it to the application data type.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098955 RTE49999 when a compu method declared CompuScales that would result in the same symbol	
Component@Subcomponent:	Rte_Asr4@Generator
First affected version:	4.00.00
Fixed in versions:	4.18.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with an RTE49999 error "key already exists"	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when a texttable, bitfield texttable or scale linear and texttable compu method contains multiple compu scales that would result in the the same symbol in the code.	
Resolution Description:	
Workaround:	

Use different symbols for the compu scales.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098966 Missing reference about the cluster index for the Nm Gateway Coordination Extension	
Component@Subcomponent:	Nm_Asr4NmIf@Doc_TechRef
First affected version:	8.00.00
Fixed in versions:	11.00.02
Problem Description:	
What happens (symptoms):	

It is not clear what index it is used for the Nm Gateway Coordination Extension when setting the remote filters and the PartEcu active Channels for a node Id.	
An info box is needed in which it is clear that the "Nm Gateway Coordination Extension " uses cluster indexes.	
When does this happen:	

When reading the technical reference.	
In which configuration does this happen:	

All configurations with SDLC Gateway handling.	
Resolution Description:	
Workaround:	

Use the description provided in this issue ticket.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099018 NPE during generation with invalid ComRxDataTimeoutSubstitutionValue	
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	15.01.00
Problem Description:	
What happens (symptoms):	

An NPE is raised:	
COM90005 Exception in Com generator during Validation encountered:	
java.lang.NullPointerException	
/ActiveEcuC/Com	
When does this happen:	

During generation of a configuration	
In which configuration does this happen:	

an invalid value is set to ComRxDataTimeoutSubstitutionValue (e.g. a string "test" instead of an integer value)	
Resolution Description:	
Workaround:	

set a valid value to ComRxDataTimeoutSubstitutionValue	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099049 RTE49999: when task type is set to basic and cyclic trigger implementation is set to none	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.05.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with a RTE49999 error.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains a task with task type set to basic and when this task contains a runnable entity that also is a schedulable entity and when the cyclic trigger implementation is set to none.	
Resolution Description:	
Workaround:	

Set the cyclic trigger implementation to Auto and configure a software timer for the generated alarms.	
When the APIs of the software timer are not called, the OS will not trigger the task and it can still be triggered with a schedule table.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099057 EcuM Wakeup Source defines are generated multiple times with numerical postfix in case of variance**Component@Subcomponent:** SysService_Asr4EcuM@GenTool_GeneratorMsr**First affected version:** 8.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

Wakeup Source Defines are generated multiple times with an numerical postfix, but the same value.

```
#define ECUM_WKSOURCE_POWER (EcuM_WakeupSourceType)(1UL)
#define ECUM_WKSOURCE_POWER_1 (EcuM_WakeupSourceType)(1UL)
```

When does this happen:

During generation of the code.

In which configuration does this happen:

Only in variant configurations.

Resolution Description:

Workaround:

The defines with the numerical postfix can be just ignored.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099105 Compiler error: Rte.c accesses Rte_ActivationVector variable that is declared static in Rte_<osappl>.c	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.15.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms): ----- Compilation fails because a COM callback sets an activation reason in an Rte_ActivationVector variable that is declared in another file. The declaration is static.	
When does this happen: ----- The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen: ----- This happens when the configuration uses memory protection or multiple cores and when activation reasons are configured for a runnable that is triggered by a data reception, data reception error, or data send completion trigger of a COM signal or LDCOM PDU.	
Resolution Description:	
Workaround: ----- Use distinct runnables instead of the activation reason feature.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00099156		Missing description of OS_VTH_FORCED_TERMINATION and OS_VTHP_THREAD_CLEANUP
Component@Subcomponent:		Os_CoreGen7@Doc_TechRef
First affected version:		1.00.01
Fixed in versions:		
Problem Description:		
What happens (symptoms):		

Missing description of OS_VTH_FORCED_TERMINATION and OS_VTHP_THREAD_CLEANUP		
When does this happen:		

Always		
In which configuration does this happen:		

All		
Resolution Description:		
Workaround:		

No workaround available.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00099160 Patch action fails because file path is too long**Component@Subcomponent:** _3rdParty_McalIntegration_Helper@VectorIntegration**First affected version:** 1.00.00**Fixed in versions:** 2.03.01**Problem Description:**

What happens (symptoms):

Performing patch actions on a SIP where the path and/or file names are too long avoid the execution of the patch action. The 3rd Party Integration Helper Tool reports an error in the GUI.

When does this happen:

If files which have to be patched have a too long path or file name.

In which configuration does this happen:

All with long path names (> 259 characters).

Resolution Description:

Workaround:

Copy the SIP into a directory with short path. If error still occurs try again with a shorter path. Try as long as path is so short that the error does not occur.

Resolution:

The described issue is corrected by modification of all affected work-products:

- PatchFiles.PatchFileGeneric(...):
 - redesign
 - use LongPathFile.Open(...) to open files because this method can handle long paths
- FileDirectoryExt.CopyFileLong(...): add log entry if file which have to be copied does not exist
- LongPathFile.CheckReadOnly(...): fix spelling error in

Fixed in following RC:

Components.Mcal2: 3rdParty_McalIntegration_Helper@root[2.03.01.2]

ESCAN00099169 Compiler error/warning: Unreferenced formal parameter watchdog in actX25519.c	
Component@Subcomponent:	SysService_CryptoCv@Impl_actCLib
First affected version:	2.04.00
Fixed in versions:	2.09.02
Problem Description:	
What happens (symptoms):	

Compiler error due to unreferences parameter can occure.	
1> actX25519core.c	
1>..\..\..\external\BSW\SecMod\actX25519.c(108): error C4100: 'watchdog' : unreferenced formal parameter	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

If ED25519 is used and compiled.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099179 Compiler error: MemMap_Common.h: Wrong pragma command / unknown memory section used	
Component@Subcomponent:	Tp_Asr4TpCan@Implementation
First affected version:	3.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

The following error is reported during compilation:	
MemMap_Common.h:1763: #error "MemMap_Common.h: Wrong pragma command / unknown memory section used. Please use only valid pragma commands and known memory sections."	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens in configurations having more than 254 Tx SDUs or more than 254 Rx SDUs.	
Resolution Description:	
Workaround:	

Add the missing memory sections (CANTP_START_SEC_VAR_NOINIT_16BIT and CANTP_STOP_SEC_VAR_NOINIT_16BIT) to the MemMap.h file.	
Resolution:	

Not yet available.	

ESCAN00099186**Compiler error: Inconsistent setting for number of channels; with dynamic channel assignment, more SDUs than channels are expected****Component@Subcomponent:** Tp_Asr4TpCan@Implementation**First affected version:** 2.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

The following error is issued by the preprocessor:

Inconsistent setting for number of channels; with dynamic channel assignment, more SDUs than channels are expected

When does this happen:

fatal error C1189: #error : "Inconsistent setting for number of channels; with dynamic channel assignment, more SDUs than channels are expected"

In which configuration does this happen:

This happens in configurations where all of the following conditions are fulfilled:

1. Only Rx SDUs have been configured (no CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu containers are present).
2. Dynamic Channel assignment is enabled (CanTp/CanTpGeneral/CanTpDynamicChannelAssignment is set).
3. "Postbuild-Selectable" is enabled for the module or "POST-BUILD-LOADABLE" has been selected as the module's "Implementation Variant" (In any of the two cases the macro CANTP_USE_INIT_POINTER will be defined as STD_ON in the generated file CanTp_Cfg.h).

Resolution Description:

Workaround:

Add a user config file to the CanTp with the following code:

```
#if (CANTP_USE_INIT_POINTER != STD_ON) && (CANTP_DYN_CHANNEL_ASSIGNMENT ==  
STD_ON) && (CANTP_NUM_TX_SDUS == 0)  
# undef CANTP_NUM_TX_CHANNELS  
# define CANTP_NUM_TX_CHANNELS 0  
#endif
```

Resolution:

Not yet available.

ESCAN00099189	a2I: Calculation of CAN-FD parameter SECONDARY_SAMPLE_POINT in CanXcp.a2I is incorrect
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	3.00.02
Problem Description: What happens (symptoms): ----- The generated fragment CanXcp.a2I contains an CAN-FD specific parameter named SECONDARY_SAMPLE_POINT. This parameter is calculated incorrectly. As a result CAN communication on CAN-FD networks might not work due to this incorrect parameter. When does this happen: ----- Always and immediately In which configuration does this happen: ----- When CAN-FD is used for XCP communication.	
Resolution Description: Workaround: ----- The generated a2I file CanXcp.a2I is a textfile that is used outside of ECU software and can therefore be modified manually with a correct parameter for SECONDARY_SAMPLE_POINT Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00099274 Null pointer exception when data mapping exists in all variants but signal group does not

Component@Subcomponent: Rte_Asr4@GenTool_GeneratorMsr

First affected version: 4.05.00

Fixed in versions: 4.19.00

Problem Description:

What happens (symptoms):

RTE generation aborts with a null pointer exception.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the configuration contains complex data elements that are mapped to a system signal group in all variants and when a COM signal group does not exist for the system signal group in one of the variants.

Resolution Description:

Workaround:

Create a variation point for the data mapping.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099313 Test case 109 of WdgM Verifier does not fail anymore

Component@Subcomponent: SysService_Asr4WdM@root

First affected version: 2.01.00

Fixed in versions: 2.01.02

Problem Description:

What happens (symptoms):

In the Safety Manual one safety manual item (SMI-1578) describes, that a test case could fail under certain circumstances.

This test runs now always correct and must be passed during the test procedure.

When does this happen:

If a MICROSAR OS Gen 7 is used in combination with WdgM.

Resolution Description:

Workaround:

Test case 109 must be PASSED.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099318 Test case 107 of WdgM Verifier is marked as NOT PASSED	
Component@Subcomponent:	SysService_Asr4WdM@root
First affected version:	2.01.00
Fixed in versions:	2.01.03
Problem Description:	
What happens (symptoms):	

Test case 107 of WdgM Verifier could be marked as NOT PASSED erroneously.	
When does this happen:	

Always if the WdgM Verifier is used and test case 107 is marked as NOT PASSED.	
Resolution Description:	
Workaround:	

The user of MICROSAR Safe shall verify manually if 'WdgMTriggerTimeout' in instance(s) of WdgM_TriggerModeType (in WdgM_PBcfg.c) has a value greater than 0 if a 'WdgMWatchdogMode' other than WDGIF_OFF_MODE is configured. The WdgM Verifier cannot not verify this step currently.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099345 Exception in Generator when XcpCalibration container is not present**Component@Subcomponent:** Cp_Asr4Xcp@GenTool_GeneratorMsr**First affected version:** 1.00.00**Fixed in versions:** 3.00.02**Problem Description:**

What happens (symptoms):

The XCP generator raises an exception during code generation when the container /MICROSAR/Xcp/XcpCmdConfig/XcpCalibration is not present. This container is used to enable/disable XCP write access depending on whether this container is present/deleted.

