Issue Report

VECTOR

License Number	Customer
CBD1900078	Aptiv (China) Technology Company Ltd. Package: MSR_Vector_SLP4 - MSR Brand license "ASDM" for the brand Trumpchi of GAC

Maintenance Expiry Date

2020-05-01

SIP Version

21.06.27

SLP	Delivery Number
MSR_Vector_SLP4	D01

Report Creation Date

2019-05-07

Scope

This issue report applies to the License Number CBD1900078 and Delivery Number D01 only. The issue report does not apply to earlier or later deliveries.

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

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

Index

ESCAN00096622 Tasking compiler option "--section-name-with-symbol" is not applicable

Component@Subcomponent: Os_PlatformTriCoreAurixGen7@Implementation
First affected version: 1.00.00
Fixed in versions:

Problem Description:
What happens (symptoms):

If the compiler option --section-name-with-symbol is used the MPU may raise memory protection errors upon stack operations.

When does this happen:

After the MPU has been initialized by the OS any stack operation leads to a memory protection violation.

In which configuration does this happen:

SC3 or SC4

Resolution Description:
Workaround:

--section-name-with-symbol must not be used.
generated linker scripts have to be adapted after generation.

Resolution:

This issue cannot be resolved

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.

ESCAN00100968 MICROSAR OS does support only 1023 different task priorities.

Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	1.00.00
Fixed in versions:	2.30.00

Problem Description:

What happens (symptoms):

Task scheduling does not work as expected.
Tasks with a calculated priority higher then 1023 are not scheduled correctly.

When does this happen:

This happens always during system runtime.

In which configuration does this happen:

This happens in configurations where more then 1023 tasks are configured with different priorities.
(/MICROSAR/Os/OsTask/OsTaskPriority)

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101216	Unconnected NV PR port with initial value triggers the protection hook
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.18.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

The OS enters the protection hook.

When does this happen:

During Rte_Start.
Always and immediate.

In which configuration does this happen:

This happens when the configuration contains unconnected PR NV ports with initial value that are mapped to a different partition than the NVM and when the BSW runs in a nontrusted partition.

Resolution Description:

Workaround:

Connect the PR port to a NV block SWC.

Resolution:

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

ESCAN00101305 Reception via LdCom TP fails

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

If Rte_LdComCbkStartOfReception is called with SduLength=0, the reception buffer is not locked although BUFREQ_OK is returned.

If another SDU is to be received, the corresponding Rte_LdComCbkCopyRxData function returns BUFREQ_E_NOT_OK.

Thus the reception via LdCom TP fails.

When does this happen:

During runtime whenever data is received via LdCom TP callbacks.

In which configuration does this happen:

This happens when the configuration uses the LdCom TP API.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101439 RTE generator does not trigger init runnables

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.05.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

The RTE does not trigger init runnables.

When does this happen:

During runtime. When the RTE is started.

In which configuration does this happen:

This happens when the configuration contains a basic task with multiple entities that are triggered with different cycle times and an init runnable.

Resolution Description:

Workaround:

Move the init runnables to a separate task.

Resolution:

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

ESCAN00101544 Runnable may not be executed or may be executed with delay

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

A runnable may not be executed or may be executed with delay.

When does this happen:

During runtime.

In which configuration does this happen:

This happens when the configuration contains

- a multiple instantiated software component with a reentrant runnable "A"
- AND the runnable "A" is assigned to different tasks
- AND one of the tasks must be assigned another runnable "B" which follows "A".
- AND runnable "A" triggers runnable "B", e.g. via an on data reception trigger

OR

- a runnable "A" that triggers a runnable "B"
- AND "B" is also triggered by itself
- AND "B" is mapped to same task as "A"
- AND "B" follows "A"

Then it is possible that runnable "B" is not triggered in time.

Resolution Description:

Workaround:

Change task mapping, e.g. map all instances of the reentrant runnable to same task.

Resolution:

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

ESCAN00101729 Mode machine disables unrelated runnables

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.95, 1.20.00

Problem Description:

What happens (symptoms):

A runnable is triggered in a mode for which a mode disabling is configured
or
a runnable is not triggered in a mode for which no mode disabling is configured

When does this happen:

During runtime when an unrelated mode machine is switched.

In which configuration does this happen:

This happens when all of the following conditions evaluate to true
- a software component contains a runnable entity with a trigger that can be disabled in certain modes
- the trigger is disabled by two different mode r- or pr-ports
- one of the ports is not connected
- one of the ports is connected
- the trigger is disabled in the initial mode of the unconnected port
- there are multiple OS applications
- there are additional unrelated mode disablings

In this case switching the mode machine of the connected port will also disable or enable unrelated runnable entities.

Resolution Description:

Workaround:

Configure an additional runnable entity that is disabled in the same modes as the runnable but is not disabled by the inactive mode machine.

Resolution:

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

ESCAN00101801 Mode machine incorrectly disables runnables during initialization

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

A runnable entity is not triggered after Rte_Start although it is configured
to be active in the initial modes.

When does this happen:

During runtime

In which configuration does this happen:

This happens when the following conditions evaluate to true

- cross partition mode switches are configured
- a trigger is disabled by two mode ports
- one mode port disables the trigger in the initial mode
- the other mode port disables the runnable in a different mode
- another runnable has a mode disabling on the same mode that does not disable the other runnable in the initial mode

Resolution Description:

Workaround:

Provide one global mode machine for the mode disabling of all runnables and update it depending on the other ECU modes.

Resolution:

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

ESCAN00101810	Cross core mode triggered (OnEntry; OnExit; OnTransition) runnable is not executed
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.95, 1.20.00

Problem Description:

What happens (symptoms):

Waitpoint is missing in the mode switch task that initiates the switch.
Mode switch acknowledge will be set immediately, mode switch task will not wait for cross core mode triggered runnables.
Cross core mode triggered runnable may not be executed if Rte_Switch is triggered rapidly.

When does this happen:

During runtime.

In which configuration does this happen:

- cross core mode switches are configured
- runnable with OnEntry, OnExit or OnTransition trigger on a different core than the Rte_Switch
- no mode disablings are configured on the core with the OnEntry, OnExit and OnTransition runnables

Resolution Description:

Workaround:

Create another trigger on the mode triggered runnable that contains mode disablings.

Resolution:

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

ESCAN00102335	**Wrong InitValue Selection Due to duplicate constant name with different values in different packages**
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

Rte_Read, Rte_DRead, Rte_IRead, Rte_CData, Rte_Prm, Rte_IrvRead or Rte_IrvIRead API return the wrong initial value.

When does this happen:

During runtime, when no data was sent previously.

In which configuration does this happen:

This happens when the configuration contains multiple constants in different packages with the same name and different values.

Resolution Description:

Workaround:

Use unique names for the constants.

Resolution:

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

ESCAN00102366 APIs with queues not printed to configuration feedback

Component@Subcomponent:	Rte_Analyzer@Application
First affected version:	0.05.00
Fixed in versions:	1.03.00

Problem Description:

What happens (symptoms):

Configuration feedback does not list APIs with queues in the configuration feedback.
As a result, the user of SafeRTE might miss to apply integration measures that are described in the Safety Manual when the issue is not detected
during an review of the RTE Analyzer report what is required according to SMI-322 in the RTE Safety Manual.
The issue itself does not lead to a wrong generation of the embedded code, however problems in the code might not be detected during integration.

When does this happen:

When RTE Analyzer is not started with the -e option.

In which configuration does this happen:

This happens when the configuration contains APIs with queue e.g. Rte_Call or Rte_Receive in an ASIL partition.

Resolution Description:

Workaround:

Start RTE Analyzer with the option -e to enabled the configuration feedback for all partitions.

Resolution:

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

ESCAN00102393	NvM sets a write once block to write protected though the block was never written
Component@Subcomponent:	MemService_AsrNvM@Implementation
First affected version:	3.01.01
Fixed in versions:	5.09.01

Problem Description:

What happens (symptoms):

Never written block (not existing in NV RAM) is write protected after the start up (NvM_ReadAll).

When does this happen:

Each time the NvM processes the NvM_ReadAll, a write once block does not exist or is defect (not readable) in NV RAM and the underlying device delivers a result other than MEMIF_BLOCK_INVALID.

In which configuration does this happen:

Block configuration:
/MICROSAR/NvM/NvMBlockDescriptor/NvMWriteBlockOnce == true
/MICROSAR/NvM/NvMBlockDescriptor/NvMSelectBlockForReadAll does not matter, because NvM will check the content of a write once block during a ReadAll though the SelectForReadAll is disabled.

Resolution Description:

Workaround:

Workaround possible via configuration "update", if re-flashing is possible:
1. use a configuration where write once is disabled, write data, shutdown
2. use a "new" configuration with write once enabled. Since only the write once flag shall change, the configuration identifier shall not be updated, NvM shall start normally, Fee delivers the data, NvM knows that the write once blocks are available and sets the write protection.

Without re-flashing there is no workaround, the block cannot be written because write protected and therefore it never will be OK!

Resolution:

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

ESCAN00102405	Wrong usage of Os_InitialEnableInterruptSources leads to protection violation.
Component@Subcomponent:	Os_CoreGen7@Doc_TechRef
First affected version:	2.24.00
Fixed in versions:	2.25.00

Problem Description:

What happens (symptoms):

The call of the OS API Os_InitialEnableInterruptSources leads to an protection violation.
The system ends up in protection hook or shutdown.

When does this happen:

This happens during the call of the service Os_InitialEnableInterruptSources.

In which configuration does this happen:

This happens in all configurations with Scalability Class SC3 or SC4
AND
Unprivileged Applications
AND
Task of an unprivileged application is calling the service.

Resolution Description:

Workaround:

The API shall only be called in privileged task context. This is a precondition by design.

Resolution:

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

ESCAN00102633 Wrong compiler abstraction: variable WdgMSupervisedEntity_pt should be const

Component@Subcomponent:	SysService_Asr4WdM@Implementation
First affected version:	5.00.00
Fixed in versions:	5.04.01

Problem Description:

What happens (symptoms):

The variable WdgMSupervisedEntity_pt is a pointer in RAM pointing to constants, but should be a constant pointer accessing constants (it is a P2CONST and should be CONSTP2CONST). This may have the following consequences:

1. The variable might not be mapped correctly via the memory mapping mechanism. This depends on the compiler. If the variable is mapped to the wrong memory section, it is no longer protected by the MPU and memory corruption might be a consequence, leading to undefined behavior.
2. The variable might not be initialized correctly via the StartUp Code, if the StartUp Code is optimized. This would again lead to undefined behavior.

When does this happen:

The 1. consequence may happen depending on the used compiler and architecture. It is recommended to check that the variable WdgMSupervisedEntity_pt is mapped to the correct memory section, by checking the linker file.

The 2. consequence can only happen if not all variables are initialized by the StartUp Code, due to optimization options in the StartUp Code.

In which configuration does this happen:

The variable WdgMSupervisedEntity_pt is only defined in case multicore usage is enabled, that is if # define WDGM_MULTICORE_USAGE STD_ON has been generated to WdgM_Cfg_Features.h

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102672 Wrong compiler abstraction/memory mapping: variable WdgIfStateCombinerSlaveTriggerPattern should be const

Component@Subcomponent:	If_Asr4IfWd@Implementation
First affected version:	5.02.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The variable WdgIfStateCombinerSlaveTriggerPattern (within the struct WdgIf_StateCombinerConfigType) is a pointer in RAM pointing to constants, but should be a constant pointer accessing constants. This may have the following consequences:
1. The variable might not be mapped correctly via the memory mapping mechanism. This depends on the compiler. If the variable is mapped to the wrong memory section, it is no longer protected by the MPU and memory corruption might be a consequence, leading to undefined behavior.
2. The variable might not be initialized correctly via the StartUp Code, if the StartUp Code is optimized. This would again lead to undefined behavior.

When does this happen:

The 1. consequence may happen depending on the used compiler and architecture. It is recommended to check that the variable WdgIfStateCombinerSlaveTriggerPattern is mapped to the correct memory section, by checking the linker file.
The 2. consequence can only happen if not all variables are initialized by the StartUp Code, due to optimization options in the StartUp Code.

In which configuration does this happen:

The variable WdgIfStateCombinerSlaveTriggerPattern is only defined in case state combiner usage is enabled, that is if # define WDGIF_USE_STATECOMBINER STD_ON has been generated to WdgIf_Cfg_Features.h

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102727	Missing validation of parameter PageSize.
Component@Subcomponent:	DrvMem_vMem_30_XXspi01@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Currently there's no validation for the parameter "vMemInstance -> vMemSectorList -> vMemSector -> vMemPageSize".
It is important that the value of this parameter is a multiple of a power of two (e.g. 8, 16, 32, 64 ...).
In any other case, memory corruption may occur.

Several external flash devices connected to the ECU via Spi specify a page range (e.g. from 1 - 256 Bytes).
This page may be written several times as long as only erased memory cells are accessed.

These pages with a size of up to (e.g.) 256 Bytes are also aligned to the value of their size (e.g. 256 Bytes aligned).

In case a write operation exceeds such a page boundary, a wrap around is performed an the write access continuous
at the page's start address.
From the system's point of view:
Different NvM blocks may share the same physical page on an external flash device. A wrap around during a write
access may result in the data corruption of another NvM block.

When does this happen:

If the parameter vMemPageSize is configured incorrectly.

In which configuration does this happen:

See above.

Resolution Description:

Workaround:

The value of the BSWMD parameter vMemPageSize must be configured as a multiple of a power of two (e.g. 8, 16, 32, 64 ...).

Resolution:

Add a validator to the generator project to validate the value of the vMemPageSize parameter.

ESCAN00102760 NvM sets a write once block not selected for ReadAll to NVM_REQ_OK during ReadAll

Component@Subcomponent:	MemService_AsrNvM@Implementation
First affected version:	3.01.01
Fixed in versions:	5.09.01

Problem Description:

What happens (symptoms):

After the NvM_ReadAll the block error status is NVM_REQ_OK for a specific block (see configuration description below), though no data is available and the block is not even marked to be read during NvM_ReadAll.

When does this happen:

During a NvM_ReadAll if a block matches the configuration below.

In which configuration does this happen:

Block configuration:

- NvM/NvMBlockDescriptor/NvMWriteBlockOnce true
- NvM/NvMBlockDescriptor/NvMSelectBlockForReadAll false

Resolution Description:

Workaround:

Configure NvM/NvMBlockDescriptor/NvMSelectBlockForReadAll == true, or NvM/NvMBlockDescriptor/NvMWriteBlockOnce == false.

Otherwise there is no workaround available.

Resolution:

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

ESCAN00102774 Missing memory Gaps between OS Stacks

Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

According to the Core Architecture manual of Infineon (TC1.6.1 and TC1.6.2) a buffer/gap of at least 15 unused bytes is needed between adjacent memory regions which shall be protected by the core MPU.
The generated linker scripts of the OS omits these buffers/gaps. Unnaturally aligned pointer accesses (e.g. 32bit access on an two-byte aligned pointer) may lead to memory overwrites in the adjacent (upper address) linked stack.
The MPU is not capable to detect these overwrites.

When does this happen:

Normal stack operations (as generated by compiler) do not lead to corruption!

Only pointer accesses may lead to memory corruption.
The following prerequisites have to be fulfilled:
1. The pointer points to the uppermost address of the current valid stack (which is logical the begin of the current valid stack).
2. A 32 Bit access or 64 bit write access is performed on that pointer.
3. The start address of the pointer must be within the MPU range which is specified by DPR0_L and DPR0_U (Lower and Upper addresses of the stack protection region)

A pointer to a 32 Bit variable may lead to memory corruption
- If the Pointer points into the current valid stack (uppermost address / DPR0_U - 2) and then perform a 32bit or a 64 bit write access on that pointer.
A pointer to a 64 Bit variable may lead to memory corruption
- If the Pointer points into the current valid stack (uppermost address / DPR0_U - 2 or DPR0_U - 4) and then perform a 32bit or a 64 bit write access on that pointer.

In which configuration does this happen:

If the scalability class is SC3 or SC4

Resolution Description:

Workaround:

The OS generated file for linking stacks (Os_Link_Core<x>_Stacks.lsl - Tasking; Os_Link_Core5_Stacks.ld - HighTec) shall not be used.
Instead a user written linker command file which applies a gap of 16 Bytes between the stacks shall be used

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

ESCAN00069428	[only Tricore Multican && Tasking Compiler] Postbuild does not work CommonAsr_Tricore@EcuC_PreCfgFile_Tasking
ESCAN00093052	DET during initialization of Rtm Monitoring_Asr4Rtm@GenTool_GeneratorMsr
ESCAN00093648	Immediate TxConfirmation not handled by transport layer Cp_XcpOnTcpIpAsr@Implementation
ESCAN00095190	Misinterpretation of header information on reception of PTP frames. SysService_AsrTSynEth@Implementation
ESCAN00096102	Main function tick time resolution smaller than 1 ms is not supported Ccl_Asr4SmCan@GenTool_GeneratorMsr
ESCAN00096158	Ethernet firewall issues a transmission deny although EthIf shall just process a buffer release If_AsrIfEth@Implementation
ESCAN00097961	ECU sends retransmissions of FIN segment if connection actively closed by both peers Tp_AsrTpTcpIp@Implementation
ESCAN00098754	Transmission via TriggerTransmit with IP fragmentation fails If_AsrIfSoAd@Implementation
ESCAN00099928	Communication Interface Queued Rx API forwarding routing path do not work with regular PduR queue Gw_AsrPduRCfg5@Implementation
ESCAN00100378	An immediate NM message is not transmitted Nm_Asr4NmCan@Implementation
ESCAN00100584	Dem doesn't return DEM_CLEAR_FAILED although all ClearEventAllowed callbacks return FALSE Diag_Asr4Dem@Implementation
ESCAN00100732	Dem_DisableDTCRecordUpdate is not processed Diag_Asr4Dem@Implementation
ESCAN00100877	Tx Path is disabled longer than expected Nm_Asr4NmCan@Implementation
ESCAN00101124	Reopening TCP cwnd fails after it has been set to 0 when detecting a too small PathMtu Tp_AsrTpTcpIp@Implementation
ESCAN00101275	Reset behavior of WdgM is not AUTOSAR compliant SysService_Asr4WdM@Implementation
ESCAN00101563	Time synchronisation inconsistency DrvEth__coreAsr@Implementation
ESCAN00101622	Processing of queued Tcp packets is incorrect when Path Mtu is changed to a large value Tp_AsrTpTcpIp@Implementation
ESCAN00101788	SignalGroup invalidation does not work for RX UINT8_DYN signals Il_AsrComCfg5@Implementation
ESCAN00101789	Signal or SignalGroup invalidation does not work for UINT8_DYN TX signals Il_AsrComCfg5@Implementation
ESCAN00101852	IpV4_Icmp_SendDestUnreachable() includes 64 bytes of payload instead of 64 bits Tp_AsrTpTcpIp@Implementation

Index

ESCAN00069428 [only Tricore Multican && Tasking Compiler] Postbuild does not work

Component@Subcomponent:	CommonAsr_Tricore@EcuC_PreCfgFile_Tasking
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Unexcepted behaviour after usage of postbuild data

Postbuild generator does not work for Tasking Compiler.

1.) StructAlignment=Auto:
Pointer-Alignment in struct with 32bit is OK, but 32bit Value in struct is aligned with 16bit (by compiler but not by Generator (32bit))

2.) Struct Alignment=16bit:
Pointer-Alignment in struct with 32bit (by compiler but not by Generator (16bit)) is not ok, but 32 bit Value in struct is algined with 16bit (ok).

There is no possibility to change the compiler alignment for elements in structs. Or different alignment for pointer or normal data.

When does this happen:

When using postbuild loadable

In which configuration does this happen:

- Tricore Multican AND
- Tasking Compiler (tested: 2.5R2, 4.2.R2 the behavior of other versions is unknown)

Resolution Description:

ESCAN00069428 [only Tricore Multican && Tasking Compiler] Postbuild does not work

- use Compiler to translate postbuild data

or change compiler settings to:

```
Cmd line parameter |Description
-----------------------------------------------------------------------------------------
 --align-enum="Align16Bit" |Change natural alignment of type enum to 16Bit.
 |Only required if parameter "SizeOfEnum"
 |in EcuC module is configured to Size32Bit.
-----------------------------------------------------------------------------------------
 --align-int="Align16Bit" |Change natural alignment of type int.
-----------------------------------------------------------------------------------------
 --align-long="Align16Bit" |Change natural alignment of type long.
-----------------------------------------------------------------------------------------
 --align-float="Align16Bit" |Change natural alignment of type float.
-----------------------------------------------------------------------------------------
 --align-double="Align16Bit" |Change natural alignment of type double.
-----------------------------------------------------------------------------------------
 --align-struct-ge16="Align16Bit" |16Bit alignment of unions/structures
 |that are >= 16Bit.
-----------------------------------------------------------------------------------------
 --align-struct-ge64="Align32Bit" |32Bit alignment of unions/structures
 |that are >= 64Bit.
-----------------------------------------------------------------------------------------
```

ESCAN00093052 DET during initialization of Rtm

Component@Subcomponent: Monitoring_Asr4Rtm@GenTool_GeneratorMsr
First affected version: 3.00.00
Fixed in versions:

Problem Description:
What happens (symptoms):

A DET is thrown during initialization of Rtm (Rtm_Init).

If DET is not available or disabled for Rtm, there is a memory out of bounds access.

When does this happen:

During initialization (Rtm_Init).

In which configuration does this happen:

Occurs if:
 - a Vector Generation 6 OS with Scalability Class 3 is used AND
 - At least one measurement point's type (/MICROSAR/Rtm/RtmMeasurementPoint/RtmMeasurementType) is set to ExecutionTime_Nested or ExecutionTime_NonNested.

Resolution Description:
Workaround:

There is no workaround.

Resolution:

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

ESCAN00093648	Immediate TxConfirmation not handled by transport layer
Component@Subcomponent:	Cp_XcpOnTcpIpAsr@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

When the interface layer calls a TxConfirmation before its Transmit function is finished the Xcp transport layer will be in a recursive loop, because the TxConfirmation will trigger a follow up transmission.

This might lead to stack issues.

When does this happen:

Always and immediately

In which configuration does this happen:

When the interface call TxConfirmation in context of the Transmit function.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00095190 Misinterpretation of header information on reception of PTP frames.

Component@Subcomponent:	SysService_AsrTSynEth@Implementation
First affected version:	1.00.00
Fixed in versions:	9.00.00

Problem Description:
What happens (symptoms):

On reception of e.g. PTP Follow-Up frames, the module EthTSyn as PTP slave does not synchronize properly with PTP master.

When does this happen:

The issue does occur always and immediately when PTP frames are expected to be received and processed properly.

In which configuration does this happen:

This symptom may occur on 64-bit hardware platforms.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

The structures are no longer used to access the Ethernet Rx/Tx buffer.
Instead a Byte-Wise-Access with offests is used. This behavior was implemented with:
STORY-5180: Remove structure overlay for Ethernet Rx/Tx buffers

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.

ESCAN00096158 Ethernet firewall issues a transmission deny although EthIf shall just process a buffer release

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

Problem Description:

What happens (symptoms):

Ethernet firewall issues a transmission deny in its logging mechanism although the EthIf shall just process a buffer release which shouldn't involve the Ethernet firewall.
Dependent on the configuration of the Ethernet firewall this results in:
- User-Drop-Event-Notification-Callout enabled -> Call of the User-Notification for dropped frames although there wasn't a "real" frame dropped but a buffer release triggered
- User-Drop-Event-Notification-Callout disabled -> No detectable event

In both cases the issue doesn't affect the functionality of the stack. In first case just the User-Notification is called although it shouldn't. If user tracks dropped frames due to firewall deny this could lead to wrong statistics.

When does this happen:

During transmission abort of an Ethernet frame.

In which configuration does this happen:

Configurations containing an EthFw (Ethernet firewall) module

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00097961	ECU sends retransmissions of FIN segment if connection actively closed by both peers
Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	7.00.05, 9.00.00

Problem Description:

What happens (symptoms):

The ECU sends retransmissions of the FIN,ACK segment after the connection was closed by both peers.

The connection is closed via RST by the ECU after the configured user timeout has expired.

When does this happen:

This issue does only occur if close of TCP connection is initiated by both sides (ECU and remote peer).

Note: In most use cases only one peer actively triggers connection close and the other peer passively closes the connection.
In these cases this issue does not occur.

ECU:
--> Call to TcpIp_Close()
The FIN segment is not send immediately but in the next TcpIp_MainFunction()

<-- TcpIp_RxIndication(): TCP segment with FIN is received from remote host.
This is not a response to a FIN sent by the ECU but the result of an active close call on the remote host.

--> TcpIp_MainFunction()
First transmission of FIN segment by the ECU.

<-- TcpIp_RxIndication(): ACK for FIN of ECU is received.

--> ECU retransmits FIN segment

In which configuration does this happen:

All configurations using TCP.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00098754 Transmission via TriggerTransmit with IP fragmentation fails

Component@Subcomponent: If_AsrIfSoAd@Implementation

First affected version: 9.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Transmission not possible since transmission request is rejected.

When does this happen:

On transmisision of a PDU on configurations described below.

In which configuration does this happen:

Issue happens if the following configuration applies for a PDU transmission:

1. IP fragmentation is enabled:
TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIpV6FragmentationConfig
TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig
and
2. Trigger Transmit is enabled:
SoAd/SoAdConfig/SoAdPduRoute/SoAdTxIfTriggerTransmit
and
3. UDP socket:
SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00099928 Communication Interface Queued Rx API forwarding routing path do not work with regular PduR queue

Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	11.03.00
Fixed in versions:	13.00.00

Problem Description:

What happens (symptoms):

Communication Interface Queued Rx API Forwarding routing paths do not work correctly.
The Queue will always overflow and will therefore be flushed. A DET error with API-ID PDUR_FCT_RMIF_FQ and Error code PDUR_E_PDU_INSTANCES_LOST is reported if error reporting is enabled.
After the flush the same behavior occurs again.
The Pdu which is first in the queue will be forwarded to the upper layer in context of the MainFunction. All other Pdus will remain in the queue and will not be forwarded. They will be discarded at the queue flush.

When does this happen:

Always during runtime.

In which configuration does this happen:

The routing path must satisfy all of the following conditions:
- Communication interface Pdu
- API Forwarding
- Rx direction
- Queue depth set to some value > 0

Resolution Description:

Workaround:

For Multicore routing paths use the so called 'Multicore queue'. The queueing can't be used for singlecore routing paths.

Resolution:

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

ESCAN00100378 An immediate NM message is not transmitted

Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	1.00.00
Fixed in versions:	6.03.02, 9.00.00

Problem Description:

What happens (symptoms):

--

An NM message is not transmitted in the expected time frame.