As a result of this issue write access can not be disabled by this configuration option but must be disabled by write access mechanism during runtime.

When does this happen:

Always and immediately

In which configuration does this happen:

When /MICROSAR/Xcp/XcpCmdConfig/XcpCalibration is not present
AND
/MICROSAR/Xcp/XcpCmdConfig/XcpDaqAndStim is present

Resolution Description:

Workaround:

Add container /MICROSAR/Xcp/XcpCmdConfig/XcpCalibration and perform write access check during runtime in the call-back XcpAppl_CalibrationWrite.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099398 Compiler error: Incorrect expansion of Com_ReceiveShadowSignal with COM_RECEIVE_SIGNAL_MACRO_API**Component@Subcomponent:** IL_AsrComCfg5@Implementation**First affected version:** 8.01.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

Compile error occurs if Com_ReceiveShadowSignal is used having COM_RECEIVE_SIGNAL_MACRO_API enabled:

In file included from ..:

```
src/Com.h:718:54: error: pasting "Com_Get" and "(" does not give a valid preprocessing token
# define Com_ReceiveSignal(SignalId, SignalDataPtr) Com_Get##SignalId((SignalDataPtr))
^
```

```
src/Com.h:739:66: note: in expansion of macro 'Com_ReceiveSignal'
# define Com_ReceiveShadowSignal(SignalId, SignalDataPtr) (void)
```

```
Com_ReceiveSignal((SignalId), (SignalDataPtr))
^
```

```
test/test_all.c:519:3: note: in expansion of macro 'Com_ReceiveShadowSignal'
Com_ReceiveShadowSignal(GrpSig_1, someBuffer)
```

```
^
```

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

/MICROSAR/Com/ComGeneral/ComReceiveSignalMacroAPI is enabled

AND

Com_ReceiveShadowSignal is used

Resolution Description:

Workaround:

Use Com_ReceiveSignal API instead of deprecated Com_ReceiveShadowSignal API

OR

Disable /MICROSAR/Com/ComGeneral/ComReceiveSignalMacroAPI

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099413 Compiler error: Duplicated variable definition in case of N:M communication with external and internal receivers	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

Compilation fails because the RTE declares multiple variables with the same name.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when the configuration contains multiple senders that send the same data element and when	
the data element is mapped to a COM signal or a LDCOM PDU and is also sent to an additional internal receiver on the same ECU.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099473 The value <X> is not in the range of the specified datatype UINT_16	
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	5.00.00
Fixed in versions:	15.00.00
Problem Description:	
What happens (symptoms):	

RuntimeExeption "COM90500 The value <X> with comment () is not in the range of the specified datatype" is thrown during generation of the module.	
When does this happen:	

The error occurs during generation of COM	
In which configuration does this happen:	

COM has Rx I-PDUs configured with a /MICROSAR/EcuC/EcucPduCollection/Pdu/PduLength greater than UINT_16 and /MICROSAR/Com/ComConfig/ComIPdu/ComIPduSignalProcessing set to DEFERRED.	
Resolution Description:	
Workaround:	

set /MICROSAR/Com/ComConfig/ComIPdu/ComIPduSignalProcessing to IMMEDIATE	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099474 a2l: Parameter XCP_MAX_ODT_ENTRY_SIZE fixed to 7	
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

The a2l parameter XCP_MAX_ODT_ENTRY_SIZE is currently fixed to 7. This is correct for CAN but leads to warnings in the Tool should another bus system with a bigger payload should be used.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

On bus systems other than standard CAN.	
Resolution Description:	
Workaround:	

Usually this issue does not lead to problems because the command GET_DAQ_RESOLUTION_INFO returns the correct value, overwriting the a2l file.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099518 CanXcp.a2l not variant specific	
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	3.00.02
Problem Description:	
What happens (symptoms):	

The generated file CanXcp.a2l is not variant specific. For multiple variants only one file with the same name is generated. As a result an exception is thrown and only the last variant is available as file.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

When variance is used	
Resolution Description:	
Workaround:	

The generated a2l file can be copied and manually adapted to cover the missing variant.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099525 CanTpEnableSynchronousTransmit cannot be used with non MICROSAR Components	
Component@Subcomponent:	Tp_Asr4TpCan@Doc_TechRef
First affected version:	3.01.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

Add a warning box to chapter 3.1.2.8 and to the integration chapter that CanTpEnableSynchronousTransmit cannot be used with non MICROSAR components.	
When does this happen:	

At runtime.	
In which configuration does this happen:	

Any configuration where CanTp/CanTpGeneral/CanTpEnableSynchronousTransmit is configured to true.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

Not yet available.	

ESCAN00099526 CanTpEnableSynchronousTransmit cannot be used with non MICROSAR Components	
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	5.00.00
Fixed in versions:	14.02.00
Problem Description:	
What happens (symptoms):	

The feature CanTpEnableSynchronousTransmit cannot be used with non MICROSAR components because the functionality is not specified by AUTOSAR.	
When does this happen:	

At runtime.	
In which configuration does this happen:	

Any configuration where the PduR is used and suggests to set CanTpEnableSynchronousTransmit to true.	
Resolution Description:	
Workaround:	

Set the CanTp/CanTpGeneral/CanTpEnableSynchronousTransmit to UserDefined and false.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099539 RTE49999: N:1 Inter-Partition Client-Server Communication with IOCs	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.03.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

RTE Generation aborts with an RTE49999 error.	
When does this happen:	

During generation.	
In which configuration does this happen:	

All of the following conditions have to be fulfilled:	
<ul style="list-style-type: none"> - IOCs are used AND - a server implemented by a runnable has at least three clients AND - the server runnable is <ul style="list-style-type: none"> a) either unmapped or b) at least one client runnable is mapped to the same task and no other client runnables are mapped to other task in the server's OS application AND - at least two other OS applications have client runnables with asynchronous or scheduled communication. 	
Resolution Description:	
Workaround:	

Assign the client runnable which shared the OS application with the server runnable to another OS application.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099548	InitMonitorReason DEM_INIT_MONITOR_REENABLED is missing in callback description
Component@Subcomponent:	Diag_Asr4Dem@Doc_TechRef
First affected version:	3.00.00
Fixed in versions:	9.03.00
Problem Description:	
What happens (symptoms):	

In chapter 6.5.1.4 CbInitEvt_<EventName>(), the InitMonitorReason DEM_INIT_MONITOR_REENABLED is missing.	
When does this happen:	

Always and immediately	
In which configuration does this happen:	

all	
Resolution Description:	
Workaround:	

Possible parameter values for InitMonitorForEvent callback are the possible values for Dem_InitMonitorReasonType as specified e.g. in Document ID 019:	
AUTOSAR_SWS_DiagnosticEventManager, version 4.1.2.	
Add "DEM_INIT_MONITOR_REENABLED: Enable conditions fulfilled for event or Control DTC settings enabled." to chapter "6.5.1 Callouts -> 6.5.1.4 CbInitEvt_<EventName>()"	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099553	State diagram of the EcuM with fixed state machine shows call of EcuM_AL_DriverRestart in the wrong transition.
Component@Subcomponent:	SysService_Asr4EcuM@Doc_TechRef
First affected version:	3.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	-----
	State diagram of the EcuM with fixed state machine shows call of EcuM_AL_DriverRestart in the transition from ECUM_STATE_GO_SLEEP to ECUM_STATE_WAKEUP_VALIDATION.
	The call should be located in the transition from ECUM_STATE_SLEEP to ECUM_STATE_WAKEUP_VALIDATION.
When does this happen:	-----
	During reading of the TechRef.
In which configuration does this happen:	-----
	Only in EcuM with fixed behavior configurations (EcuM/EcuMGeneral/EcuMEnableFixBehavior == true).
Resolution Description:	
Workaround:	-----
	No workaround available.
Resolution:	-----
	The described issue is corrected by modification of all affected work-products.

ESCAN00099582 Compiler error: actAES.h:23 missing argument for macro P2FUNC	
Component@Subcomponent:	SysService_CryptoCv@Impl_actCLib
First affected version:	2.00.00
Fixed in versions:	3.02.00, 2.10.01
Problem Description:	
What happens (symptoms):	

actAES.h:23 23 missing argument for macro P2FUNC:23 23 missing argument for macro P2FUNC	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

If aes is used.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099596 Compiler error: Missing dataSig variables	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.12.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

Compilation fails because a Rte_COMCbk_ callback accesses a variable dataSig<DataType> that is not declared.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when a SystemSignal or ISignal references a compu method with factor and offset and when the signal is mapped to a data element with float data type and when the receiver is mapped to a different partition than the BSW.	
Resolution Description:	
Workaround:	

Create a proxy component that receives the data element and forward the converted data to the original receiver.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099599 Dem_SetOperationCycleState can not be called in the pre-initialized mode	
Component@Subcomponent:	Diag_Asr4Dem@Doc_TechRef
First affected version:	1.00.00
Fixed in versions:	9.03.00
Problem Description:	
What happens (symptoms):	

The description of API Dem_SetOperationCycleState says that it can be called in preinitialized mode, but this results in DET and negative return value.	
When does this happen:	

Always	
In which configuration does this happen:	

Always	
Resolution Description:	
Workaround:	

Rework chapter 6.2.6.6 Dem_SetOperationCycleState()->Functional Description:	
Instead of	
"It is allowed to call this run in pre-initialized mode to start the operation cycle of BSW events before full initialization. "	
write	
"The API can not be used until Dem_MasterInit() has completed.".	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099644 Compiler error: QOverflowType struct without declaration	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.02.00
Fixed in versions:	1.19.00
Problem Description: What happens (symptoms): ----- The compiler states: QOverflowType expected a declaration ----- When does this happen: ----- The error is issued by the compiler during compilation of the code in case the configuration is as described below. ----- In which configuration does this happen: ----- Inter-partition communication used in combination with array of variable size	
Resolution Description: Workaround: ----- There are two possible workarounds 1. Change the variable array to fixed if the length information is not needed or is sent through a different interface. 2. Replace the array data type in the data element with a record data type with two elements - integer element (for the size) - array element with the maximum array size ----- The application can then set and get the actual array size from the size element. The record with integer size and array can then be directly passed to the RTE APIs. ----- Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00099667 Null pointer exception when no BswImplementation for LDCOM exists	
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.04.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

RTE generation aborts with a null pointer exception.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains a non Vector LDCOM module without BSW Implementation is configured. e.g. when the AUTOSAR definition is used	
Resolution Description:	

ESCAN00099667 Null pointer exception when no BswImplementation for LDCOM exists

Workaround:

Manually add a MODULE-DESCRIPTION-REF to the ECUC file

```
<AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:schemaLocation="http://autosar.org/schema/r4.0
AUTOSAR_4-2-1.xsd">
  <AR-PACKAGES>
    <AR-PACKAGE>
      <SHORT-NAME>ActiveEcuC</SHORT-NAME>
      <ELEMENTS>
        <ECUC-MODULE-CONFIGURATION-VALUES UUID="f21af5ba-362f-40e2-be00-da66c2aaebda">
          <SHORT-NAME>LdCom</SHORT-NAME>
          <DEFINITION-REF DEST="ECUC-MODULE-DEF">/AUTOSAR/EcucDefs/LdCom</DEFINITION-REF>
          <IMPLEMENTATION-CONFIG-VARIANT>VARIANT-PRE-COMPILE</IMPLEMENTATION-CONFIG-
VARIANT>
          <MODULE-DESCRIPTION-REF DEST="BSW-IMPLEMENTATION">/MICROSAR/LdCom_Impl</
MODULE-DESCRIPTION-REF>
        </ECUC-MODULE-CONFIGURATION-VALUES>
      </ELEMENTS>
    </AR-PACKAGE>
  </AR-PACKAGES>
</AUTOSAR>
```

And add an arxml file with the following content to the InternalBehavior directory of the configuration:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:schemaLocation="http://autosar.org/schema/r4.0
AUTOSAR_4-2-1.xsd">
  <AR-PACKAGES>
    <AR-PACKAGE UUID="ff55d029-386d-43a2-b3c2-dfef4c7a5e36">
      <SHORT-NAME>MICROSAR</SHORT-NAME>
      <ELEMENTS>
        <BSW-IMPLEMENTATION UUID="fb38aa70-5ae2-4fd0-8ecf-7d91f63bcc9c">
          <SHORT-NAME>LdCom_Impl</SHORT-NAME>
          <PROGRAMMING-LANGUAGE>C</PROGRAMMING-LANGUAGE>
          <SW-VERSION>2.00.00</SW-VERSION>
          <USED-CODE-GENERATOR>DaVinci Configurator</USED-CODE-GENERATOR>
          <VENDOR-ID>30</VENDOR-ID>
          <AR-RELEASE-VERSION>4.02.02</AR-RELEASE-VERSION>
        </BSW-IMPLEMENTATION>
      </ELEMENTS>
    </AR-PACKAGE>
    <AR-PACKAGE UUID="53f8d579-f844-4767-8c49-63a1009124f9">
      <SHORT-NAME>LdCom_ib_bswmd</SHORT-NAME>
      <AR-PACKAGES>
        <AR-PACKAGE UUID="706f5b57-21d1-4ddc-b79d-3f2a27e3fc54">
          <SHORT-NAME>BswModuleDescriptions</SHORT-NAME>
          <ELEMENTS>
            <BSW-MODULE-DESCRIPTION UUID="6c8dc0dd-d742-4f2d-a4b2-200794c35a12">
              <SHORT-NAME>LdCom</SHORT-NAME>
              <INTERNAL-BEHAVIORS>
                <BSW-INTERNAL-BEHAVIOR UUID="938e0732-4116-43de-a78b-52a578a001bc">
                  <SHORT-NAME>LdComBehavior</SHORT-NAME>
                </BSW-INTERNAL-BEHAVIOR>
              </INTERNAL-BEHAVIORS>
            </BSW-MODULE-DESCRIPTION>
          </ELEMENTS>
        </AR-PACKAGE>
      </AR-PACKAGES>
    </AR-PACKAGE>
  </AR-PACKAGES>
</AUTOSAR>
```


ESCAN00099667 Null pointer exception when no BswImplementation for LDCOM exists

```
<SHORT-NAME>XcpEvents</SHORT-NAME>
</AR-PACKAGE>
</AR-PACKAGES>
</AR-PACKAGE>
</AR-PACKAGES>
</AR-PACKAGE>
</AR-PACKAGES>
</AUTOSAR>
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099693 Compiler error: Incompatible Declaration in Rte_Csm.h and Csm.h

Component@Subcomponent: SysService_AsrCsm@Implementation

First affected version: 2.02.00

Fixed in versions:

Problem Description:

What happens (symptoms):

e.g.: declaration is incompatible with "Std_ReturnType Csm_SymEncryptStart(..."

When does this happen:

The error is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

If the underlying Cry Driver includes Csm.h instead of Csm_Types.h

Resolution Description:

Workaround:

Adding the following three lines in a global include file (e.g. Compiler_Cfg.h):

```
#ifdef CSM_CFG_SOURCE
#define _RTE_CSM_H
#endif
```

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099714 Compiler error/warning: argument type of string of Ioc_Write does not match prototype for implicit write	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

Typical compiler warning / error explanations may be:	
"Rte\\Rte_*.c": argument type does not match prototype	
for code of type	
(void)IocWrite_Rte_<...>(*(&<DataElement>));	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

When an implicit write access for a byte array uses IOCs.	
Resolution Description:	
Workaround:	

Use explicit write.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099814 Wrong references to CanTp_Cfg.c exist	
Component@Subcomponent:	Tp_Asr4TpCan@Doc_TechRef
First affected version:	3.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

CanTp_Cfg.c is mentioned as a generated file, but the file is actually not generated.	
When does this happen:	

Always and Immediately.	
In which configuration does this happen:	

All of them.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

Not yet available.	

ESCAN00099816 Compiler error: Missing buffer definition within struct Rte_tsRB	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.07.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

Compilation fails because a task accesses a Rte_RB structure element that does not exist.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens if the config contains a component with multiple instantiations and implicit accesses and when one instance of the component can access the data element without temporary buffer and one instance requires a temporary buffer.	
Resolution Description:	
Workaround:	

Configure invalidation or never received handling.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099948 Compiler error: Duplicated lower limit variable in RTE Analyzer stubs	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.14.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

Analysis with RTE Analyzer fails.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when a component uses different implementation data types with the same name that are located in different AUTOSAR packages.	
Resolution Description:	
Workaround:	

Remove duplicated data types.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099951 Compiler error: CanNm_SetMsgRequest undefined	
Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	6.02.00
Fixed in versions:	8.00.00
Problem Description:	
What happens (symptoms):	

A compiler warning similar to the following occurs: "CanNm_SetMsgRequest' undefined; assuming extern returning int"	
This also leads to a linker error: "unresolved external symbol _CanNm_SetMsgRequest referenced in function _CanNm_SetSleepReadyBit"	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

IF	
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmComUserDataSupport' is disabled	
AND	
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmCoordinatorSyncSupport' is enabled	
Resolution Description:	
Workaround:	

Use '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmComUserDataSupport' in case User Data support AND '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmCoordinatorSyncSupport' is needed.	
In case User Data support is not needed, '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmComUserDataSupport' must still be activated.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099953 Compiler error: Too many struct initializers when InitializeImplicitBuffers is configured	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

Compilation fails, because the initializer for the implicit task buffer does not match the type of the variable.	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when InitializeImplicitBuffers is configured to true and when the configuration contains implicit read accesses that only receive a subset of the sent data.	
Resolution Description:	
Workaround:	

Create a wrapper SWC that receives the full data element and only forwards the relevant data to the implicit receiver.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099959 Compiler error: undefined reference to `CanTp_IsTxSduCfgIndUsedOfRxPduMap'	
Component@Subcomponent:	Tp_Asr4TpCan@Implementation
First affected version:	2.01.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

The following errors are shown when trying to build the project:	
CanTp.c:(.text.CanTp_RxIndication+0x30): error: undefined reference to `CanTp_IsTxSduCfgIndUsedOfRxPduMap'	
CanTp.c:(.text.CanTp_RxIndication+0x3a): error: undefined reference to `CanTp_GetTxSduCfgIndStartIdxOfRxPduMap'	
CanTp.c:(.text.CanTp_RxIndication+0x40): error: undefined reference to `CanTp_GetTxSduCfgInd'	
CanTp.c:(.text.CanTp_RxIndication+0x436): error: undefined reference to `CanTp_IsTxSduCfgIndUsedOfRxPduMap'	
When does this happen:	

The error is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens in configurations fulfilling all of the following conditions:	
- The CanTp is configured to support Postbuild-Selectable (the macro CANTP_POSTBUILD_VARIANT_SUPPORT in CanTp_Cfg.h is defined as STD_ON)	
- The Implementation variant is set to VARIANT-PRE-COMPILE (the macro CANTP_CONFIGURATION_VARIANT in CanTp_Cfg.h is defined as CANTP_CONFIGURATION_VARIANT_PRECOMPILE)	
- No Tx SDUs are configured (the macro CANTP_NUM_TX_SDUS in CanTp_Cfg.h is defined as 0)	
Resolution Description:	

**ESCAN00099959 Compiler error: undefined reference to
`CanTp_IsTxSduCfgIndUsedOfRxPduMap'**

Workaround:

Add a user configuration file to the CanTp (CanTp/CanTpGeneral/CanTpUserConfigFile) with the following lines:

```
#if (CANTP_POSTBUILD_VARIANT_SUPPORT == STD_ON) &&  
(CANTP_CONFIGURATION_VARIANT == CANTP_CONFIGURATION_VARIANT_PRECOMPILE) &&  
(CANTP_NUM_TX_SDUS == 0)
```

```
# if !defined (CanTp_IsTxSduCfgUsedOfTxSduSnv2Hdl)  
# define CanTp_IsTxSduCfgUsedOfTxSduSnv2Hdl(x) FALSE  
# endif
```

```
# if !defined (CanTp_GetTxSduCfgIdxOfTxSduSnv2Hdl)  
# define CanTp_GetTxSduCfgIdxOfTxSduSnv2Hdl(x)(PduIdType)0  
# endif
```

```
# if !defined (CanTp_IsTxSduCfgUsedOfRxSduCfg)  
# define CanTp_IsTxSduCfgUsedOfRxSduCfg(x) FALSE  
# endif
```

```
# if !defined (CanTp_GetTxSduCfgIdxOfRxSduCfg)  
# define CanTp_GetTxSduCfgIdxOfRxSduCfg(x) (PduIdType)0  
# endif
```

```
# if !defined (CanTp_IsTxSduCfgIndUsedOfRxPduMap)  
# define CanTp_IsTxSduCfgIndUsedOfRxPduMap(x) FALSE  
# endif
```

```
#endif
```

Resolution:

Not yet available.