When does this happen:

--

A network is actively requested, therefore an immediate NM message is triggered.
One of the lower modules (e.g. CanIf) does not accept the trigger of the NM message that is originated from the CanNm module.
The transmission is not retried after the expected time, but after the regular NM message cycle.

In which configuration does this happen:

--

/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/
CanNmImmediateNmTransmissions == 1

Resolution Description:

Workaround:

--

No workaround available.

Hint: The problem might be solved when the CanSM_MainFunction is executed prior to the CanNm_MainFunction.

Resolution:

--

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

ESCAN00100584 Dem doesn't return DEM_CLEAR_FAILED although all ClearEventAllowed callbacks return FALSE

Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	13.00.00
Fixed in versions:	16.00.00

Problem Description:

What happens (symptoms):

The Dem doesn't return DEM_CLEAR_FAILED and clears global statistics, although no DTC was cleared.

Global statistics can for example be (depending on selected group and origin):

- Data for Service 0x19 with subfunction 0x0B, 0x0C, 0x0D and 0x0E
- Data for Dem_GetEventMemoryOverflow
- Data for OBD PIDs (e.g. PID21, PID4D, PID30, PID31, PID4E, PID01)
- Data for WWHOBD Continuous MI-Counter
- Event debounce data
- Data for Pending FIDs

When does this happen:

When a clear request for a group of DTCs is triggered and the ClearEventAllowed callbacks for all DTCs of the group return FALSE (in case of event combination ClearEventAllowed returns FALSE for at least one event of the DTC).

In which configuration does this happen:

All DTCs of the group to clear have a ClearEventAllowed callback configured (Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed).

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00100732	Dem_DisableDTCRecordUpdate is not processed
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	14.05.00
Fixed in versions:	16.00.00

Problem Description:

What happens (symptoms):

Dem_DisableDTCRecordUpdate is not processed and returns DEM_PENDING.
After a (configurable) number of DEM_PENDING, Dcm cancels requested UDS Service and sends negative response 0x10 (generalReject)

When does this happen:

In rare cases when requesting Service $19 (ReadDTCInformation) with subfunctions

0x04 (reportDTCSnapshotRecordByDTCNumber)
0x06 (reportDTCExtDataRecordByDTCNumber)
0x10 (reportMirrorMemoryDTCExtDataRecordByDTCNumber)
0x18 (reportUserDefMemoryDTCSnapshotRecordByDTCNumber)
0x19 (reportUserDefMemoryDTCExtDataRecordByDTCNumber)

via DCM.

In which configuration does this happen:

Dem/DemGeneral/DemDcmSupport == TRUE

Resolution Description:

Workaround:

No workaround available.
The problem should only occur rarely due to unlikely race condition. Repeating the UDS request will be successful.

Resolution:

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

ESCAN00100877	Tx Path is disabled longer than expected
Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	1.00.00
Fixed in versions:	This ticket is not considered for fixing.
Problem Description:	

ESCAN00100877 Tx Path is disabled longer than expected

What happens (symptoms):

----

(1) The effects depend on the 'Tx Timeout Exception Callback' implementation within an upper layer (normally CDD) - this function call will occur later than expected.
(2) If Partial Networking is used, the CAN cell reinitialization will occur later than expected.

Note: The delay depends on how many message transmit requests are performed within a distance smaller than the message timeout time.

When does this happen:

A further NM message transmit request is issued before the current one is acknowledged and NM messages cannot be sent successfully (e.g. due to a BusOff or due to no other ECU is awake in Partial Network Case).

In which configuration does this happen:

----

(1) '/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmTxTimeoutErrorCallback' is configured

OR

(2) 'CANNM_PN_TX_TIMEOUT_EXCEPTION_FEATURE_ENABLED == STD_ON' (check CanNm_Cfg.h)

Required for both variants:
(
 '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmImmediateNmCycleTime'
< '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmMsgTimeoutTime'
)

OR

(
 Function order as follows, e.g. in one task:
 Nm_MainFunction
 CanNm_MainFunction
AND
 '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmBusSynchronizationEnabled' == ON
)

OR

(
 '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmComUserDataSupport' ==ON
AND
 '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmUserDataTxPdu' is set to DIRECT and some of its signals to TRIGGERED in Com
)

OR

(
 '/MICROSAR/CanNm/CanNmGlobalConfig/CanNmCoordinatorSyncSupport' == ON
)

Resolution Description:

ESCAN00100877 Tx Path is disabled longer than expected

Workaround:

No workaround available.

Resolution:

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

ESCAN00101124 Reopening TCP cwnd fails after it has been set to 0 when detecting a too small PathMtu

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	9.00.00

Problem Description:

What happens (symptoms):

Sending of data on a certain TCP socket is not possible.

When does this happen:

The wrong behaviour happens after first calling TcpIp_Tcp_CbkPathMtuChg (usually from IpV4 or IpV6) for an IP address that is used by an active TCP socket and with a very small length (smaller than a useful minimum), and later calling TcpIp_Tcp_CbkPathMtuChg for the same IP address with a bigger length. Then the TCP CWND (congestion window) will not be opened and therefore sending any data will not be possible.

In normal environments this should never happen. Transport layers carrying IP packets should support a minimum segment size of 576 Bytes (RFC 1122), but 'normal' Ethernet environments support an MTU of 1500 Bytes. So it is very unlikely that the TCP receives a call of TcpIp_Tcp_CbkPathMtuChg with a size smaller than the maximum TCP header size (60 Bytes) that would cause the cwnd to be set to 0.

In which configuration does this happen:

In all configurations using TCP

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is corrected by modification of all affected work-products.
Error is resolved in file "TcpIp_Tcp.c" function "TcpIp_Tcp_CbkPathMtuChg".

ESCAN00101275 Reset behavior of WdgM is not AUTOSAR compliant

Component@Subcomponent:	SysService_Asr4WdM@Implementation
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The reset behavior of the WdgM is not according to AUTOSAR. The WdgM expects the WdgIf API WdgIf_SetTriggerCondition to return a value of type Std_ReturnType, whereas AUTOSAR specifies this API to be void.

Currently, this return value is evaluated by the WdgM and this does not make sense by now because AUTOSAR Wdg driver and WdgIf are expected and thus no return value will be available.

This return value is used to decide if the Mcu_PerformReset() function shall be called. But this is only done if the configuration parameter WdgM_SecondResetPath is set to STD_ON. WdgM_SecondResetPath is a non AUTOSAR parameter specified.

Since only AUTOSAR compliant Wdg drivers are allowed the Mcu_PerformReset will never be called (no return value will be available).

AUTOSAR specifies another reset behavior:
[SWS_WdgM_00133] If the configuration parameter WdgMImmediateReset [ECUC_WdgM_00339] is set to TRUE and the Global Supervision Status has reached the state WDGM_GLOBAL_STATUS_STOPPED, the Watchdog Manager module shall call the MCU service Mcu_PerformReset on the MCU Driver module.

[SWS_WdgM_00122] If the Global Supervision Status has recomputed as WDGM_GLOBAL_STATUS_STOPPED, then the Watchdog Manager module shall call WdgIf_SetTriggerCondition for all watchdogs not configured as WDGIF_OFF_MODE [ECUC_WdgM_00332] with set to WdgMWatchdogDeviceRef [ECUC_WdgM_00348] and set to zero.

Setting the trigger condition to zero will immediately prevent the Watchdog Driver module from triggering the hardware watchdog.

Hint: The WdgIf could return a return value even if the Wdg driver is a void function (AUTOSAR driver). But only E_NOT_OK is returned on config error (DET)!

When does this happen:

Only on reset cause. This happens if the WdgM calculates the global supervision status to be STOPPED which results in an unrecoverable supervision failure,

In which configuration does this happen:

Always.

Resolution Description:

ESCAN00101275 Reset behavior of WdgM is not AUTOSAR compliant

Workaround:

No workaround available.

Resolution:

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

ESCAN00101563 Time synchronisation inconsistency

Component@Subcomponent:	DrvEth__coreAsr@Implementation
First affected version:	3.00.00
Fixed in versions:	5.00.00, 4.03.01

Problem Description:

What happens (symptoms):

During time synchronization (PTP) time jumps of about 4 ms are occurring which can lead to out of sync situations

When does this happen:

During runtime

In which configuration does this happen:

/MICROSAR/Eth_Tc3xx/Eth/EthConfigSet/EthCtrlConfig/EthCtrlEnablePtp == TRUE

AND

(
 ETH stack running on different controller cores (multicore use usage)

 OR

 operating the ETH driver in polling mode == non interrupt mode
)

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101622	Processing of queued Tcp packets is incorrect when Path Mtu is changed to a large value
Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	9.00.00

Problem Description:

What happens (symptoms):

When Path Mtu is set to a large value (greater than maximum segment length), queued messages sent should be of maximum segment length size.

But it is observed that the queued Tcp messages are not sent out in this case.

When does this happen:

When TcpIp_Cbk_VPathMtuChg is called and Path Mtu is changed to a value larger than Path Mtu.

In which configuration does this happen:

/MICROSAR/TcpIp/TcpIpGeneral/TcpIpTcpEnabled == TRUE

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101788 SignalGroup invalidation does not work for RX UINT8_DYN signals

Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	3.00.00
Fixed in versions:	16.02.00

Problem Description:

What happens (symptoms):

----

signal invalidation is not performed for UINT8_DYN groupSignals.
Invalid values are still received and no invalid action will be performed.

When does this happen:

During runtime whenever an invalid value is received for a signalGroup which contains an UINT8_DYN groupSignal.

In which configuration does this happen:

/MICROSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComSignalDataInvalidValue is configured and set to some value
AND
/MICROSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComSignalType is configured to UINT8_DYN
AND
/MICROSAR/Com/ComConfig/ComSignalGroup/ComDataInvalidAction is configured and set to some value
AND
/MICROSAR/Com/ComConfig/ComIPdu/ComIPduDirection is set to RECEIVE

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101789 Signal or SignalGroup invalidation does not work for UINT8_DYN TX signals

Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	3.00.00
Fixed in versions:	16.02.00

Problem Description:
What happens (symptoms):
----
signal/signalGroup invalidation does not work for TX signals UINT8_DYN signals.
The signal/signalGroup will never be set to invalid for UINT8_DYN signals whenever the application tries to indicate that no valid signal value can be provided.

When does this happen:

During runtime whenever the application tries to invalidate a signal or signalGroup.

In which configuration does this happen:

(

(
/MICROSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComSignalDataInvalidValue is configured and set to some value
AND
/MICROSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComSignalType is set to UINT8_DYN
)

OR

(
/MICROSAR/Com/ComConfig/ComSignal/ComSignalDataInvalidValue /MICROSAR/Com/ComConfig/ComSignal/ComSignalDataInvalidValue is configured and set to some value
AND
/MICROSAR/Com/ComConfig/ComSignal/ComSignalType is set to UINT8_DYN
)
)

AND

/MICROSAR/Com/ComConfig/ComIPdu/ComIPduDirection is set to TRANSMIT

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00101852	**IpV4_Icmp_SendDestUnreachable() includes 64 bytes of payload instead of 64 bits**
Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	6.02.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

IpV4_Icmp_SendDestUnreachable includes 64 bytes of Payload instead of to include only 8 bytes (64 bits) of Payload.

When does this happen:

When ECU receives a unicast packet with the following properties:
- The upper layer protocol is unknown.
- The UDP destination port is not open.

In which configuration does this happen:

Controller has configured
/ActiveEcuC/TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/
TcpIpIcmpConfig[0:TcpIpIcmpDestinationUnreachableEnabled] = TRUE

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101962 UserData is always '0' in Offset Sub-Tlv

Component@Subcomponent:	SysService_AsrTSynEth@Implementation
First affected version:	4.00.00
Fixed in versions:	8.00.00

Problem Description:

What happens (symptoms):

The UserData included in the Ofs Sub-TLV is always '0'

When does this happen:

At transmission of a FollowUp message

In which configuration does this happen:

/MICROSAR/EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/
EthTSynGlobalTimeMaster/EthTSynTLVFollowUpOFSSubTLV == TRUE

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

error is resolved in file "EthTSyn_Master_Int.c" function
"EthTSyn_MasterTx_AppendArFollowUpOfsTlvs()"

ESCAN00102065 Undefined UserData passed to StbM

Component@Subcomponent:	SysService_AsrTSynEth@Implementation
First affected version:	5.00.00
Fixed in versions:	8.00.00

Problem Description:

What happens (symptoms):

Undefined UserData is passed to StbM

When does this happen:

At reception of a FollowUp message without a UserData Tlv on a slave port

In which configuration does this happen:

/MICROSAR/EthTSyn/EthTSynGeneral/EthTSynMessageCompliance == FALSE
AND
/MICROSAR/EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeSlave is condfigured

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

error is resolved in file "EthTSyn_Slave_Int.c" function "EthTSyn_SlaveRx_HandleArFollowUpTlvs()":

```
/* #10 Handle possible FollowUp Tlvs: Retrieve Pointer to the first Ar Sub-Tlv and the Sub-Tlv count. */
retVal = EthTSyn_SlaveRx_GetFirstArSubTlvPtrAndCnt(PortIdx, ComMsgHdrPtr, &firstSubTlvPtr, &subTlvCnt); /* SBSW_ETHTSYN_PTR_FORWARD_STRUCT_MEMBER_AND_LOC_VAR */
 if ((retVal == E_OK) && (firstSubTlvPtr != NULL_PTR))
 {
 [...]
 }
-- Add
else
{
 (*UserDataPtrPtr) = NULL_PTR;
}
--
```

ESCAN00102098 Qbv scheduling issue for cascaded switches

Component@Subcomponent:	DrvEthSwitch_Sja1105PQRSAsr@Implementation
First affected version:	4.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

If more than one switch is used in a cascaded switch configuration and a Qbv scheduling is started on more than one switch than the Qbc scheduling is not working for any switch beside of the one where Qbv has been started first.

When does this happen:

Always when Qbv scheduling is started on a cascaded switch and on one switch Qbv scheduling has been started already before.

In which configuration does this happen:

A cascaded switch configuration with active Qbv scheduling on more than one switch.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00102183 Time Recording without applied Time Correction

Component@Subcomponent:	SysService_AsrStbM@Implementation
First affected version:	5.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The local time elements LocSeconds and LocNanoSeconds of Global Time Precision Measurements contain the local time without Time Correction applied.

When does this happen:

This might happen during runtime, when measurement data is provided for Synchronized Time Bases.

In which configuration does this happen:

This happens for Time Slaves or Gateways with /MICROSAR/StbM/StbMGeneral/StbMTimeRecordingSupport set to TRUE and Time Correction (/MICROSAR/StbM/StbMSynchronizedTimeBase/StbMTimeCorrection) configured.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102187 Time leap checks don't consider Time Correction

Component@Subcomponent:	SysService_AsrStbM@Implementation
First affected version:	4.02.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Time Leap checks might be inaccurate because of missing Time Correction of the local time.

When does this happen:

This might happen during runtime, when StbM_BusSetGlobalTime() is called.

In which configuration does this happen:

This happens for Time Slaves or Gateways with Time Leap checks (/MICROSAR/StbM/StbMSynchronizedTimeBase/StbMTimeLeapFutureThreshold or /MICROSAR/StbM/StbMSynchronizedTimeBase/StbMTimeLeapPastThreshold) and Time Correction (/MICROSAR/StbM/StbMSynchronizedTimeBase/StbMTimeCorrection) configured.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102202	**StbM_GetTimeLeap() might erroneously output 0 for the Time Leap value**
Component@Subcomponent:	SysService_AsrStbM@Implementation
First affected version:	5.03.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The Time Leap value returned by StbM_GetTimeLeap() might be 0 although a Time Leap occured.

When does this happen:

This happens during runtime if an StbMTimeLeapPastThreshold exceedance is followed by an StbMTimeLeapFutureThreshold exceedance.

In which configuration does this happen:

This might happen for time slaves or gateways with time leap detection /MICROSAR/StbM/StbMSynchronizedTimeBase/StbMTimeLeapPastThreshold and /MICROSAR/StbM/StbMSynchronizedTimeBase/StbMTimeLeapFutureThreshold configured.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102425 EthTSyn answers to last Pdelay_Req instead of first Pdelay_Req

Component@Subcomponent:	SysService_AsrTSynEth@Implementation
First affected version:	4.00.00
Fixed in versions:	8.00.00

Problem Description:
What happens (symptoms):

EthTSyn answers to last Pdelay_Req

When does this happen:

When multiple consecutive Pdelay_Req messages are received on the same Port before EthTSyn answers with Pdelay_Resp and Pdelay_Resp_FollowUp

In which configuration does this happen:

/MICROSAR/EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPdelayConfig/
EthTSynGlobalTimePdelayRespEnable == TRUE
AND
/MICROSAR/EthTSyn/EthTSynGeneral/EthTSynEnableSwitchMgmt == FALSE

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

error is resolved in file "EthTSyn_Pdelay_Int.c" function "EthTSyn_Pdelay_ProcRcvdReqMsg()":
- Add validation of Pdelay response state machine state

ESCAN00102432 Reception of valid IF-PDU over TCP is rejected

Component@Subcomponent:	If_AsrIfSoAd@Implementation
First affected version:	11.00.00
Fixed in versions:	13.00.00

Problem Description:

What happens (symptoms):

IF-PDU sent over TCP is not received correctly.

When does this happen:

On reception of the last TCP segment of an segmented IF-PDU when TCP segment further PDUs.
The TCP segmentation may be forced by the TCP protocol or by a segmented reception buffer.

In which configuration does this happen:

IF-PDU reception over TCP with PDU header option.

SoAd/SoAdConfig/SoAdSocketRoute with
/SoAdSocketRouteDest/SoAdRxUpperLayerType == IF and
/SoAdRxPduHeaderId exists over
SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection with
SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp exists.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102488 Tx interval might get longer for delayed TxConfirmation()

Component@Subcomponent:	SysService_AsrTSynCan@Implementation
First affected version:	3.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Periodic SYNC/OFS messages might be delayed.

When does this happen:

This happens during runtime for time masters, if the durations between CanIf_Transmit() and CanTSyn_TxConfirmation() for a SYNC/FUP or OFS/OFNS sequence are too long in total.

In which configuration does this happen:

This might happen for Time Masters with CanTSynMasterConfirmationTimeout greater than (CanTSynGlobalTimeTxPeriod - 2 x CanTSynMainFunctionPeriod) / 2.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102558 Two ARP entries are created for the same destination address which leads to TCP_RST

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	10.00.00

Problem Description:

What happens (symptoms):

Two arp entries are created in ARP cache for the same destination address. It leads to TCP_RST.
Two ARP requests are sent for the same destination address.

When does this happen:

When ARP Response is received immediately after ARP request is sent.

In which configuration does this happen:

/TcpIp/TcpIpGeneral/TcpIpIpV4General/TcpIpArpEnabled == TRUE

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102605 Restart of cyclic transmission after Immediate Transmit might be delayed.

Component@Subcomponent:	SysService_AsrTSynCan@Implementation
First affected version:	3.03.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The restart of the cyclic transmission of SYNC/FUP and OFS/OFNS sequences after an Immediate Time Synchronization sequence might be delayed.

When does this happen:

This might happen during runtime after an Immediate Time Synchronization sequence has been send.

In which configuration does this happen:

This happens for Time Masters with /MICROSAR/CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynImmediateTimeSync set to TRUE and /MICROSAR/CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynCyclicMsgResumeTime smaller than /MICROSAR/CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeTxPeriod

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102650 [IPV6] Invoking TcpIp_ReleaseIpAddrAssignment() for manual DhcpV6 address does not have any effect

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

When user invokes TcpIp_RequestIpAddrAssignment() for a Manually triggered DhcpV6 address id, DhcpV6 messages (SOLICIT) will be observed.

Now, when user invokes TcpIp_ReleaseIpAddrAssignment() before the address is assigned to the address id, further Dhcpv6 messages for the address id should not be seen.

But it is observed that ECU continues to send further Dhcpv6 messages.

When does this happen:

When TcpIp_ReleaseIpAddrAssignment() is invoked for manual DhcpV6 address before a valid address has been assigned.

In which configuration does this happen:

The issue does only observed if all of the following conditions are fulfilled:

- IPV6 is enabled
- Manually triggered DhcpV6 Address is configured

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102703	**Received data of a Mode Switch interface can be interpreted in a wrong way by SWCs for data of BswM Mode Declaration Groups in case that numerical values are used instead of symbols**
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	9.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

The received data of a Mode Switch interface can be interpreted in a wrong way by SWCs and therefore code is executed which might lead to unintended behavior.

The following shows an example with the usage of a numerical value for comparison. Instead a symbolic name which is provided for each mode declaration by the Rte should be used.

Example with numerical value usage:

```
if(Rte_Mode_<PortName>_<ModeDeclarationGroupPrototypeName>() == 0u)
{
 // 0 could be of different meaning between BswM configuration and Rte
}
```

When does this happen:

If SWC code uses the numerical values of the BswM configuration directly for comparison of Mode Declarations instead of the symbolic defines provided by the Rte. The values which are configured for the BswM Mode Declaration Group in the BswM configuration might not fit to the values provided by the Rte via the symbols.

In which configuration does this happen:

In all configurations which use numerical values instead of symbolic names for comparison of values of Mode Declaration Groups provided by BswM [/MICROSAR/BswM/BswMConfig/BswMRteModeDclGroup].

Resolution Description:
Workaround:

Use symbolic names.

Resolution:

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

ESCAN00102854 Synchronization of OS Schedule Tables is not working properly for Offset Time Bases

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

Problem Description:

What happens (symptoms):

Synchronization of OS Schedule Tables is not working properly for Offset Time Bases.

When does this happen:

Triggered Customer is configured for Offset Time Base.

In which configuration does this happen:

/MICROSAR/StbM/StbMTriggeredCustomer/StbMSynchronizedTimeBaseRef is configured to reference an Offset Time Base.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

Technical resolution:
- In StbM.c the function StbM_TriggerCustomers() was adapted to handle Offset Time Bases as well.
- The test suite was adapted in order to consider also Offset Time Bases for TriggerCustomer tests.

ESCAN00102881 StbM doesn't handle more than one Slave per Domain in EthTSyn correctly

Component@Subcomponent:	SysService_AsrStbM@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Time Recording and Gateway configurations are not recognized by StbM, which means STBM_TIME_RECORDING_SUPPORT_USED and STBM_TIMEGATEWAYOFTIMEBASECONFIG are defined STD_OFF and the corresponding data structures are invalid.

When does this happen:

This happens at generation time if the configuration is as below.

In which configuration does this happen:

There is more than one /MICROSAR/EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeSlave for the same /MICROSAR/EthTSyn/EthTSynGlobalTimeDomain configured in EthTSyn and either a /MICROSAR/EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeMaster is additionaly configured for the same EthTSynGlobalTimeDomain or /MICROSAR/StbM/StbMSynchronizedTimeBase/StbMTimeRecording is configured for the /MICROSAR/StbM/StbMSynchronizedTimeBase referenced by the EthTSynGlobalTimeDomain.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102882 Payload pointer in Sync/FollowUpSent callback points to wrong memory location

Component@Subcomponent:	SysService_AsrTSynEth@Implementation
First affected version:	4.00.00
Fixed in versions:	9.00.00

Problem Description:

What happens (symptoms):

The payload provided via SyncPtr/FollowUpPtr in Sync/FollowUpSent callback does not correspond to the actual transmitted payload.

When does this happen:

When the Sync/FollowUpSent callback is called by the EthTSyn (i.e. Sync/FollowUp message was transmitted)

In which configuration does this happen:

When
/MICROSAR/EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeMaster/EthTSynSyncSentCallBackFunction
AND/OR
/MICROSAR/EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeMaster/EthTSynFollowUpSentCallBackFunction
is/are configured/present

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

Due to changes in the access method for Ethernet Rx/Tx buffers, the signature of the Sync/FollowUpSent callback was changed and the Issue was resolved implicitly.

For a possible fix in early versions, the Sync/FollowUp payload pointer has to be created by using the TxBufferPtr with the corresponding offset (i.e. length of common message header).

ESCAN00102953 IF TxConfirmation is called with locked interrupts

Component@Subcomponent:	If_AsrIfSoAd@Implementation
First affected version:	11.00.00
Fixed in versions:	14.00.00

Problem Description:

What happens (symptoms):

<Up>_[SoAd][If]TxConfirmation is called with locked interrupts.
In case OS API is called in context of this function an OS error may occur. In case no such functionality is used this issue has no symptom.

When does this happen:

On transmission confirmation of a PDU with configuration described below.

In which configuration does this happen:

Issue occurs on a
SoAd/SoAdConfig/SoAdPduRoute with SoAdTxUpperLayerType set to "IF"
over a
SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection with existing SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp (i.e. PDU transmission over TCP).

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

Index

Index

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.

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:	8.06.04, 12.01.00

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 that 65535

Resolution Description:

Workaround:

The equation shall become true: (<TimeParameter> / DcmTaskTime) < 63535.

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.

ESCAN00093263 Missing "if" statement

Component@Subcomponent:	DrvCan_Mpc5700McanLl@Implementation
First affected version:	2.08.00
Fixed in versions:	2.09.01

Problem Description:

What happens (symptoms):

The transition to "Start" or "Stop" Mode is returned erroneously as "Done" to upper layers.
E.g. the MCAN can still be active with pending Tx requests although Stop Mode reached is notified.

When does this happen:

At run time.

In which configuration does this happen:

Only for AutoSar 4.x
AND
MCAN Revision 2.x, 3.0.0, 3.0.1
AND
CAN_BOSCH_ERRATUM_008 == STD_ON.

Resolution Description:

Workaround:

Enable "Hardware Loop Check by application" and check for timeout notifications for "kCanLoopStart"/"kCanLoopStop".
If a timeout appears the requested mode change must be repeated.

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:	This ticket is not considered for fixing.

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.

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:	15.03.00

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.

ESCAN00096249	Validator PDUR12501 is not shown as error for PduRSrcPduRef for not supported source modules
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	9.01.00
Fixed in versions:	15.03.00

Problem Description:

What happens (symptoms):

Validator message PDUR12501 is shown on a PduRSrcPduRef. It is shown as information only, but in reality shall be an error message for source modules which do not support N:1 fan in on global Pdu level.

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 PduRSrcPduRef is referenced by more than one other container. This kind of N:1 routing path is only supported for communication interface Pdus of SoAd, CanIf and FrIf.

Resolution Description:

Workaround:

Check if the validation message is shown for a PduRSrcPduRef.
If No, then you're not affected.
If Yes, this kind of configuration is only supported for communication interface Pdus of the SoAd, CanIf and FrIf module. Everything else shall not be configured.