ESCAN00099970 Wrong error message for ScheduleTable ExpiryPoint offset	
Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	2.21.00
Problem Description: What happens (symptoms): ----- The '/MICROSAR/Os/OsScheduleTable/OsScheduleTableExpiryPoint/OsScheduleTblExpPointOffset' can not be configured to the value of '/MICROSAR/Os/OsCounter/OsCounterMinCycle' for the first ExpiryPoint (with the smallest offset) of a ScheduleTable. If configured so, the DaVinci CFG5 shows the following error message (here with example values): OS03430 OsScheduleTblExpPointOffset(value=200) must be greater than 200. Set OsScheduleTblExpPointOffset(value=200) to 201 /ActiveEcuC/Os/SystemTimer[OsCounterMinCycle] /ActiveEcuC/Os/MyScheduleTable/OsScheduleTableExpiryPoint[OsScheduleTblExpPointOffset] ----- When does this happen: ----- At configuration time. ----- In which configuration does this happen: ----- If the '/MICROSAR/Os/OsScheduleTable/OsScheduleTableExpiryPoint/OsScheduleTblExpPointOffset' of the first OsScheduleTableExpiryPoint is configured to the same value as '/MICROSAR/Os/OsCounter/OsCounterMinCycle' of the driving Counter of the ScheduleTable. Which is referenced in '/MICROSAR/Os/OsScheduleTable/OsScheduleTableCounterRef'.	
Resolution Description: Workaround: ----- Either decrease '/MICROSAR/Os/OsCounter/OsCounterMinCycle' or increase '/MICROSAR/Os/OsScheduleTable/OsScheduleTableExpiryPoint/OsScheduleTblExpPointOffset'. ----- Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00099987 RTE49999: when union is used for N:1 connections or PR ports	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.14.00
Fixed in versions:	1.19.00
Problem Description:	
What happens (symptoms):	

RTE generation aborts with a RTE49999 exception.	
When does this happen:	

During generation.	
In which configuration does this happen:	

This happens when the configuration contains union data types that are used for N:1 sender receiver connections or for PR ports.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099988		Description: Lower Limit for parameter DemExtendedDataRecordNumber wrong
Component@Subcomponent:	Diag_Asr4Dem@Description	
First affected version:	4.00.00	
Fixed in versions:		
Problem Description:		
What happens (symptoms):		

"Min Value" of /MICROSAR/Dem/DemGeneral/DemExtendedDataRecordClass/ DemExtendedDataRecordNumber can be configured to value 0 although this value is reserved (by UDS diagnostic specification and later AUTOSAR specifications). No error of Configuration-Tool is given. "Min. Value" should be 1.		
When does this happen:		

At configuration time if creating Extended Data Record configuration.		
In which configuration does this happen:		

In configurations supporting UDS service 0x19 with sub-function 0x06 (reportDTCExtDataRecordByDTCNumber) or 0x10 (reportMirrorMemoryDTCExtDataRecordByDTCNumber).		
Resolution Description:		
Workaround:		

Do not use ExtendedDataRecordNumber with value 0.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00100169 Parameter description of CheckRemoteSleepIndication is incorrect	
Component@Subcomponent:	Nm_Asr4NmCan@Doc_TechRef
First affected version:	1.00.00
Fixed in versions:	7.02.02
Problem Description:	
What happens (symptoms):	

The description of the input parameter 'nmRemoteSleepIndPtr' is incorrect.	
The current chapter CanNm_CheckRemoteSleepIndication states:	
'Pointer where PDU Data out of the most recently received NM message shall be copied to'	
but should be:	
'Pointer where current remote sleep state is copied to.'	
When does this happen:	

-	
In which configuration does this happen:	

-	
Resolution Description:	
Workaround:	

Parameter description for the CanNm_CheckRemoteSleepIndication API should be as follows:	
'nmRemoteSleepIndPtr : Pointer where current remote sleep state is copied to.'	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00100193 Improve description**Component@Subcomponent:** AN-ISC-8-1184_Compiler_Warnings@Doc_ApplicationNote**First affected version:** 1.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

The application note can be misinterpreted how globally accepted compiler warnings are handled.

When does this happen:

When reading the application note.

In which configuration does this happen:

N/A**Resolution Description:**

Workaround:

Keep this ESCAN in mind when reading the application note

Resolution:

The described issue is corrected by modification of all affected work-products.**ESCAN00100240 RTE generator creates duplicated A2L group objects in case of PR Ports****Component@Subcomponent:** Rte_Core@Implementation**First affected version:** 1.04.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

An invalid A2L measurement description file is generated that contains duplicated group objects.

When does this happen:

During generation.

In which configuration does this happen:

This happens when measurement is enabled for PR Ports.**Resolution Description:**

Workaround:

Manually remove duplicated group entries from the A2L description.

Resolution:

The described issue is corrected by modification of all affected work-products.

2.6 Compiler Warnings

As a service we also provide the known compiler warnings. The occurrence of a compiler warning may depend on the used software module configuration and compiler settings.

Index

ESCAN00065890	Compiler warning: cast discards '___attribute__((noreturn))' qualifier from pointer target type DrvCan_Mpc5700McanLI@Implementation
ESCAN00065891	Compiler warning: cast increases required alignment of target type DrvCan_Mpc5700McanLI@Implementation
ESCAN00067159	Compiler warning: cast truncates constant value MemService_AsrNmM@Implementation
ESCAN00068434	Compiler warning: conditional expression or part of it is always true/false DrvCan__coreAsr@Implementation
ESCAN00068872	Compiler warning: the order of volatile accesses is undefined in this statement DrvCan__coreAsr@Implementation
ESCAN00074793	Compiler warning: Condition is always constant Diag_Asr4Dem@Implementation
ESCAN00086650	Compiler warning: pointless comparison of unsigned integer with zero SysService_AsrCsm@Implementation
ESCAN00088061	BswM_Lcfg.c: warning: 'function' : conversion from 'const BswM_ImmediateUserStartIdxOfModeReqeustMappingType' to 'BswM_SizeOfImmediateUserType', possible loss of data SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00089241	Compiler warning: multiple warnings SysService_CryptoCv@Impl_actCLib
ESCAN00089425	Compiler warning: missing braces around initializer SysService_CryptoCv@Impl_ESLib
ESCAN00089543	Compiler warning: dead assignment to "errorId" eliminated Nm_Asr4NmIf@Implementation
ESCAN00089544	Compiler warning: conversion to 'uint8' from 'int' may alter its value Nm_Asr4NmIf@Implementation
ESCAN00090114	Compiler Warning: Assignment in condition SysService_CryptoCv@Impl_actCLib
ESCAN00090161	Compiler warning: condition evaluates always to true/false Ccl_Asr4ComMCfg5@Implementation
ESCAN00090806	Compiler warning: C4310: cast truncates constant value Gw_AsrPduRCfg5@Implementation
ESCAN00090831	Compiler warning: integer conversion resulted in a change of sign IL_AsrComCfg5@Implementation
ESCAN00091295	Compiler warning: dead assignment / variable set but not used Ccl_Asr4ComMCfg5@Implementation
ESCAN00091340	Compiler warning: cast truncates constant value If_AsrIfCan@Implementation
ESCAN00091343	Compiler warning: warning C4310: cast truncates constant value If_AsrIfCan@Implementation
ESCAN00091547	Compiler warning: condition is always false Diag_Asr4Dem@Implementation
ESCAN00092315	Compiler warning: function "CanLL_WakeUpHandling" was declared but never referenced DrvCan_Mpc5700McanLI@Implementation
ESCAN00092713	Preprocessor parse error DrvCan_Mpc5700McanLI@Implementation
ESCAN00094232	Compiler warning: "conditional expression is constant" Nm_Asr4NmCan@Implementation

Index

ESCAN00095299	Compiler warning: Mismatch between signature of function declarations and definitions Diag_Asr4Dcm@Implementation
ESCAN00095459	Compiler warning: Crc.c: constant out of range SysService_AsrCrc@Implementation
ESCAN00095486	Compiler warning: Function "Dcm_Svc14_XX_RepeaterProxySelectDTC" was declared but never referenced Diag_Asr4Dcm@Implementation
ESCAN00095488	Datatype conversion for description routing, possible loss of Data Il_AsrComCfg5@Implementation
ESCAN00095508	Compiler warning: Function "Dcm_MemMgrWriteMemory" was declared but never referenced Diag_Asr4Dcm@Implementation
ESCAN00095513	Compiler warning: Function "Dcm_MemMgrReadMemory" was declared but never referenced Diag_Asr4Dcm@Implementation
ESCAN00095530	Compiler warning: variable "IERecStoredIndex" was set but never used Diag_Asr4Dem@Implementation
ESCAN00095542	Compiler warning: 'PduR_RmIf_SingleBufferHandling' declared 'static' but never defined Gw_AsrPduRCfg5@Implementation
ESCAN00095907	Compiler warning: Com.c(4142): warning C4100: 'idxFilterInfo' : unreferenced formal parameter Il_AsrComCfg5@Implementation
ESCAN00096038	Compiler warning: Passing of incompatible pointers Rte_Core@Implementation
ESCAN00096133	Compiler warning: warning C4310: cast truncates constant value Cp_Asr4Xcp@Implementation
ESCAN00096246	Compiler warning: C4100: 'CanId' : unreferenced formal parameter If_AsrIfCan@Implementation
ESCAN00096376	Compiler warning: Unreachable code in ESLib_EdDSA.c SysService_CryptoCv@Impl_ESLib
ESCAN00096911	Compiler warning: type qualifier is meaningless on cast type Cp_Asr4Xcp@Implementation
ESCAN00096913	Compiler warning: Static function 'Rte_QUnqueueElementCallbackSpecific<OsApplicationName>()' is not used within this translation unit Rte_Core@Implementation
ESCAN00097289	Compiler warning: integer literal is too large to be represented in a signed integer type, interpreting as unsigned Rte_Core@Implementation
ESCAN00097375	Compiler warning: 'ITxHdl' : local variable is initialized but not referenced Tp_Asr4TpCan@Implementation
ESCAN00097692	Compiler warning: conversion from 'int' to 'Dcm_CfgNetBufferSizeOptType' Diag_Asr4Dcm@Implementation
ESCAN00097790	Compiler warning: 'CanTpTxSduId' : unreferenced formal parameter Tp_Asr4TpCan@Implementation
ESCAN00097980	Compiler warning: Unreferenced formal parameter due to reduction of rom data Il_AsrComCfg5@Implementation
ESCAN00098038	Compiler warning: number of arguments don't match Rte_Core@Implementation
ESCAN00098070	Compiler warning: NvM_Cfg.c: 'ServiceId' : unreferenced formal parameter MemService_AsrNvM@GenTool_GeneratorMsr

Index

ESCAN00098195	Compiler warning: possible loss of data when least types are larger than the platform types Rte_Core@Implementation
ESCAN00099190	Compiler warning: BswM_Lcfg.c(2990): warning C4100: 'handleId' : unreferenced formal parameter SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00099286	Compiler warning: unused parameter 'ipduGroupVector' and 'initialize' in function 'Com_IpduGroupControl' Il_AsrComCfg5@Implementation
ESCAN00099287	Compiler warning: unused parameter 'deferredfctPtrCacheStrctPtr' in function 'Com_RxProcessDeferredPDU' Il_AsrComCfg5@Implementation
ESCAN00099291	Compiler warning: unused parameter 'PduInfoPtr' in function 'Com_RxSignalProcessing' Il_AsrComCfg5@Implementation
ESCAN00099292	Compiler warning: unused parameter 'idxTxSigInfo' in function 'Com_SendSignal_WriteSignal' Il_AsrComCfg5@Implementation
ESCAN00099981	Compiler warning: the order of volatile accesses is undefined Cp_Asr4Xcp@Implementation
ESCAN00100176	Compiler warning: cast truncates constant value If_AsrIfCan@Implementation
ESCAN00100182	Compiler warning: A value that cannot be used to initialize an entity with a function pointer type Gw_AsrPduRCfg5@GenTool_GeneratorMsr