Resolution:

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

ESCAN00096363	TcpIp reports too low MSS if size of ethernet TX buffers is smaller than size of RX buffers
Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

```
--------------------------------------------------------------------
```

The Maximum Segment Size (MSS) announced by the TCP is smaller than expected.

When does this happen:

```
--------------------------------------------------------------------
```

When the TCP sends the SYN / SYN+ACK segment containing the MSS option.

In which configuration does this happen:

```
--------------------------------------------------------------------
```

The issue only occurs if the ethernet buffer sizes are optimized for special use cases where the ECU always sends smaller ethernet packets than it receives:

Eth/EthConfigSet/EthCtrlConfig/EthCtrlTxBufLenByte < Eth/EthConfigSet/EthCtrlConfig/EthCtrlRxBufLenByte.

Resolution Description:

Workaround:

```
--------------------------------------------------------------------
```

Configure at least one ethernet TX buffer that is equal to or greater than the ethernet RX buffers.

see Eth/EthConfigSet/EthCtrlConfig/EthTxBufConfig

Note: It is also possible to configure just a few full-sized TX buffers and additional smaller TX buffers.

Resolution:

```
--------------------------------------------------------------------
```

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

ESCAN00096364 Frame priority set for TCP listen sockets is not inherited to accepted sockets

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The ethernet frames sent on accepted TCP connections do not use the frame priority that was set for the TCP listen socket.

When does this happen:

The issue occurs, if the frame priority is only set for a TCP listen socket via the API TcpIp_ChangeParameter() instead of setting the value in the <Up>_TcpAccepted()-Callback for each accepted socket.

In which configuration does this happen:

If SoAd is the upper layer of the TcpIp the issue occurs in the following configuration:

SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp/
SoAdSocketTcpInitiate == FALSE
SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketFramePriority != TcpIp/TcpIpConfig/
TcpIpCtrl/TcpIpIpFramePrioDefault

Resolution Description:

Workaround:

Configure the default frame priority for the TcpIp controller TcpIp/TcpIpConfig/TcpIpCtrl/
TcpIpIpFramePrioDefault accordingly.

If necessary a different frame priority may be configured inside the SoAd for UDP socket connection groups and TCP socket connection groups with SoAdSocketTcpInitiate == TRUE.

- or -

If not SoAd is the upper layer: Set the desired frame priority for each accepted TCP socket.

Resolution:

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

ESCAN00097402 Init API uses incorrect API Id for DET reporting

Component@Subcomponent:	DrvEth__coreAsr@Implementation
First affected version:	3.02.00
Fixed in versions:	4.03.00, 5.00.00

Problem Description:

What happens (symptoms):

If a DET error is reported by Eth_30_DERIVATIVE_Init API the incorrect API Id of Eth_30_DERIVATIVE_ControllerInit (0x02) is used instead of correct API Id for Eth_30_DERIVATIVE_Init (0x01)

When does this happen:

Always if a DET error is reported from Eth_30_DERIVATIVE_Init API

In which configuration does this happen:

DET reporting enabled

Resolution Description:

Workaround:

Ignore Error ID.

Resolution:

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

ESCAN00097986 Using the same STM module on different cores may cause incorrect timer initialization.

Component@Subcomponent: Os_PlatformTriCoreAurixGen7@Implementation

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Using the same STM module on different cores may cause incorrect timer initialization.
Compare channel 0 is used by a different core than compare channel 1 in such a usecase.

The following behaviors may be observed:

- The timer interrupt never occurs.
- The timer interrupt is routed to the wrong core.
- The timer interrupt occurs on an unexpected frequency

When does this happen:

There may be a racing condition among the cores during initialization of the STM hardware.
The cores may access registers simultaneously.

In which configuration does this happen:

Using the same STM module (with different channels) on different cores

Resolution Description:

Workaround:

Do not use the same STM for different cores.

Resolution:

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

ESCAN00098575	Com Timeouts occur for LIN Rx signals of inactive schedules
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Com Rx Timeouts occur for LIN signals which should be inactive because the corresponding LIN schedule is not active.

When does this happen:

Always during runtime if a LIN channel is in FULL_COMMUNICATION and the corresponding schedule is not active.

In which configuration does this happen:

Only for LIN channels with configured LinIf schedules (LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable) which refer the same LinIfFrame (LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame) multiple times.

AND

For at least one signal of this Frame a Timeout is configured (Com/ComConfig/ComSignal/ComTimeout).

Resolution Description:

Workaround:

Adapt the auto-configured rules manually, so that Pdu Groups are only enabled if the corresponding schedule is active.

Resolution:

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

ESCAN00098607 Unexpected DET Call when disabling active transport protocol gateway Routing

Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	11.03.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

A DET error is reported with Module Id 'PDUR_MODULE_ID', Api Id 'PDUR_FM_ACTIVATENEXT' and Error Id 'PDUR_E_FATAL'. If the DET is non blocking, the ECU resumes normally.

When does this happen:

Always during runtime, if a destination of a transport protocol gateway routing path is deactivated via routing path groups and a new Pdu is received on this routing path.

In which configuration does this happen:

Transport protocol gateway routing path exist which use routing path groups.
AND
DET Reporting is enabled.

Resolution Description:

Workaround:

Ignore specific DET error.

Resolution:

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

ESCAN00098880	User is not reminded to configured at least 1 reassembly buffer in a fragmentation config. where "Enable Packet Reassembly" is enabled.
Component@Subcomponent:	Tp_AsrTpTcpIp@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

In case this issue occurs the user can change the value of GUI-parameter "Number of Reassembly Buffers":
 /MICROSAR/TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig/TcpIpIpNumReassDgrams
can be set to value 0 even though the value of GUI-parameter "Enable Packet Reassembly"":
 /MICROSAR/TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig/TcpIpIpFragmentationRxEnabled
is TRUE and thus states that the Rx-path might be used and thus packet reassembly is required which at least requires 1 "Reassembly Buffer".

Thus a misconfiguration is not detected by a validator and the user is able to misconfigure the IpV4.

The issue be seen by configuring a fragmentation config container wrongly as described above.
AFTER the issue occurred generation can be successfully done for TcpIp.
The consequence, the implication is that fragmented reception on IP level does not work for the affected IpV4 controller. (The issue is PERMANENT.)

When does this happen:

This happens only under specific circumstances namely in case: Refer to section "In which configuration does this happen:".

In which configuration does this happen:

Any where an IpV4 controller:
 /MICROSAR/TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpIpVXCtrl/TcpIpIpV4Ctrl
has a fragmentation config:
 /MICROSAR/TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig
where rx-fragmentation is enabled because parameter:
 /MICROSAR/TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig/TcpIpIpFragmentationRxEnabled
is TRUE and no fragmentation buffer is configured, at all, because parameter:
 /MICROSAR/TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig/TcpIpIpNumReassDgrams
has value 0.

Resolution Description:

Workaround:

For each fragmentation config of an IpV4 controller that has "Packet Reassembly" enabled:
Configure at least one "Reassembly Buffer". Otherwise there is no further workaround available.

Resolution:

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

ESCAN00098975	OS protection hook may be triggered by FiM_SetSearchEventIdTableFct
Component@Subcomponent:	SysService_Asr4FiM@Implementation
First affected version:	5.00.00
Fixed in versions:	6.00.00

Problem Description:

What happens (symptoms):

Due to a write access to a local static variable the OS protection hook may be called.

When does this happen:

During write access to searchEventIdTableFct in FiM_SetSearchEventIdTableFct.

In which configuration does this happen:

If a SC3 or SC4 OS is used with activated memory protection
AND
the FiM module does not have write access to the memory section where local static variables are located.

Resolution Description:

Workaround:

Workaround 1:
Configure the memory protection of the OS to allow write access to the memory section where local static variables are located.

Workaround 2:
Define the symbol FIM_LOCAL as follows
#define FIM_LOCAL
either by setting it in one of the header files included by FiM.c or by compiler command line (-D option).

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:	16.00.00

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:

If no event with Dem/DemConfigSet/DemEventParameter/DemEventClass/DemEventDestination == DEM_DTC_ORIGIN_SECONDARY_MEMORY is configured, set Dem/DemGeneral/DemMaxNumberEventEntrySecondary == 0.

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.

ESCAN00100272 Wrong configuration of Ethernet buffers which is not prevented by validation could lead to runtime issues

Component@Subcomponent:	DrvEth__coreAsr@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	5.00.00

Problem Description:

What happens (symptoms):

The Ethernet reception buffer configuration can be performed erroneously without a validation error message. The generation result can lead to runtime errors in these cases. The issues occur if more than one reception buffer configurations are created and QoS is disabled.

The issue can be detected during runtime, if the reception of Ethernet frames stops after the number of configured buffers in the first buffer configuration were received.
The transmission of frames is still working in the described error condition and the frames are still received on hardware. The interrupts are working as well.

When does this happen:

During runtime.

In which configuration does this happen:

EthConfigSet/EthCtrlConfig/EthCtrlEnableQoS disabled, more than one EthConfigSet/EthCtrlConfig/EthRxBufConfig configured

Resolution Description:

Workaround:

Never configure more than one EthRxBufConfig, if QoS is disabled. There is no use-case for this misconfiguration.

Resolution:

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

ESCAN00100368 IPv6 calls IpAssignmentChange callback for other source address before link-local address is configured

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation_IpV6
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The IPv6 should wait until the link-local address is configured before notifying upper layers about the assignment of other local addresses via the IpAssignmentChange callback.
But the IPv6 calls the IpAssignmentChange callback before the link-local address is ready.

When does this happen:

The issue may occur if a Router Advertisement is received while the link-local address is still in TENTATIVE state. (IpAssignmentChange callback for link-local address not called yet).
Occurrence of the issue depends on the random timeouts that are selected by the IPv6 in order to defer the Duplicate Address Detection process.

In which configuration does this happen:

/MICROSAR/IpV6/IpV6Ndp/IpV6DadTransmits > 0
AND
/MICROSAR/IpV6/IpV6Ndp/IpV6EnableOptimisticDad == FALSE
AND
/MICROSAR/IpV6/IpV6Ndp/IpV6EnableSlaacDelay == TRUE

Resolution Description:

Workaround:

The issue can be avoided by changing one of the parameters listed under "Description".

The recommended work-around for the SCC use case is to set IpV6EnableOptimisticDad == TRUE.

Resolution:

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

ESCAN00100376 TxTimeoutException is called earlier than configured

Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	1.00.00
Fixed in versions:	9.00.00

Problem Description:

What happens (symptoms):

The notifications Nm_TxTimeoutException and CanSM_TxTimeoutException are called earlier than configured.

Note: This issue is very unlikely to happen.

When does this happen:

During Read Modify Write of the MsgTimeoutTimer an interrupt can modify the value of the MsgTimeoutTimer.
The expected value is larger than the actual value, therefore the timer elapses earlier than expected.

In which configuration does this happen:

CanNm_MainFunction and Nm_MainFunction are mapped to different tasks.
The tasks must be able to interrupt each other (preemptive).

Resolution Description:

Workaround:

Keep the CanNm_MainFunction and the Nm_MainFunction in the same task.

Resolution:

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

ESCAN00100425	Channel starts when receiving RepeatMessageBit in BusSleep
Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	1.00.00
Fixed in versions:	9.00.00

Problem Description:

What happens (symptoms):

In BusSleep the channel starts due to a RepeatMessageBit of a received NM message.

Note: This scenario can only occur in a very small time frame at start-up or shut-down.

When does this happen:

- Channel is in BusSleep.
- Received NM message has the RepeatMessageBit set

In which configuration does this happen:

NodeDetection is enabled.

'/MICROSAR/CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmNodeDetectionEnabled'

Resolution Description:

ESCAN00100425 Channel starts when receiving RepeatMessageBit in BusSleep

Workaround:

Reference a user config file to CanIf ('ActiveEcuC/CanIf/CanIfPrivateCfg/CanIfUserConfigFile').

Add the following lines to the file:

```
/* File Start */
#if defined CANIF_LCFG_SOURCE
#define CanNm_RxIndication Workaround_CanNm_RxIndication
#endif

extern void Workaround_CanNm_RxIndication( PduIdType RxPduId, const PduInfoType
*PduInfoPtr );
/* File End */
```

In any application source file, add the following lines:

```
/* Workaround Start */
#include "CanNm.h"
#include "CanNm_Cbk.h"

void Workaround_CanNm_RxIndication( PduIdType RxPduId, const PduInfoType *PduInfoPtr )
{
 Nm_StateType state;
 Nm_ModeType mode;
 NetworkHandleType canNmChannel;

# if ( CANNM_PDUIDTOCHINDEX == STD_ON )
 canNmChannel = (NetworkHandleType)CanNm_GetPduIdToChIndex(RxPduId);
# else
 canNmChannel = (NetworkHandleType)RxPduId;
# endif

 CanNm_GetState( CanNm_GetChannelIdOfChannelConfig(canNmChannel), &state, &mode );

 if( state == NM_STATE_BUS_SLEEP )
 {
 PduInfoPtr->SduDataPtr[CanNm_GetPduCbvPositionOfChannelConfig(canNmChannel)] =
PduInfoPtr->SduDataPtr[CanNm_GetPduCbvPositionOfChannelConfig(canNmChannel)] &
(uint8)0xFEu;
 }

 CanNm_RxIndication( RxPduId, PduInfoPtr );
}

/* Workaround End */
```

Resolution:

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

ESCAN00100672 Possible buffer overflow when XCP Master ignores MaxCto/Dto

Component@Subcomponent:	Cp_XcpOnTcpIpAsr@Implementation
First affected version:	1.00.00
Fixed in versions:	4.00.00

Problem Description:

What happens (symptoms):

When the XCP Master ignores the parameters MaxCto and MaxDto reported by the Xcp Slave and assembles frames which are bigger, a buffer overflow might happen in the slave while copying the received frame to a temporary buffer.