ESCAN00065890 **Compiler warning: cast discards
'__attribute__((noreturn))' qualifier from pointer
target type****Component@Subcomponent:** DrvCan_Mpc5700McanLI@Implementation**First affected version:** 1.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

The compiler generates the following warning:

Compiling file: ../../external/BSW/Can/Can.c
../../external/BSW/Can/Can.c: In function 'CanBasicCanMsgReceived':
../../external/BSW/Can/Can.c:1745:16: warning: cast discards '__attribute__((noreturn))'
qualifier from pointer target type [-Wcast-qual]
../../external/BSW/Can/Can.c:1750:10: warning: cast discards '__attribute__((noreturn))'
qualifier from pointer target type [-Wcast-qual]
../../external/BSW/Can/Can.c:1780:55: warning: cast discards '__attribute__((noreturn))'
qualifier from pointer target type [-Wcast-qual]

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

GNU compiler and -Wcast-qual compiler option is used**Resolution Description:**

Workaround:

Omit gcc command option -Wcast-qual.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00065891 Compiler warning: cast increases required alignment of target type**Component@Subcomponent:** DrvCan_Mpc5700McanLI@Implementation**First affected version:** 1.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

Compiler generates the following warning:

Compiling file: ../../external/BSW/Can/Can.c
../../external/BSW/Can/Can.c: In function 'CanBasicCanMsgReceived':
../../external/BSW/Can/Can.c:1745:16: warning: cast increases required alignment of target type [-Wcast-align]
../../external/BSW/Can/Can.c:1750:10: warning: cast increases required alignment of target type [-Wcast-align]
../../external/BSW/Can/Can.c:1752:29: warning: cast increases required alignment of target type [-Wcast-align]
../../external/BSW/Can/Can.c:1758:30: warning: cast increases required alignment of target type [-Wcast-align]

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

GNU compiler and -Wcast-align compiler option is used**Resolution Description:**

Workaround:

Omit gcc command option -Wcast-align

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00067159 Compiler warning: cast truncates constant value**Component@Subcomponent:** MemService_AsrNvM@Implementation**First affected version:** 3.08.01**Fixed in versions:****Problem Description:**

What happens (symptoms):

```
>..\..\bsw\nvm\nvm_crc.c(229) : warning C4310: cast truncates constant value
```

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

CANoeEmu + VS2008

It depends on definition of uint16_least: Warning occurs only if uint16_least is not of type int.

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed, because the cast confirms and enforces this behavior (i.e. the value SHALL be truncated, if necessary).
Additionally: Why uint16_least is not (unsigned) int? -> this data type fulfills all requirements on a 16 bit unsigned value...

Resolution Description:

Workaround:

No workaround necessary.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00068434 Compiler warning: conditional expression or part of it is always true/false**Component@Subcomponent:** DrvCan__coreAsr@Implementation**First affected version:** 4.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

- Compiler warns for "condition is always true": This may happen depending on configuration, i.e. assert checks

in function Can_SetControllerMode following code is available

```
...
transitionRequest = kCanRequested;

CanMicroModeRestore();
}
if ( transitionRequest == CAN_NOT_OK ) /* PRQA S 3355,3356,3358,3359 */ /* MD_Can_13.7 */
{ /* PRQA S 3201 */ /* MD_Can_3201 */
    retval = CAN_NOT_OK;
    transitionDone = CAN_NOT_OK; /* at least one HW channel is not in new state (CAN_MSR40: poll later) */
}
..
```

this issues following compiler warning:

if (transitionRequest == CAN_NOT_OK) - warning (dcc:1606): conditional expression or part of it is always true/false

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

All configurations.

but not for all Platform implementations (hw always return OK for state transition)

Resolution Description:

Workaround:

Ignore warning

ESCAN00068872 Compiler warning: the order of volatile accesses is undefined in this statement	
Component@Subcomponent:	DrvCan__coreAsr@Implementation
First affected version:	3.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms): ----- Compiler issues warning messages like this: undefined behavior: the order of volatile accesses is undefined in this statement	
When does this happen: ----- The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen: ----- Rx Queue is enabled	
Resolution Description:	
Workaround: ----- Ignore Warning	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00074793 Compiler warning: Condition is always constant	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	4.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

Compiler warning 'Condition is always constant'	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Configurations without DTCs	
AND	
Precompile configuration	
Resolution Description:	
Workaround:	

The warning can be ignored	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00086650 Compiler warning: pointless comparison of unsigned integer with zero

Component@Subcomponent: SysService_AsrCsm@Implementation

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

The generic range check may produce a compiler warning if the CSM_OFFSET_* of a specific service is zero as the range check will compare "uint16 >= 0" which is always TRUE.

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

Every configuration with some compiler.

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00088061 BswM_Lcfg.c: warning: 'function' : conversion from 'const BswM_ImmediateUserStartIdxOfModeReqeustMappingType' to 'BswM_SizeOfImmediateUserType', possible loss of data

Component@Subcomponent: SysService_Asr4BswMCfg5@GenTool_GeneratorMsr

First affected version: 7.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

BswM_Lcfg.c: warning: 'function' : conversion from 'const BswM_ImmediateUserStartIdxOfModeReqeustMappingType' to 'BswM_SizeOfImmediateUserType', possible loss of data

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

All

Resolution Description:

ESCAN00089241 Compiler warning: multiple warnings	
Component@Subcomponent:	SysService_CryptoCv@Impl_actCLib
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

<ul style="list-style-type: none"> - Compiler warns for possible loss of data: Check if cast is missing and if there is really a data loss due to an implicit/explicit cast on the target platform - Compiler warns for ambiguous code, parentheses recommended. 	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Always.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00089425 Compiler warning: missing braces around initializer	
Component@Subcomponent:	SysService_CryptoCv@Impl_ESLib
First affected version:	1.01.01
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

Compiling file: ../../BSW/SecMod/ESLib_version.c	
ctc W542: ["../../BSW/SecMod/ESLib_version.c" 73/4] missing braces around initializer	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

In all configurations.	
Resolution Description:	
Workaround:	

Since ESLib_version.c is only used for component testing, it can be excluded from the build for integration.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00089543 Compiler warning: dead assignment to "errorId" eliminated**Component@Subcomponent:** Nm_Asr4NmIf@Implementation**First affected version:** 7.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

A compiler warning similar to the following one occurs for the compilation of Nm.c:
dead assignment to "errorId" eliminated

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

'Dev Error Detect' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalProperties/NmDevErrorDetect) in the NmGlobalProperties container is turned OFF in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_DEV_ERROR_REPORT STD_OFF).

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed due to the API pattern that Vector has decided to use: each API that may report development errors shall always have an errorId variable on the stack to which assignments are made - regardless of whether the variable is actually used or not.

Resolution Description:

Workaround:

Ignore the warning.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00089544 Compiler warning: conversion to 'uint8' from 'int' may alter its value

Component@Subcomponent: Nm_Asr4NmIf@Implementation

First affected version: 9.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Compiler warnings similar to the following one occur for the compilation of Nm.c:
conversion to 'uint8' from 'int' may alter its value

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

'Coordinator Support Enabled' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/
NmCoordinatorSupportEnabled) is turned ON in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_COORDINATOR_SUPPORT_ENABLED STD_ON)

AND

(
'Remote Sleep Ind Enabled' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/
NmRemoteSleepIndEnabled) is turned OFF in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_REMOTE_SLEEP_IND_ENABLED STD_OFF)

OR

all coordinated channels have 'Channel Sleep Master' (/MICROSAR/Nm/NmChannelConfig/
NmChannelSleepMaster) turned ON in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_OPTIMIZE_ALL_SLEEP_MASTER STD_ON)

OR

all coordinated channels have 'Synchronizing Network' (/MICROSAR/Nm/NmChannelConfig/
NmSynchronizingNetwork) turned ON in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_OPTIMIZE_ALL_SYNC_CHANNEL STD_ON)
)

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed because there is no risk of an invalid conversion of value to uint8.

Resolution Description:

Workaround:

Ignore the warning.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00090114 Compiler Warning: Assignment in condition	
Component@Subcomponent:	SysService_CryptoCv@Impl_actCLib
First affected version:	1.00.00
Fixed in versions:	2.09.00
Problem Description:	
What happens (symptoms): -----	
Compiling file: ../../BSW/SecMod/actBNReduce.c ../../BSW/SecMod\actBNReduce.c(117): WARNING C5909: Assignment in condition Compiling file: ../../BSW/SecMod/actBigNum.c ../../BSW/SecMod\actBigNum.c(234): WARNING C5909: Assignment in condition	
When does this happen: -----	
The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen: -----	
in all configurations	
Resolution Description:	
Workaround: -----	
No workaround available.	
Resolution: -----	
The described issue is corrected by modification of all affected work-products.	

ESCAN00090161 Compiler warning: condition evaluates always to true/false

Component@Subcomponent: Ccl_Asr4ComMCfg5@Implementation

First affected version: 7.00.01

Fixed in versions:

Problem Description:

What happens (symptoms):

Compiler warns for conditional expression being constant

a) in the function ComM_Init() when checking the generated data. Compiler warns about condition being always false in the following conditions:

if (ComM_GetWakeupStateOfChannel(ComM_ChannelIndex) >= COMM_MAX_NUMBER_OF_STATES)

if (ComM_GetSizeOfChannel() != ComM_GetSizeOfChannelPb())

if (ComM_GetSizeOfPnc() != ComM_GetSizeOfPncPb())

As secondary effect compiler might warn about unreachable code/statement.

b) in the function ComM_PncProcessRxSignalEra() compiler warns about condition being always true in

if(ComM_IsSynchronizedOfPnc(pncIndex))

c) in the functions ComM_PncSetBitInSignal() and ComM_PncClearBitInSignal() when checking the generated data. Compiler warns about condition being always true in

if(signalByteIndex < ComM_GetSizeOfPncSignalValues())

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

a) occurs when COMM_DEV_ERROR_DETECT == STD_ON

b) occurs when

- 'Pnc Support' is enabled in ComM (/MICROSAR/ComM/ComMGeneral/ComMPncSupport)

AND

- 'Pnc Gateway Enabled' is enabled in ComM (/MICROSAR/ComM/ComMGeneral/ComMPncGatewayEnabled)

AND

- Only one PNC exists (COMM_ACTIVE_PNC == 1U, can be found in ComM_Cfg.h).

c) occurs when 'Pnc Support' is enabled in ComM (/MICROSAR/ComM/ComMGeneral/ComMPncSupport)

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed because no simple remedy exist.

The warning is caused by an if-statement applied on external configuration data. Configuration data is const for the given compilation context but might be changed at post-build time.

Resolution Description:

ESCAN00090806 Compiler warning: C4310: cast truncates constant value	
Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	7.00.00
Fixed in versions:	11.00.00
Problem Description:	
What happens (symptoms):	

Compiler warns for cast truncates constant value	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

If the uint8_least is of type unsigned char	
The Platform_Types.h contains the following define	
typedef unsigned char uint8_least; /* At least 8 bit */	
Hint:	

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed due to ensure that the init value is large enough. A cast to a smaller value is acceptable and has no impact on the application.	
#define PDUR_INVALID_VARARRAYIDX ((uint16)0xFFFF) is cast for unsigned char to 0xFF which is correct.	
Resolution Description:	
Workaround:	

ignore the warning	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00090831 Compiler warning: integer conversion resulted in a change of sign	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

Compiler warns that "integer conversion resulted in a change of sign".	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

If the compiler WindRiver Diab is used. (found with version 5.9.4.2.)	
Hint:	

The compiler warning is known and has been analyzed thoroughly for its impact on the code.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00091295 Compiler warning: dead assignment / variable set but not used**Component@Subcomponent:** Ccl_Asr4ComMCfg5@Implementation**First affected version:** 5.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

Compiler warns about an useless assignment to a local variable. Typically the warnings refer to local variables 'channel', 'errorId', 'Status' or 'User'.

Example compiler warning strings:

"Useless assignment to variable 'abc'. Assigned value not used."

"Removed dead assignment"

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

EcuC Parameter 'Dummy Statement Kind' is set to 'SelfAssignment'. This can be detected in ComM_Cfg.c: #define COMM_DUMMY_STATEMENT(v) (v)=(v)

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed because no simple remedy exist.

If Dummy Statement is switched off, other compiler warnings might occur e.g. "Unused/unreferenced variable".

Resolution Description:

ESCAN00091340 Compiler warning: cast truncates constant value	
Component@Subcomponent:	If_AsrIfCan@Implementation
First affected version:	5.00.00
Fixed in versions:	6.17.00
Problem Description:	
What happens (symptoms):	

Compile warning occurs.	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

If partial network wakeup PDU filtering is active. (canifcfg.h: CANIF_PN_WU_TX_PDU_FILTER == STD_ON)	
Resolution Description:	
Workaround:	

No workaround available. Issue is checked and not critical.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00091343 Compiler warning: warning C4310: cast truncates constant value	
Component@Subcomponent:	If_AsrIfCan@Implementation
First affected version:	6.09.00
Fixed in versions:	6.17.00
Problem Description:	
What happens (symptoms):	

Compile warning occurs.	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

If transmit buffer is configured as FIFO and cancel API is supported. (canifcfg.h: CANIF_TRANSMIT_BUFFER_FIFO == STD_ON && CANIF_CANCEL_SUPPORT_API == STD_ON)	
Resolution Description:	
Workaround:	

No workaround available. Warning was checked, not critical.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00091547 Compiler warning: condition is always false	
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	11.01.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

Compiler warns for "condition is always true/false"	
Some compiler will also warn because of dead code, resulting from the constant condition	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Dem/DemGeneral/DemSafeBswModeEnabled == TRUE	
AND	
EcuC/EcucGeneral/EcuCSafeBswChecks == TRUE or undefined	
Depending on the configuration, optimization can change a configuration table into a constant macro.	
This can causes some run-time checks to check for equality of two constants.	
Resolution Description:	
Workaround:	

The warning can be ignored.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00092315		Compiler warning: function "CanLL_WakeUpHandling" was declared but never referenced
Component@Subcomponent:	DrvCan_Mpc5700McanLI@Implementation	
First affected version:	2.00.00	
Fixed in versions:	2.10.00	
Problem Description:		
What happens (symptoms):		

Compiler warning occurs: "function "CanLL_WakeUpHandling" was declared but never referenced"		
When does this happen:		

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.		
In which configuration does this happen:		

If "Sleep /Wake-up Functionality" is activated in the configuration (leading to the definition of C_ENABLE_SLEEP_WAKEUP).		
Resolution Description:		
Workaround:		

No workaround available.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00092713 Preprocessor parse error	
Component@Subcomponent:	DrvCan_Mpc5700McanLI@Implementation
First affected version:	2.07.00
Fixed in versions:	2.10.00
Problem Description:	
What happens (symptoms):	

Preprocessor stops with parsing error.	
When does this happen:	

At compilation time.	
In which configuration does this happen:	

Only for CANbedded	
AND	
Range Filtering is used (C_ENABLE_RANGE_x is defined).	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00094232 Compiler warning: "conditional expression is constant"**Component@Subcomponent:** Nm_Asr4NmCan@Implementation**First affected version:** 6.03.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

cannm.c(2649): warning C4127: conditional expression is constant

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

- CanNm is only active on one ComM (System Channel)
AND
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmApiOptimization' is ON
AND
- '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmDevErrorDetect' is ON

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed due to AN-ISC-8-1184_Compiler_Warnings.pdf**Resolution Description:**

Workaround:

No workaround available.

ESCAN00095299 Compiler warning: Mismatch between signature of function declarations and definitions	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	7.02.00
Fixed in versions:	8.03.00
Problem Description:	
What happens (symptoms):	

Compiler warns that a static declaration of a function follows a non-static declaration, for example	
DCM_LOCAL_INLINE FUNC(Std_ReturnType, DCM_CODE) Dcm_Svc2EExecuteOp(...);	
DCM_LOCAL FUNC(Std_ReturnType, DCM_CODE) Dcm_Svc2EExecuteOp(...) {...}	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

- Service 0x2E is supported (in Dcm_Cfg.h: DCM_SVC_2E_SUPPORT_ENABLED == STD_ON)	
Resolution Description:	
Workaround:	

Ignore compiler warning.	
Resolution:	

Modify definition of function to match its declaration.	

ESCAN00095459 Compiler warning: Crc.c: constant out of range**Component@Subcomponent:** SysService_AsrCrc@Implementation**First affected version:** 5.04.00**Fixed in versions:** 5.04.01**Problem Description:**What happens (symptoms):

Typical compiler warnings explanations may be:

Crc.c: constant out of range

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

The compiler warning is only relevant if CRC64 routine is used.

In which configuration does this happen:

Only relevant if CRC64 algorithm is used and

if implementation of used compiler does not adhere to C99 standard: 0xFFFFFFFFFFFFFFFFuL is interpreted as 32bit literal by compiler due to incorrect suffix.

According to C99 standard (ISO/IEC 9899:1999 Ch. 6.4.4.1) the suffix should not matter and the literal should be interpreted as 64bit.

Known compiler, which are affected by this issue and lead to incorrect CRC calculation: Wind River (v5.9.4.4)

Resolution Description:Workaround:

In Crc.c

#define CRC_FINAL_XOR_CRC64 (0xFFFFFFFFFFFFFFFFuL)

should be changed to

#define CRC_FINAL_XOR_CRC64 (0xFFFFFFFFFFFFFFFFuLL)

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095486 Compiler warning: Function "Dcm_Svc14_XX_RepeaterProxySelectDTC" was declared but never referenced**Component@Subcomponent:** Diag_Asr4Dcm@Implementation**First affected version:** 7.02.00**Fixed in versions:** 8.03.00**Problem Description:**

What happens (symptoms):

GreenHills embedded compiler warns that functions are declared but never referenced:

Dcm_Svc14_XX_RepeaterProxySelectDTC
Dcm_Svc14_XX_RepeaterProxyCheckSelectionResult
Dcm_Svc14_XX_RepeaterProxySelectDTC
Dcm_Svc14_XX_RepeaterProxyCheckSelectionResult

When does this happen:

The warning is issued by the GreenHills embedded compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

- Service 0x14 is supported (in Dcm_Cfg.h: DCM_SVC_14_SUPPORT_ENABLED == STD_ON)

AND

- DCM is configured to use DEM API versions other than 4.3.0 (in Dcm_Cfg.h: DCM_DEM_API_430_ENABLED == STD_OFF)

Resolution Description:

Workaround:

Ignore compiler warning.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00095488 Datatype conversion for description routing, possible loss of Data	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	11.00.00
Fixed in versions:	13.03.00
Problem Description:	
What happens (symptoms):	

The Compiler warns for possible loss of data caused by DataType conversion	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

In configurations with description routing configured	
Hint:	

The compiler warning is known and has been analyzed thoroughly for its impact on the code.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095508 Compiler warning: Function "Dcm_MemMgrWriteMemory" was declared but never referenced	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	7.02.00
Fixed in versions:	8.03.00
Problem Description:	
What happens (symptoms):	

GreenHills embedded compiler warns that function Dcm_MemMgrWriteMemory is declared but never referenced.	

The warning is issued by the GreenHills embedded compiler during compilation of the code in case the configuration is as described below.	

In which configuration does this happen:	

- Service 0x3D is not supported (in Dcm_Cfg.h: DCM_SVC_3D_SUPPORT_ENABLED == STD_OFF) AND - Direct memory access is supported (in Dcm_Cfg.h: DCM_MEMMGR_SUPPORT_ENABLED == STD_ON)	
Resolution Description:	
Workaround:	

Ignore warning.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00095513		Compiler warning: Function "Dcm_MemMgrReadMemory" was declared but never referenced
Component@Subcomponent:	Diag_Asr4Dcm@Implementation	
First affected version:	7.02.00	
Fixed in versions:	8.03.00	
Problem Description:		
What happens (symptoms):		

GreenHills embedded compiler warns that function Dcm_MemMgrReadMemory is declared but never referenced.		

The warning is issued by the GreenHills embedded compiler during compilation of the code in case the configuration is as described below.		

In which configuration does this happen:		

- Service 0x23 and 0x2C 0x02 are not supported (in Dcm_Cfg.h: DCM_SVC_23_SUPPORT_ENABLED == STD_OFF && DCM_SVC_2C_02_SUPPORT_ENABLED == STD_OFF) AND		
- Direct memory access is supported (in Dcm_Cfg.h: DCM_MEMMGR_SUPPORT_ENABLED == STD_ON)		
Resolution Description:		
Workaround:		

Ignore warning.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00095530 Compiler warning: variable "IRecStoredIndex" was set but never used

Component@Subcomponent: Diag_Asr4Dem@Implementation

First affected version: 11.00.00

Fixed in versions: 12.01.04, 13.05.00

Problem Description:

What happens (symptoms):

 Compiler specific warning message: variable IRecStoredIndex is not used
 This specific warning is caused by the unused variable 'IRecStoredIndex' in functions
 'Dem_Dcm_CopyERec()' and 'Dem_Dcm_CopyERecs()'

When does this happen:

 The warning is issued by the compiler during compilation of the code in case the configuration is
 as described below.

In which configuration does this happen:

 All containers Dem/DemGeneral/DemExtendedDataRecordClass only reference
 Dem/DemGeneral/DemDataClass which have parameter Dem/DemGeneral/DemDataClass/
 DemDataElementInternalData set

AND

At least on Dem/DemConfigSet/DemEventParameter references a Dem/DemGeneral/
 DemExtendedDataRecordClass

AND

The compiler used emits this specific warning.