When does this happen:

Always and immediately

In which configuration does this happen:

When the XCP Master ignores the reported MaxCto/Dto parameters and assembles frames which are bigger.

Resolution Description:

Workaround:

This issue will not arise with XCP Masters following the XCP Specification.
Correct rogue Xcp Master to meet XCP Specification by following MaxCto/Dto Parameters.

Resolution:

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

ESCAN00100913 Partial Networks are not activated due to NM Messages are not received

Component@Subcomponent:	Nm_AsrNmUdp@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	4.00.00

Problem Description:

What happens (symptoms):

--

Partial Networks are not activated or kept active as NM messages are not received.
This can have multiple side effects like messages are not sent or received (due to PDU groups are deactivated) or partial networks are not forwarded (PN coordinator use case). This could lead in the end to various error situations in the car like Emergency Run.

Note: Reason is a not correct configuration in the SoAd.

When does this happen:

--

Always at run-time when a NM message shall be received.

In which configuration does this happen:

--

In all configurations where the SoAd connection (/MICROSAR/SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection) for receiving the NM message has configured the remote address (/MICROSAR/SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection/SoAdSocketRemoteAddress) with wildcards.

Note: You can get to this socket connection by checking in UpdNm Rx PDU (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmRxPdu) the usage for the referenced global PDU (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmRxPdu/UdpNmRxPduRef).
This leads you to the socket route that links to the according global PDU and this socket route has also a reference to the according socket connection.

Resolution Description:

Workaround:

--

Change within the SoAd connection group (/MICROSAR/SoAd/SoAdConfig/SoAdSocketConnectionGroup) containing the socket connectino for receiving the NM message the following value:
Set /MICROSAR/SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp/SoAdSocketUdpListenOnly to TRUE

Resolution:

--

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

ESCAN00101027 Stack Usage of Dem functions too high

Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	14.03.00
Fixed in versions:	16.01.00

Problem Description:

What happens (symptoms):

Stack Usage of some functions too high.
Especially the stack usage of following APIs:
- Dem_GetNextFreezeFrameData
- Dem_GetNextExtendedDataRecord

When does this happen:

When using inlining during compilation.

In which configuration does this happen:

Dem/DemGeneral/DemDcmSupport == TRUE

Resolution Description:

Workaround:
----
define the macro DEM_LOCAL_INLINE and set the content to empty as follows:
#define DEM_LOCAL_INLINE

The result of the workaround depends on the used compiler and compiler options.

Resolution:

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

ESCAN00101052	**Coordinated channels connected to a sub-gateway provoke a delayed shutdown for the channels connected to the main gateway**
Component@Subcomponent:	Nm_Asr4NmIf@Implementation
First affected version:	2.00.00
Fixed in versions:	
Problem Description:	

ESCAN00101052 Coordinated channels connected to a sub-gateway provoke a delayed shutdown for the channels connected to the main gateway

What happens (symptoms):

The NM coordinator provokes a delayed shutdown on the channels connected to the main gateway ECU.
As a consequence, the ECUs connected to the main gateway are shut down later than expected.

Also, an unexpected asynchronous NM message (only on CAN or ETH channels) is sent.
This message is sent on the communication channel(s) where your ECU is connected to the main gateway (your ECU is a passive coordinator on that channel).

The coordinated shutdown on your ECU will indicate readiness to sleep later than expected, i.e. the NM messages are transmitted for a longer period of time than expected.
The shutdown itself however will be performed.

When does this happen:

The issue will occur during run-time when the following two events happen at approximately the same time (within 'Main Function Period' of ComM):

1) the active coordinator (typically the main gateway) indicates that it is ready to sleep (the Sleep Ready bit in the NM message is set to 1 for the first time).
This happens on the channel(s) where your ECU (a passive coordinator) is connected to the main gateway (active coordinator).

2) in the Application, the last user requests COMM_NO_COMMUNICATION at ComM so that all coordinated channels request No Communication.

Note:
This will only happen in very rare situations.

In which configuration does this happen:

Your ECU realizes a sub-gateway, i.e. it is a gateway ECU but it is also connected to other gateways

AND

The main gateway ECU is connected on at least one communication channel with your ECU.

AND

'Coordinator Sync Support' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmCoordinatorSyncSupport) is turned ON in the NmGlobalFeatures container in the 'Network Management General' editor / 'Basic Editor' in DaVinci Configurator (-> #define NM_COORD_SYNC_SUPPORT STD_ON can be found in Nm_Cfg.h)

AND

at least one channel that belongs to coordinator cluster (/MICROSAR/Nm/NmChannelConfig/NmCoordClusterIndex is defined and less than 255) is configured as passively coordinated (/MICROSAR/Nm/NmChannelConfig/NmActiveCoordinator = FALSE) (*)

AND

ESCAN00101052	Coordinated channels connected to a sub-gateway provoke a delayed shutdown for the channels connected to the main gateway

the following call order is possible (e.g. all the following functions are in one task container or preemption is possible that this call order may happen):
(
ComM_MainFunction_x()
BusNm_MainFunction(_x) (BusNm stands for CanNm, FrNm, UdpNm, ...) and one of the BusNm channels is passively coordinated (*) see above
Nm_MainFunction()

OR

BusNm_MainFunction(_x) (BusNm stands for CanNm, FrNm, UdpNm, ...) and one of the BusNm channels is passively coordinated (*) see above
ComM_MainFunction_x()
Nm_MainFunction()

OR

Nm_MainFunction()
ComM_MainFunction_x()
BusNm_MainFunction(_x) (BusNm stands for CanNm, FrNm, UdpNm, ...) and one of the BusNm channels is passively coordinated (*) see above

OR

Nm_MainFunction()
BusNm_MainFunction(_x) (BusNm stands for CanNm, FrNm, UdpNm, ...) and one of the BusNm channels is passively coordinated (*) see above
ComM_MainFunction_x()

OR

BusNm_MainFunction(_x) (BusNm stands for CanNm, FrNm, UdpNm, ...) and one of the BusNm channels is passively coordinated (*) see above
Nm_MainFunction()
ComM_MainFunction_x()
)

Resolution Description:

ESCAN00101052 Coordinated channels connected to a sub-gateway provoke a delayed shutdown for the channels connected to the main gateway

Workaround:

Make sure that every ComM_MainFunction_x call is succeeded by one function call of Nm_MainFunction, then succeeded by every BusNm_MainFunction(_x).
This means that only the following call order is allowed:

1. ComM_MainFunction_x
2. Nm_MainFunction
3. BusNm_MainFunction(_x)

This can for instance be accomplished by having all these functions in one task container, having the same main function period and the corresponding order.
See Runtime System, OS Configuration, Tasks in DaVinci Configurator for more details.

Note: BusSM_MainFunction (e.g. CanSM_MainFunction) needs to be called after ComM_MainFunction_x and before BusNm_MainFunction(_x).
Resolution:

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

ESCAN00101125 IpV4_Ip_VJoinMulticastGroup() reports error IPV4_E_INV_PARAM to the Det

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	9.00.00

Problem Description:
What happens (symptoms):

The function IpV4_Ip_VJoinMulticastGroup reports an error IPV4_E_INV_PARAM to the Det.
IPv4 calls EthIf_UpdatePhysAddrFilter in order to leave a multicast group that was never joined before.

The behavior should not have side effects in known use cases.

When does this happen:

When a multicast address is assigned/changed during runtime via one of the following APIs

TcpIp_RequestIpAddrAssignment()
TcpIp_ReleaseIpAddrAssignment()

In which configuration does this happen:

TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpIpAddrReceiveAllNotConfiguredMulticasts == TRUE
TcpIp/TcpIpGeneral/TcpIpDevErrorDetect == TRUE

Resolution Description:
Workaround:

The reported Det error may be ignored.

Resolution:

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

ESCAN00101409 Last written data is not written to NVRAM when dirty flags are used

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.08.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

Last written data may not written to NVRAM when dirty flags are used

When does this happen:

During runtime.

In which configuration does this happen:

- Configured NvBlockDescriptor
- DirtyFlags are used

Resolution Description:

Workaround:

The NvRunnable needs a higher priority so that it can not be interrupted by the runnable that triggers the Rte_Write.

Resolution:

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

ESCAN00101535 LinTp: Missing schedule table request notification if no data provided for functional request

Component@Subcomponent:	If_Asr4IfLin@Implementation
First affected version:	1.00.00
Fixed in versions:	7.00.00

Problem Description:

What happens (symptoms):

When functional request is going to be transmitted (MRF header is ready to be scheduled) and the PduR cannot provide the transmission data,
the connection is correctly aborted, but the BswM is not notified to switch back to applicative schedule table.

When does this happen:

When PduR cannot provide the single frame data for transmission of a functional request.

In which configuration does this happen:

Any.

Resolution Description:

Workaround:

Configure PduR so that it can always provide the six-byte buffer for a functional request to be transmitted. (Store-and-Forward TpThreshol >= 6)

Resolution:

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

ESCAN00101726 No communication after wakeup request during sleep frame transmission

Component@Subcomponent:	If_Asr4IfLin@Implementation
First affected version:	4.01.00
Fixed in versions:	7.00.00, 5.01.80

Problem Description:

What happens (symptoms):

After requesting a LIN channel to wakeup (FULL_COM request), no wakeup frame is transmitted.
The LinIf channel is in OPERATIONAL state and scheduling continues, but no communication happens as the LIN driver is in SLEEP state.

When does this happen:

The LIN channel wakeup request follows directly a sleep request (FULL_COM request directly after NO_COM request) so that the sleep mode frame is currently transmitted when the wakeup request is triggered
and a schedule table is still active or has been requested together with the wakeup request
and the current schedule slot delay is shorter than the frame transmission time of the sleep mode frame.

In which configuration does this happen:

There is at least one schedule table slot that is shorter than the maximum sleep mode frame transmission time.
Maximum sleep mode frame transmission time: Baudrate 19200: 10ms, Baudrate 10417: 17ms, Baudrate 9600: 19ms.

Resolution Description:

Workaround:

When requesting a LIN channel to go to sleep mode (NO_COM request), request a schedule table switch to NULL schedule at the same time.
OR
After requesting go to sleep mode, wait until the LinSM confirms the sleep transition respectively wait a minimum delay of 100ms before requesting FULL_COM again

Resolution:

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

ESCAN00102027	**Wrong interpretation of application and context id on little endian platforms**
Component@Subcomponent:	Monitoring_AsrDlt@Implementation
First affected version:	1.05.00
Fixed in versions:	2.01.00

Problem Description:

What happens (symptoms):

The application id and the context id shall be independent of the platform/payload endianness because they consist of 4 separate bytes, even if they are of type uint32.
The DLT applies the current payload endianness to interpret those ids.
In case of little endian platform, the ids are mixed up and do not fit the registered ids.
Therefore, it is not possible to change the settings of the DLT users (couple of application and context id).

If using the DltViewer or another DLT master tool, the changing of protocol settings for specific DLT users is not working.

When does this happen:

When receiving a control message.

In which configuration does this happen:

If /MICROSAR/Dlt/DltMultipleConfigurationContainer/DltProtocol/DltHeaderPayloadEndianes is set to LittleEndian.

Resolution Description:

Workaround:
----
1. Use a DltUserConfigFile and define the macro Dlt_IsPayloadInBigEndian as TRUE (#define Dlt_IsPayloadInBigEndian TRUE)
2. Or, switch the application id and context id manually in the DLT master tool or in the DLT application.
 -> If your application id should be called "MyAp" change it to "pAyM"

Resolution:

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

ESCAN00102074 Transport protocol routing paths do not consider metadata length for Buffer allocation

Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	9.00.00
Fixed in versions:	13.02.00

Problem Description:

What happens (symptoms):

The metadata length of a queued transport protocol routing path is not taken into account when allocating a TxBuffer.

This may lead to a situation where a too small buffer is used. In this case the source Tp module can't copy its data to the buffer as it is full. The buffer will never be read as the TpThreshold will not be reached and thus no Transmit on the destination is initiated.

The source Tp module will abort the communication if it runs into a timeout.

When does this happen:

Always and immediately if a too small buffer is allocated by the routing path.

In which configuration does this happen:

Only queued transport protocol routing paths (PduRDestPdu/PduRDestPduQueueDepth is set) which have metadata (/EcucPduCollection/Pdu/MetaDataLength is set) configured are affected.

The routing path is using shared buffer (has no assigned buffer; PduRDestPdu/PduRDestTxBufferRef is not set) and there are shared buffer which are smaller than (TpThreshold (PduRDestPdu/PduRTpThreshold) + Metadata-Length).

Resolution Description:

Workaround:

Assign suitable buffers to the routing path. Do not use shared buffers for these routing paths.

Resolution:

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

ESCAN00102254 NvM accepts out of range dataset index.

Component@Subcomponent: MemService_AsrNvM@Implementation

First affected version: 5.02.00

Fixed in versions:

Problem Description:

What happens (symptoms):

NvM accepts invalid/ out of range dataset index and creates and forwards invalid underlying module identifier (base + dataset index).

When does this happen:

If the configured number of datasets (NvM/NvMBlockDescriptor/NvMNvBlockNum) is used as parameter within NvM_SetDataIndex.

e.g. configuring 4 datasets and using the 4 as dataset index to set via SetDataIndex -> 3 is the upper valid boundary, 4 shall not be accepted. 5 will not be accepted.

In which configuration does this happen:

Every configuration using NvM/NvMBlockDescriptor/NvMBlockManagementType == NVM_BLOCK_DATASET.

Resolution Description:

Workaround:

Enable parameter checks in underlying device - Fee and Ea shall check (AUTOSAR) the passed block number -> block number will be calculated via the base block number and the set dataset index -> Fee/ Ea shall detect the invalid block number.

Since NvM does not recognize the last valid dataset index + 1, the user has to ensure the NvM will never be called with this value. All dataset indexes > last valid index + 1 will be recognized as invalid.

Resolution:

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

ESCAN00102294 RTE generator generates wrong trusted setting to the OIL file

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.18.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

Rte.oil contains the wrong TRUSTED setting for the BSW OS application.
The attribute is generated as TRUE although it is FALSE in the input configuration.
For AUTOSAR3, the ECUC synchronization reports different trusted settings.
For AUTOSAR4, the OS configuration in CFG5 is not modified.

When does this happen:

During ECUC Synchronization / OIL file generation.

In which configuration does this happen:

This happens when the BSW OS application is nontrusted and another nontrusted partition accesses a NVBlock
software component.
This issue is only relevant for old operating systems that do not use ECUC based OS configurations.

Resolution Description:

Workaround:

Fix the TRUSTED setting in the configuration that is used for generating the OS.

Resolution:

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

ESCAN00102374	Measurement of RTE internal variables delivers corrupted values
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

Corrupted values are measured.

When does this happen:

When an RTE internal variable is measured (with XCP).

In which configuration does this happen:

This happens when measurement is enabled for an unconnected sender port and when the data element is accessed by multiple tasks that can interrupt each other.

Resolution Description:

Workaround:

Enable "api usage by address" for the port.

Resolution:

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

ESCAN00102561	Eth_30_Core_LL_TimeSync_Init not called any more from Eth_30_Core_TimeSync_Init
Component@Subcomponent:	DrvEth__coreAsr@Implementation
First affected version:	3.04.02
Fixed in versions:	

Problem Description:

What happens (symptoms):

Eth_30_Core_LL_TimeSync_Init function call has been removed from Eth_30_Core_TimeSync_Init

If no derivative specific implementation of Eth_30_Core_LL_TimeSync_Init is present the function call is actually obsolete and there is no error.

If derivative specific implementation in Eth_30_Core_LL_TimeSync_Init is present or needed, the code is missing and the hardware timestamps retrieved by the Ethernet controller might be incorrect. This could lead to incorrect timestamp-information being forwarded to the upper layer.

When does this happen:

At runtime

In which configuration does this happen:

- Eth/EthConfigSet/EthCtrlConfig/EthCtrlEnablePtp Enabled
- Derivative specific code necessary for TimeSync (Eth_30_Core_LL_TimeSync_Init implementation not empty)

Resolution Description:

Workaround:

The PostControllerInit-Callout can be used to perform the call, if it should be necessary

Resolution:

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

ESCAN00102622 CanControllerDefaultBaudrate has no effect

Component@Subcomponent:	DrvCan__baseAsr@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	4.06.01

Problem Description:
What happens (symptoms):

----

configuration of "CanControllerDefaultBaudrate" has no effect. The first baud rate will be used always (index 0).

When does this happen:

Always and immediately

In which configuration does this happen:

CanControllerDefaultBaudrate != first baud rate
AND
CanSetBaudrateApi == true

Resolution Description:
Workaround:

----

setup first baud rate as your default baud rate.

Resolution:

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

ESCAN00102687 Internal behavior is not updated when modifying main function period parameter

Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	13.00.00
Fixed in versions:	15.01.01

Problem Description:
What happens (symptoms):

The internal behavior is not updated. This leads to an incorrect scheduling of the BswM main function, since the old value is still used.

When does this happen:

At configuration time when the Main Function Period is modified.

In which configuration does this happen:

Each configuration.

Resolution Description:
Workaround:

Remove the internal behavior file and restart the configuration.

Resolution:

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

ESCAN00102688 Swc is not rebuilt when modifying main function period parameter

Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	15.01.01

Problem Description:

What happens (symptoms):

The file BswM_swc.arxml is not rebuilt when only modifying the parameter Main Function Period.
Therefore the swc does not contain the newest values for the main function period.

When does this happen:

At configuration time when changing the Main Function Period parameter.

In which configuration does this happen:

Each configuration where the BswM_swc.arxml is necessary.

Resolution Description:

Workaround:

Restart the configuration.

OR

Change a parameter that causes an Swc rebuild, e.g. the creation of an Swc Mode Request Port.

Resolution:

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

ESCAN00102709	**Validator 'UDPNM01017' prohibits a valid configuration**
Component@Subcomponent:	Nm_AsrNmUdp@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	4.00.00

Problem Description:

What happens (symptoms):

The validator "UDPNM01017" prohibits to enable the message acceptance filter.

This validation message is too restrictive, since there are use cases, in which the message acceptance filter is required.

Due to the solving action of the valdiator an erroneous configuration is created. This leads to unaccepted NM messages.

When does this happen:

During configuration.

In which configuration does this happen:

Each configuration in which the MessageAcceptanceFilter is required to be enabled.

Note: This problem might occur if UdpNm transmission is done via Multicast or Unicast.

Resolution Description:

Workaround:

Set the parameter 'SoAdSocketMsgAcceptanceFilterEnabled' to User Defined.

Resolution:

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

ESCAN00102830 RTE does not reset update flags when Com_ReceiveSignal returns COM_SERVICE_NOT_AVAILABLE

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

Rte_IsUpdated returns TRUE instead of FALSE when an IPDU group is stopped before calling Rte_Read after a signal reception.

When does this happen:

During runtime when the following event chain happens:
- COM calls the notification callback
- COM stops the IPDU group
- The application calls Rte_Read
- The application calls Rte_IsUpdated

In which configuration does this happen:

This happens when enable update is configured and when COM and the receiving SWC are mapped to the same partition.

Resolution Description:
Workaround:

Handle the data element as "not updated" as long as Rte_Read returns RTE_E_COM_STOPPED.

Resolution:

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

ESCAN00102895	**GET_CAL_PAGE command is not checking for invalid segment number**
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The command GET_CAL_PAGE is checking for correct mode parameter. It is not checking, if the segment is in range. This should be possible to be performed in the user call-back XcpAppl_GetCalPage. As a result the command is returning a cal page even for segments that are not available.

When does this happen:

Always and immediately

In which configuration does this happen:

When page switching is active:
/MICROSAR/Xcp/XcpCmdConfig/XcpPageSwitching

Resolution Description:

Workaround:

In the function XcpAppl_GetCalPage, use the following application code:

```
 /* #10 Prepare a error response with the error codes as parameter. */
 if (Segment >XCP_MAX_SEGMENT)
 {
 Xcp_ChannelInfo[0].ResponseFrame.b[0] = XCP_PID_ERR;
 Xcp_ChannelInfo[0].ResponseFrame.b[1] = XCP_ERR_SEGMENT_NOT_VALID;
 return 0;
 }
 else
 {
 /* #20 If all parameters are valid, return the active calibration page of the specified segment. */
 return (uint8)XcpAppl_CalPageInfo[Segment].ActiveCalPage;
 }
```

Resolution:

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

ESCAN00102934 Out of bounds read access leads to a MPU violation

Component@Subcomponent: Il_AsrComCfg5@Implementation

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Depending on the signal layout and the memory configuration, the Com tries to read neighboured memory. In case a MPU does not permit read access, this issue leads to a MPU violation.
The issue might occur in one of the following functions:

Com_Signal_ReadSignal_LB_NBITNBYTE
Com_Signal_ReadSignal_LB_SW_NBITNBYTE
Com_Signal_ReadSignal_HB_NBITNBYTE
Com_Signal_ReadSignal_HB_SW_NBITNBYTE

Note: Apart from the MPU violation, there are no side-effects, as the unrelated bits will be masked out in the succeeding processing steps.

When does this happen:

This issue occurs at runtime. This issue will always occur if the PDU with the specific signal is received.

In which configuration does this happen:

- MPU is configured which does not permit read access to neighboured memory
AND
- Rx Signal overlays multiple bytes, but has a gap at the beginning or end to the respective byte boundary.

Resolution Description:

Workaround:
----
1) Permit read access to the neighboured memory address, if possible.

OR

In case, violation is caused by a deferred PDU:
2) add padding to the linker script for Com_RxDefPduBuffer
OR
3) modify the linker script in a way that the variable <Com_RxDefPduBuffer> is not placed at the end of its "region"

Resolution:

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

ESCAN00102935 Out of bound access in configuration array leads to erroneous switch hardware port configuration

Component@Subcomponent:	DrvEthSwitch_Sja1105PQRSAsr@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The configuration of the port slew rate cannot be processed correctly, if one (or more) of the hardware ports are left unconfigured in the DaVinci Configurator. The generated data leads to an out-of-bounds access in the corresponding array which could lead to a wrongfully and unintended configuration of the hardware switch port.

When does this happen:

At runtime.

In which configuration does this happen:

In configurations where not all hardware switch ports are mapped to a EthSwtConfig/EthSwtPort configuration container, e.g.:

EthSwtConfig/EthSwtPort0/EthSwtHardwarePort = ETHSWT_PORT_1
EthSwtConfig/EthSwtPort1/EthSwtHardwarePort = ETHSWT_PORT_2
EthSwtConfig/EthSwtPort2/EthSwtHardwarePort = ETHSWT_PORT_4
EthSwtConfig/EthSwtPort3/EthSwtHardwarePort = ETHSWT_PORT_5

Resolution Description:

Workaround:

If all hardware ports are configured, the issue does not occur. The workaround is thus to configure the unused ports as "Dummys" and set its parameters 'EthSwtPortSlewRate' = EMPTY.

Resolution:

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

ESCAN00102971 When STIM is disabled and SET_DAQ_LIST_MODE command uses STIM as direction no error code is returned.

Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The command SET_DAQ_LIST_MODE has an mode option which can be used to set the direction of an DAQ list.

When STIM feature is disabled and the Master tries to set STIM as direction no error code is returned.

When STIM is disabled this flag is not evaluated. Instead the configured STIM DAQ lists is treated as regular DAQ list and the measured values are sent to the XCP Master.

When does this happen:

When the XCP Master is configured to use STIM mode and tries to set up the DAQ lists.

In which configuration does this happen:

When STIM is disabled

/MICROSAR/Xcp/XcpCmdConfig/XcpDaqAndStim/XcpSynchronousDataStimulation

Resolution Description:

Workaround:

Interpret the resulting error or warning message in the XCP Master about unexpected DTOs as possible missing STIM feature.

Activate the STIM feature in the Slave, if it is required.

Otherwise configure the XCP Master such that no STIM Event is used.

Resolution:

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

ESCAN00102972 Compile Error: EcuM_GetCoreStatus is not defined

Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	10.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Due to an erroneous generated file a compile error is issued.

When does this happen:

At compile time. Root cause at generation time.

In which configuration does this happen:

If the Os module name is not called exactly "Os" in the DaVinci Configurator.

EcuFlex Configuration is chosen.

Resolution Description:

Workaround:

The module name of the Os can be changed in the project settings. Set the module name to "Os" and generate the EcuM.

Note: If no further dependencies to the original name are existing, the name "Os" can be used for the entire generation. If there are additional dependencies, the EcuM can be generated and the name has to be reset to the original one for all other modules to generate.

Resolution:

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

ESCAN00103054	Commands DOWNLOAD and UPLOAD always increment the MTA, even when an error condition occurred.
Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The commands DOWNLOAD and UPLOAD shall increment the MTA if the operation was successful.
In case an error condition occurred (e.g. illegal length as input parameter or busy), the specification allows an repeat of the operation with new input parameters.
The MTA now will always be incremented, such that a repeat of the operation is not possible without setting the MTA again.

When does this happen:

Always and immediately.

In which configuration does this happen:

All configurations.

Resolution Description:

Workaround:

Use a SET_MTA command in the command sequence, should a DOWNLOAD or UPLOAD return with a negative response.

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) Os_CoreGen7@Implementation
ESCAN00089701	BETA version - the BSW module has a feature with BETA state (Executing trusted applications in non privileged mode) Os_CoreGen7@Implementation
ESCAN00091204	BETA version - the Nm module has a feature with BETA state (FEAT-1865) Nm_Asr4NmIf@Implementation
ESCAN00092470	BETA version - the BSW module has a feature with BETA state (FEAT-1454) SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00092471	BETA version - the BSW module has a feature with BETA state (FEAT-1454) SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00092764	BETA version - the BSW module has a feature with BETA state (FEAT-1454) Ccl_Asr4ComMCfg5@Implementation
ESCAN00093797	BETA version - the BSW module has a feature with BETA state (Barriers) Os_CoreGen7@Implementation
ESCAN00093813	BETA version - the BSW module has a feature with BETA state (Fast Trusted Functions) Os_CoreGen7@Implementation
ESCAN00098377	BETA version - the BSW module has a feature with BETA state (FEAT-2721) SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00102003	BETA version - the BSW module has a feature with BETA state (FEAT-2354) If_AsrIfEth@Implementation
ESCAN00102831	BETA version - the BSW module has a feature with BETA state (STORYC-6731) DrvCan_Mpc5700McanLl@Implementation

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:

- 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.