Hint: in affected configurations, DEM_CFG_SUPPORT_USER_ERECs is generated to STD_OFF

Resolution Description:

Workaround:

 The warning can be ignored.

Resolution:

 The described issue is corrected by modification of all affected work-products.

ESCAN00095542		Compiler warning: 'PduR_RmIf_SingleBufferHandling' declared 'static' but never defined
Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation	
First affected version:	9.00.00	
Fixed in versions:	10.00.00	
Problem Description:		
What happens (symptoms):		

Compiler warns for 'PduR_RmIf_SingleBufferHandling' declared 'static' but never defined.		
When does this happen:		

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.		
In which configuration does this happen:		

If a Fifo queued communication interface routing is configured but no single buffer routing exists.		
- any routing with		
/MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/ PduRDestPduQueueDepth > 1		
AND no routing with		
/MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/ PduRDestPduQueueDepth == 1		
Resolution Description:		
Workaround:		

ignore the warning. Only a unnecessary function declaration.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00095907 Compiler warning: Com.c(4142): warning C4100: 'idxFilterInfo' : unreferenced formal parameter

Component@Subcomponent: Il_AsrComCfg5@Implementation

First affected version: 2.01.00

Fixed in versions: 13.03.01

Problem Description:

What happens (symptoms):

Compiler warns for an unused define which is used in a special configuration: Can be accepted

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

In configurations with comstackLib optimizations enabled:

```
/MICROSAR/Com/ComGeneration/ComBoolDataInArrayOfStructStrategy : TRUE
/MICROSAR/Com/ComGeneration/ComDataDeduplicationStrategy: WITH_CAST
/MICROSAR/Com/ComGeneration/ComDeduplicateIndirectedData : Enabled
/MICROSAR/Com/ComGeneration/ComOutOfBoundsWriteProtectionStrategy : NONE
/MICROSAR/Com/ComGeneration/ComReduceBoolDataByNumericalOperandStrategy:
WITH_ANY_VALUE
/MICROSAR/Com/ComGeneration/ComReduceConstantData2Define : Enabled
```

All comstackLib optimizations which are not mentioned above have following configuration:
Thresholds are set to 0 and all other comstackLib checks are disabled.

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code.

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096038 Compiler warning: Passing of incompatible pointers**Component@Subcomponent:** Rte_Core@Implementation**First affected version:** 1.00.00**Fixed in versions:** 1.16.00**Problem Description:**

What happens (symptoms):

The compiler warns that an incompatible pointer is passed to a server runnable.

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

This happens when the configuration contains a client-server connection and when the client operation only uses a primitive out parameter whereas the server operation uses an array.**Resolution Description:**

Workaround:

Use the same data types on both sides of the client-server connection.

Resolution:

The described issue is corrected by modification of all affected work-products.**ESCAN00096133 Compiler warning: warning C4310: cast truncates constant value****Component@Subcomponent:** Cp_Asr4Xcp@Implementation**First affected version:** 1.00.00**Fixed in versions:** 1.00.01, 2.01.00**Problem Description:**

What happens (symptoms):

The following compiler warning is thrown:

"..\..\..\external\BSW\Xcp\Xcp.c(5699): warning C4310: cast truncates constant value"

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

In all configurations where XCP_GET_SESSION_STATUS_API is enabled.**Resolution Description:**

Workaround:

No workaround available beside disabling XCP_GET_SESSION_STATUS_API.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00096246 Compiler warning: C4100: 'CanId' : unreferenced formal parameter	
Component@Subcomponent:	If_AsrIfCan@Implementation
First affected version:	6.13.00
Fixed in versions:	6.14.01
Problem Description:	
What happens (symptoms):	

Compiler warning: C4100: 'CanId' : unreferenced formal parameter -> Affected API: CanIf_HIIndicationSubULCall()	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

- CANIF_SUPPORT_NMOSEK_INDICATION == STD_OFF [see file: CanIf_Cfg.h] and - CANIF_META_DATA_RX_SUPPORT == STD_ON [see file: CanIf_Cfg.h]	
Resolution Description:	
Workaround:	

The compiler warning is not critical and hence please ignore it.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096376 Compiler warning: Unreachable code in ESLib_EdDSA.c	
Component@Subcomponent:	SysService_CryptoCv@Impl_ESLib
First affected version:	2.05.00
Fixed in versions:	2.06.02
Problem Description:	
What happens (symptoms):	

The code of ESLib_EdDSA.c has statements which are not reachable. This can lead to an compiler warning on some compilers.	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Elliptic Curve Digital Signature Algorithm is used	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096911 Compiler warning: type qualifier is meaningless on cast type	
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	3.00.00, 1.00.01
Problem Description:	
What happens (symptoms):	

The compiler throws several warnings of the type: "../Core/BSW/Xcp/Xcp.c", line 4844: warning #191-D: type qualifier is meaningless on cast type daqNumber = Xcp_GetVal16(&CmdPtr[XCP_CRO_ALLOC_ODT_ENTRY_DAQ]);	
This warning warns about an unnecessary cast and can be ignored.	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

all configurations	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00096913		Compiler warning: Static function 'Rte_QUnqueueElementCallbackSpecific<OsApplication>' is not used within this translation unit
Component@Subcomponent:	Rte_Core@Implementation	
First affected version:	1.04.00	
Fixed in versions:	1.17.00	
Problem Description:		
What happens (symptoms):		

Compiler warns for an unused function.		
When does this happen:		

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.		
In which configuration does this happen:		

On a partitioned or multicore system with fanin queued communication at least on one partition/core and no queued communication on another partition/core.		
Resolution Description:		
Workaround:		

No workaround available.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00097289	Compiler warning: integer literal is too large to be represented in a signed integer type, interpreting as unsigned
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.05.00
Fixed in versions:	1.17.00
Problem Description:	
What happens (symptoms):	

Compiler issues a warning: integer literal is too large to be represented in a signed integer type, interpreting as unsigned when a lower limit define from the RTE is used.	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

This happens when the configuration contains application data type that are mapped to signed 64-bit integer types and when the lower limit is configured to -9223372036854775808.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097375 Compiler warning: 'ITxHdl' : local variable is initialized but not referenced	
Component@Subcomponent:	Tp_Asr4TpCan@Implementation
First affected version:	3.02.00
Fixed in versions:	3.03.02
Problem Description:	
What happens (symptoms): ----- The following warning is issued during compilation: 'ITxHdl' : local variable is initialized but not referenced When does this happen: ----- The warning is issued by the compiler during compilation of the code in case the configuration is as described below. In which configuration does this happen: ----- This warning is issued in configurations where a single Tx SDU exists (i.e., the macro CANTP_NUM_TX_SDUS is defined as 1).	
Resolution Description:	
Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097692 Compiler warning: conversion from 'int' to 'Dcm_CfgNetBufferSizeOptType'	
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	4.01.00
Fixed in versions:	9.02.00
Problem Description: What happens (symptoms): ----- Compiler warns for possible loss of data: Check if cast is missing and if there is really a data loss due to an implicit/explicit cast on the target platform When does this happen: ----- The warning is issued by the compiler during compilation of the code in case the configuration is as described below. In which configuration does this happen: ----- - Service 0x22 is supported and handled by DCM (in Dcm_Cfg.h: #define DCM_SVC_22_SUPPORT_ENABLED == STD_ON) AND - DCM is configured to support DIDs with multiple signals (in Dcm_Cfg.h: #define DCM_DIDMGR_MULTISIGNAL_ENABLED == STD_ON) AND - At least one DID supports read operation (in Dcm_Cfg.h: #define DCM_DIDMGR_OPTYPE_READ_ENABLED == STD_ON)	
Resolution Description: Workaround: ----- Ignore the warnings, since there is no danger of data value truncation. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00097790 Compiler warning: 'CanTpTxSduId' : unreferenced formal parameter	
Component@Subcomponent:	Tp_Asr4TpCan@Implementation
First affected version:	3.02.00
Fixed in versions:	3.03.02
Problem Description:	
What happens (symptoms):	

The following warning is issued during compilation: 'CanTpTxSduId' : unreferenced formal parameter	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

The warning is issued in configurations fulfilling all of the following conditions:	
1. The macro CANTP_DEV_ERROR_DETECT has been defined as STD_OFF.	
2. The macro CANTP_CONFIGURATION_VARIANT has been defined as	
CANTP_CONFIGURATION_VARIANT_PRECOMPILE.	
3. The macro CANTP_NUM_TX_SDUS has been defined as 1.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00097980 Compiler warning: Unreferenced formal parameter due to reduction of rom data	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	1.00.00
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

Compiler warning occurs: Unreferenced formal parameter	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

/MICROSAR/Com/ComGeneration/ComReduceConstantData2Define is enabled	
OR	
/MICROSAR/Com/ComGeneral/ComOptimizeConstArrays2Define is enabled (in older releases).	
Hint:	

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed due to existence of a sufficient workaround.	
Resolution Description:	
Workaround:	

Disable	
/MICROSAR/Com/ComGeneration/ComReduceConstantData2Define (in newer releases)	
OR	
/MICROSAR/Com/ComGeneral/ComOptimizeConstArrays2Define (in older releases).	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098038 Compiler warning: number of arguments don't match	
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.13.00
Fixed in versions:	1.18.00
Problem Description:	
What happens (symptoms):	

Compiler warns for a mismatch in number of arguments.	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Two port prototypes use the same port interface AND the transformer error handling flags differ for one data element prototype AND the generated API is Rte_Write, Rte_Send, Rte_Invalidate, Rte_Read, Rte_DRead or Rte_Receive AND the component needs a port data structure.	
Resolution Description:	
Workaround:	

Ensure that the error handling flags are the same for all interfaces in a port data structure.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098070 Compiler warning: NvM_Cfg.c: 'ServiceId' : unreferenced formal parameter	
Component@Subcomponent:	MemService_AsrNvM@GenTool_GeneratorMsr
First affected version:	3.01.02
Fixed in versions:	
Problem Description:	
What happens (symptoms):	

1> NvM_Cfg.c	
1>..\..\Appl\GenDataVtt\NvM_Cfg.c(588): warning C4100: 'ServiceId' : unreferenced formal parameter	
with Visual Studio compiler	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Any configuration with disabled NvMMultiBlockCallback and NvMBswMMultiBlockJobStatusInformation	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00098195 Compiler warning: possible loss of data when least types are larger than the platform types

Component@Subcomponent: Rte_Core@Implementation

First affected version: 1.11.00

Fixed in versions: 1.18.00

Problem Description:

What happens (symptoms):

Compiler warns that casting <dt>_least to <dt> may result in a loss of data.

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

This happens when the configuration contains variable length arrays.

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099190 Compiler warning: BswM_Lcfg.c(2990): warning C4100: 'handleId' : unreferenced formal parameter

Component@Subcomponent: SysService_Asr4BswMCfg5@GenTool_GeneratorMsr

First affected version: 6.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Compiler warns about C4100: 'handleId' : unreferenced formal parameter in BswM_Lcfg.c.

When does this happen:

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:

In all configurations which use actions of type BswMTimerControl.

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00099286		Compiler warning: unused parameter 'ipduGroupVector' and 'initialize' in function 'Com_IpduGroupControl'
Component@Subcomponent:	Il_AsrComCfg5@Implementation	
First affected version:	1.00.00	
Fixed in versions:	15.00.00	
Problem Description:		
What happens (symptoms):		

Compiler warns for an unused argument: Can be accepted - is usually because of a standardized API which doesn't respect unused parameters in special configurations		
When does this happen:		

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.		
In which configuration does this happen:		

Any configuration where COM_ACTIVATABLERXCOMIPDUS and COM_ACTIVATABLETXCOMIPDUS is defined to STD_OFF.		
Resolution Description:		
Workaround:		

No workaround available.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00099287		Compiler warning: unused parameter 'deferredfctPtrCacheStrctPtr' in function 'Com_RxProcessDeferredPDU'
Component@Subcomponent:	II_AsrComCfg5@Implementation	
First affected version:	10.00.00	
Fixed in versions:	15.00.00	
Problem Description:		
What happens (symptoms):		

Compiler warns for an unused argument: Can be accepted - is usually because of a standardized API which doesn't respect unused parameters in special configurations		
When does this happen:		

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.		
In which configuration does this happen:		

Any configuration where COM_EXISTS_DEFERRED_SIGNALPROCESSINGOFRXPDUINFO is defined to STD_ON AND COM_COM_RXSIGINFOENDIDXOFRXPDUINFO is defined to STD_OFF AND COM_RXSIGGRPINFOINDENDIDXOFRXPDUINFO is defined to STD_OFF.		
Resolution Description:		
Workaround:		

No workaround available.		
Resolution:		

The described issue is corrected by modification of all affected work-products.		

ESCAN00099291 Compiler warning: unused parameter 'PduInfoPtr' in function 'Com_RxSignalProcessing'	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	12.00.00
Fixed in versions:	15.00.00
Problem Description:	
What happens (symptoms):	

Compiler warns for an unused argument: Can be accepted - is usually because of a standardized API which doesn't respect unused parameters in special configurations	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

Any configuration where	
/MICROSAR/Com/ComConfig/ComSignal/ComBitSize is set to zero for all configured rx signals.	
Resolution Description:	
Workaround:	

No workaround available.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00099292 Compiler warning: unused parameter 'idxTxSigInfo' in function 'Com_SendSignal_WriteSignal'	
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	9.00.00
Fixed in versions:	15.00.00
Problem Description:	
What happens (symptoms): ----- Compiler warns for an unused argument: Can be accepted - is usually because of a standardized API which doesn't respect unused parameters in special configurations	
When does this happen: ----- The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen: ----- Any configuration where	
1) /MICROSAR/Com/ComConfig/ComSignal/ComBitSize is set to zero for all signals 2) No signalGroups are configured	
Resolution Description:	
Workaround: ----- No workaround available.	
Resolution: ----- The described issue is corrected by modification of all affected work-products.	

ESCAN00099981 Compiler warning: the order of volatile accesses is undefined**Component@Subcomponent:** Cp_Asr4Xcp@Implementation**First affected version:** 1.00.00**Fixed in versions:****Problem Description:**

What happens (symptoms):

The compiler issues the following warning:

Xcp.c:

if (((Xcp_ChannelInfo[XCP_CHANNEL_IDX].SendStatus & (uint8)XCP_SEND_PENDING) == 0u)
^

"Xcp\Xcp.c",2166 Warning[Pa082]: undefined behavior: the order of volatile accesses is undefined in this statement

(void)Xcp_CallTIFunction_3_Param(XCP_CHANNEL_IDX,
^

"Xcp\Xcp.c",2174 Warning[Pa082]: undefined behavior: the order of volatile accesses is undefined in this statement

There is an access to two volatile variables in theses statements. The order of access is not guaranteed.

The variables are RAM only variables and not dependent on each other, so access can happen independently.

Therefore this warning does not cause an issue and can be accepted.

When does this happen:

The warning is issued by the compiler during compilation of the code.

The used compiler information is:

Version: IAR ANSI C/C++ Compiler V8.20.1.14183/W32 for ARM

VectorDefaultOptions: --cpu Cortex-M4 --fpu=none --cpu_mode=thumb --endian=little -On --debug --header_context -e

VectorBuildEnvironmentOptions: -DBRS_DERIVATIVE_S32K144 -DBRS_OSC_CLK=8 -DBRS_TIMEBASE_CLOCK=80 -DBRS_OS_USECASE_BRS -DBRS_EVA_BOARD_HSR_706 -DBRS_PROGRAM_CODE_LOCATION_FLASH -DBRS_VECTOR_TABLE_LOCATION_FLASH -DBRS_CPU_CORE_CORTEX_M4 -DBRS_STACK_SIZE=0x2500 -DBRS_PLATFORM_ARM -DBRS_COMP_IAR -DBRS_INSTRUCTION_SET_THUMB -lc lst/.lst -oobj\o

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.

ESCAN00100176 Compiler warning: cast truncates constant value	
Component@Subcomponent:	If_AsrIfCan@Implementation
First affected version:	5.00.00
Fixed in versions:	6.17.00
Problem Description:	
What happens (symptoms):	

The compiler warns for a truncation of a constant value due to cast.	
When does this happen:	

The warning is issued by the compiler during compilation of the code in case the configuration is as described below.	
In which configuration does this happen:	

CANIF_WAKEUP_VALIDATION is enabled AND CANIF_WAKEUP_VALID_ALL_RX_MSGS is disabled AND CANIF_WAKEUP_VALID_ONLY_NM_RX_MSGS is enabled	
Resolution Description:	
Workaround:	

No workaround available. Warning was checked, not critical.	
Resolution:	

The described issue is corrected by modification of all affected work-products.	

ESCAN00100182 Compiler warning: A value that cannot be used to initialize an entity with a function pointer type	
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	
Problem Description: What happens (symptoms): ----- Compiler warns for a value that cannot be used to initialize an entity with a function pointer type with MSN_CopyRxData, MSN_CopyTxData and MSN_StartOfReception. When does this happen: ----- The warning is issued by the compiler during compilation of the code in case the configuration is as described below. In which configuration does this happen: ----- Any configuration using a BSW or CDD adjacent to the PduR which has been implemented based on an AUTOSAR Specification that uses const in API parameters of MSN_CopyRxData, MSN_CopyTxData and MSN_StartOfReception. The AUTOSAR Specifications have been changed multiple times between ASR 4.00.03 and today. There is no common solution for all versions possible and you have to accept the compile warning until all components in the system have implemented the Com-Stack API harmonization introduced with ASR 4.03.00.	
Resolution Description: Workaround: ----- No workaround available. Resolution: ----- The described issue is corrected by modification of all affected work-products.	

3. New Issues for Information

Issues which should not have an effect on the usage of the license as the issues are relevant for use cases other than those defined in the questionnaire. The list contains issues that have been detected since the last report.

Issues listed in this section are not relevant for the use case that has been documented in the questionnaire provided to Vector. However, the issues may be relevant for other use cases. Also issues that have been accepted or are tolerated by the OEM (as defined in the questionnaire) are reported here.

No issue to be reported.

4. Report Legend

Issue Report	
Report Creation Date 2011-02-25	
Index	The ID number identifies the Issue
ESCAN0002257 Headline describes symptoms and consequences of the Issue in one sentence DrvCan baseAsr@GenTool_GeneratorGeny	
ESCAN0002257 Headline describes symptoms and consequences of the Issue in one sentence	
Component@Subcomponent: First affected version: Version fixed: Problem Description: What happens (symptoms):	Component@Subcomponent describes the group of workproducts which are composed of the source code, project documentation, User Manual and Generation Tool. The Subcomponent describes the certain affected work-product in which part of the Component the issue appears. e.g. inside of the source code (e.g. Implementation) or inside of the User Manual (e.g. Documentation) or inside of the concerning Generation Tool code.
<p>// to be removed: Describe FROM CUSTOMERS NON TECHNICAL POINT OF VIEW, - which symptoms one will get if this issue occurs? - How can the issue be seen? - if it cannot be seen, how can the customer detect it? - what happens AFTER the issue occurred? - What is the consequence, the implication?</p> <p>Consider the following questions: If the issue is TEMPORARY: Does the issue cause the malfunction once but after that ECU continues to work and probably works correctly? In which situation (ECU reset / wakeup) does the ECU recover? If the issue is PERMANENT: ECU is blocked until Watch-Dog reset. ECU blocked forever and Watch-Dog cannot help.</p> <p>When does this happen:</p> <p>// to be removed: Describe FROM CUSTOMERS NON TECHNICAL POINT OF VIEW, which circumstances, operational situations, API function calls lead to the issue. With this information the customer wants to find out, whether he is affected by this issue or not.</p> <p>Consider the following questions: When (during runtime) does the issue occur and how can the customer find the issue? (1) Always and immediately (2) Only under specific circumstances (describe them) (3) Rarely, very rarely or unlikely Can the probability of occurrence of the issue be estimated?</p> <p>In which configuration does this happen:</p> <p>// to be removed: Describe FROM CUSTOMERS POINT OF VIEW, which configurations of e.g. GenTool, database (attributes), OEM, compiler, components, ... lead to the issue.</p> <p>Resolution Description: Workaround: No workaround available.</p> <p>// to be removed: If there is a workaround available, please replace the default text. Describe FROM CUSTOMERS POINT OF VIEW, what has to be done to avoid this issue.</p> <p>Resolution: The described issue is corrected by modification of all affected workproducts.</p> <p>// to be removed: technical resolution: e.g. error is resolved in file "xyz" function "opq"</p>	<p>The First affected Version describes in which version of the Component the Issue appears first and the Version fixed describes the corrected version of the Component in which the Issue does not appear anymore.</p> <p>The Problem description expresses the Issue content, eventually impact, etc. What happens: Symptoms, consequences and/or the detection way is described. When does it happen: Ignition, trigger point of the Issue In which configuration does this happen: Dependencies to a certain functionality or another component</p> <p>The Resolution description describes a workaround, if available and the resolution of the Issue.</p>

5. 3rd Party Software Issues

This issue report does not include issues of 3rd party software. If 3rd party software was included in the SIP, the documentation of the issue reporting process is included in the SIP: .\Doc\DeliveryInformation\IssueHandling_<Name>.pdf. Please follow the given instructions.

6. Quality Management Contact

Quality Management
Productline Embedded Software (PES)

Vector Informatik GmbH
Ingersheimer Str. 24
D-70499 Stuttgart

Phone: +49 711 80670-3700
Fax: +49 711 80670-399
eMail: QualityPES@vector.com