- Memory Protection in trusted applications.

To ensure that only productive code is used verify that:

- 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:

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:

ESCAN00092471 BETA version - the BSW module has a feature with BETA state (FEAT-1454)

Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	8.00.00
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 PNC references for WakeupSources

HINT: Only Ethernet regarding PNCs might be references by WakeupSources, other busses are not supported!

Resolution Description:

ESCAN00092764	BETA version - the BSW module has a feature with BETA state (FEAT-1454)
Component@Subcomponent:	Ccl_Asr4ComMCfg5@Implementation
First affected version:	8.00.00
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.
- FEAT-1454: Configuration of Switch Ports (switchable per PNC)

The above feature/function has following limitations:
- CFG5 do not provide any validations rules. A proper feature configuration has to be ensured manually.
- Use case PB-L is not supported.
- PNCs having at least one /MICROSAR/ComM/ComMConfigSet/ComMPnc/ComMPortGroupsPerPnc require a ComM user.

To ensure that only productive code is used verify that:
- /MICROSAR/ComM/ComMConfigSet/ComMPnc/ComMPortGroupsPerPnc does not exist.

and
- ensure that COMM_WAKEUPENABLEDOFPNC is defined to STD_OFF in ComM_Cfg.h.
 Note: otherwise MSSV fails with error message 'assertion 'COMM_WAKEUPENABLEDOFPNC: "STD_ON"' == '"STD_OFF"' does not hold'.

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:

ESCAN00098377 BETA version - the BSW module has a feature with BETA state (FEAT-2721)

Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	13.00.00
Fixed in versions:	15.01.00

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.
- Configuration of more than one BswMConfig container / MultiPartition Support

Resolution Description:

No workaround available.

ESCAN00102003	**BETA version - the BSW module has a feature with BETA state (FEAT-2354)**
Component@Subcomponent:	If_AsrIfEth@Implementation
First affected version:	7.02.00
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.
- Ethernet firewall interface

To ensure that only productive code is used verify that:
- EthFw isn't contained in the configuration

Resolution Description:

ESCAN00102831 BETA version - the BSW module has a feature with BETA state (STORYC-6731)

Component@Subcomponent:	DrvCan_Mpc5700McanLl@Implementation
First affected version:	5.00.00
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 package is NOT tested for MCAN Revisions less than 3.1.0

To ensure that only productive code is used verify that "Can/CanGeneral/CanMcanRevision" is set to:

"MCAN_REV_310": MCAN Release 3.1.0 (non-compatibility change to previous revisions)
"MCAN_REV_315": MCAN Release 3.1.5 (CAN-FD support with ISO-11898-1 compatibility)

The value "MCAN_REV_315" also includes all subsequent MCAN Revisions for 3.2.x

Resolution Description:

Workaround:

----

not available

Resolution:

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

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

Index

Index

Index

Index

Index

ESCAN00101731	Strapping pin configuration isn't overriden by the driver for all parameters where it might be necessary on TJA1102 DrvTrans_Tja1100EthAsr@GenTool_GeneratorMsr
ESCAN00101798	Auto Configuration Ecu State Handling - Additional modes of a Mode Declaration Group are removed after an update SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00101829	PDUR90005 Exception in PduR generator during Validation encountered Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00101835	Compiler switch is generated erroneously for the Tasking compiler DrvCan_Mpc5700McanLl@GenTool_GeneratorMsr
ESCAN00101870	Null pointer exception when module description for LDCOM is missing Rte_Asr4@GenTool_GeneratorMsr
ESCAN00101880	Compiler error: Compilation fails due to missing NvM.h Rte_Core@Implementation
ESCAN00101955	BswM does not support Mode Request R Ports with Target Variable Data Prototypes and Data Type Mapping Sets SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00101968	Wrong value of define DIAG_ASR4DCM_VERSION Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00102032	OsResources are not configurable for each EcuM Core SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00102127	Compiler error: Linker error: ltc E106: unresolved external Os_CoreGen7@GenTool_GeneratorMsr
ESCAN00102181	DVCfgRteGen terminates unexpectedly Rte_DaVinciBase@Generator
ESCAN00102223	Description of Service IDs in Technical Reference don't match Sercive IDs used in the implementation DrvEth__coreAsr@Doc_TechRef
ESCAN00102235	Compiler error: Use of undeclared identifier TriggerDisableFlags Rte_Core@Implementation
ESCAN00102287	Auto-Configuration Communication Control: No support of J1939TpTxDirectNPdu SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00102325	Compiler error: Client server communication uses undefined trusted function calls Rte_Core@Implementation
ESCAN00102364	Generator error RTE49999 when scaling of child constants fails Rte_Asr4@Generator
ESCAN00102408	Compiler error: no member named Rte_ModeSwitchAck_ Rte_Core@Implementation
ESCAN00102438	Ambiguous description of error handling behavior in OS TechRef. Os_CoreGen7@Doc_TechRef
ESCAN00102445	PduR Mainfunction is not written into the internal behavior in Singlecore configurations Gw_AsrPduRCfg5@GenTool_GeneratorMsr
ESCAN00102467	Size of data computed by comfort view is lower then the actual needed size If_AsrIfFee@GenTool_GeneratorMsr
ESCAN00102477	Compiler error: illegal zero-sized array // undeclared identifier Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00102541	RTE49999: When mode disablings are used over core boundaries and the receiver port is assigned to core 0 Rte_Core@Implementation
ESCAN00102543	Validator limits SPI external buffer length to 256 bytes although more is needed DrvEthSwitch_Sja1105PQRSAsr@GenTool_GeneratorMsr

Index

Index

ESCAN00072881 'Port Access' references the wrong destination

Component@Subcomponent:	EcuAb_AsrIoHwAb@Description
First affected version:	4.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The connection between runnables and assigned port prototypes erroneously references the whole port interface, whereas it should point to a data element inside the very port interface. As a result of this, too many unnecessary consistency checks will be applied

When does this happen:

This happens always and immediately.

In which configuration does this happen:

This happens in all configurations.

Resolution Description:

Workaround:

Transfer of all data elements within one runnable call.

Resolution:

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

ESCAN00073545 Final FBL response not cancelled on protocol preemption

Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.05.00
Fixed in versions:	This ticket is not considered for fixing.

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.

ESCAN00078425 Version Define with wrong value

Component@Subcomponent:	CommonAsr_Tricore@Impl_CompAbstract_Tasking
First affected version:	2.01.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

#define COMPILER_SW_PATCH_VERSION
contains 0 but should be 1

When does this happen:

Always and immediately

In which configuration does this happen:

----

all

Resolution Description:

Workaround:

----

use #define COMMONASR_TRICORE_IMPL_COMPABSTRACT_RELEASE_VERSION
instead

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:	This ticket is not considered for fixing.

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.

ESCAN00086913	**Confusing validation-message: CANTRCV01900 There are unneeded "CanTrcvAccess" containers which can be removed.**
Component@Subcomponent:	DrvTrans_GenericCandioAsr@GenTool_GeneratorMsr
First affected version:	2.02.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

After creating of a "CanTrcvSpiSequenceAccess"-container within the Generic-CAN-transceiver driver following confusing validation-message occurs (e.g.):

CANTRCV01900 Not needed "CanTrcvAccess" containers. (1 message)
CANTRCV01900 There are unneeded "CanTrcvAccess" containers which can be removed.

with solving action:
- Remove all unused "CanTrcvAccess" containers.

When does this happen:

During modification of configuration using the Configurator 5

In which configuration does this happen:

Parameter "CanTrcvHwInterface" == SPI

AND

at least one container "CanTrcv_GenericCan/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiSequence/CanTrcvSpiSequenceAccess" is configured

Resolution Description:

Workaround:

Please do NOT execute the solving action. Just set at least one parameter within each affected "CanTrcvSpiSequenceAccess"-container as "user defined" and ignore the validation-message.
-> Please NOTE: The validation rule does NOT affect the generated data. At all NO SPI-specific generated data is provided by the Generic-CAN-transceiver driver, because it depends largely from the implementation of the CAN-transceiver driver!

Resolution:

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

ESCAN00087264 VTT only: parameter settings from VttEcuC not used (EcuC used instead)

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

Problem Description:

What happens (symptoms):

VTT does not use the settings given in VttEcuC modul like:

- SafeBswChecks (SafeBsw used but should not --> more runtime consumption)
- DummyFunction (compiler warnings)
- DummyStatement (compiler warnings)
- OsType (same for Hardware)

When does this happen:

----

while compile time for DummyFunction and DummyStatement
while runtime for SafeBswChecks

In which configuration does this happen:

VTT used
and
Platform settings in EcuC modul differ from settings in VttEcuC modul

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00087305	Restricted functionality of compiler abstraction
Component@Subcomponent:	If_AsrIfSoAd@Implementation
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The source code does not compile.

When does this happen:

If compiler abstraction is used to optimize the code.

In which configuration does this happen:

In all configurations.

Resolution Description:

Workaround:

Do not use special optimizations in compiler abstraction.

Resolution:

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

ESCAN00087460 SIP migration from MSR4 release 12 to release 14 or higher is not performed

Component@Subcomponent:	Tp_AsrTpTcpIp@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	This ticket is not considered for fixing.

Problem Description:

What happens (symptoms):

The configuration is invalid after updating a configuration based on a MICROSAR4 release 12 delivery to delivery with release 14 or higher.

When does this happen:

During update of the SIP.

In which configuration does this happen:

All configurations.

Resolution Description:

Workaround:

- Before migration edit the TcpIp_bswmd.arxml and change the value of "SW version number" in /MICROSAR/TcpIp_Impl to value 4.00.00.
- Perform the migration
- Set SW version number in bswmd file to original value.

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.

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.

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	**Validiation 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 **Validiation 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 **Validiation 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.

ESCAN00092246 Loading of EthIf generator fails without any EthTrcv

Component@Subcomponent:	If_AsrIfEth@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

When loading a configuration an error dialog will pop up indication that the EthIf generator cannot be loaded. The reason is described in the "Details" window:
Generator EthIf [com.vector.cfg.gen.If_AsrIfEth]: The Initialization of the Generator failed.
Results:
 [Error] ETHIF90110 - Definition Ref is undefined
 - Missing required Definition: DefinitionRef: /[ANY]/EthTrcv/EthTrcvGeneral/EthTrcvGetTransceiverWakeupModeApi
 [Error] ETHIF90110 - Definition Ref is undefined
 - Missing required Definition: DefinitionRef: /[ANY]/EthTrcv/EthTrcvGeneral/EthTrcvWakeUpSupport.

This happens even, if there is no EthIf module available in the ECUC. It is sufficient to have this module in the SIP.

When does this happen:

This happens when loading a configuration.

In which configuration does this happen:

This happens in all configurations when the SIP contains and EthIf, but no instance of any EthTrcv driver.

Resolution Description:

Workaround:

No workaround necessary. It is still possible to generate a configuration without EthIf.
When generating through command line ignore the return code.

Resolution:

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

ESCAN00092372 ETHIF90005 Exception in EthIf generator

Component@Subcomponent:	If_AsrIfEth@GenTool_GeneratorMsr
First affected version:	2.06.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The following error is displayed:

ETHIF90005 Exception in EthIf generator during Validation encountered:
java.lang.NumberFormatException: Not a boolean value : null
/ActiveEcuC/EthIf

When does this happen:

During configuration or validation

In which configuration does this happen:

Configurations where EthTrcv is not fully configured

Resolution Description:

Workaround:

Instantiate the following parameters:

/ActiveEcuC/EthTrcv/EthTrcvGeneral[EthTrcvIndex]
/ActiveEcuC/EthTrcv/EthTrcvGeneral[EthTrcvStartAutoNegotiationApi]
/ActiveEcuC/EthTrcv/EthTrcvGeneral[EthTrcvSetTransceiverModeApi]

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.

ESCAN00092892 Compiler error: function "EcuM_BswErrorHook" has no prototype

Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	2.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler throws the following error:

function "EcuM_BswErrorHook" has no prototype

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 configurations with any PB Modules but EcuM is not configured as PB

AND

The module which uses the API EcuM_BswErrorHook() includes 'EcuM.h' instead of 'EcuM_BswErrorHook.h'.

Resolution Description:

Workaround:

Include the file 'EcuM_Error.h' additional to the include 'EcuM.h', e.g. via a user configuration file.

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 theCONT 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.

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/BswMAction 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. BswMComMModeLimitation
- Configure Communication Control is used once again and Finish is clicked. An option if 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/BswMAction/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.

ESCAN00093413 Auto Configuration Module Initialization - Changed User Include Files always restores

Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	2.00.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

If the EcuM_Init_PBCfg.h entry in the User Config File (/MICROSAR/BswM/BswMGeneral/ BswMUserIncludeFiles/BswMUserIncludeFile) list is overwritten by some other value or being replaced, it is being restored after the Module Configuration Auto Configuration is applied again and the other value might be removed.

When does this happen:

During the configuration with DaVinci Configurator in the BSW Management Editor in the following sequence:

- Configure Module Initialization is clicked
- Finish is clicked
- One of the /MICROSAR/BswM/BswMGeneral/BswMUserIncludeFiles/BswMUserIncludeFile has the value EcuM_Init_PBCfg.h, this one is being changed or deleted.
- Configure Module Initialization is clicked once again
- Finish is clicked
- the dialog 'Manual Adaptions' does not pop up or it pops up but the change is not displayed
- Finish is clicked in the 'Manual Adaptions' dialog if it is displayed

In which configuration does this happen:

Any configuration using the Module Initialization Auto Configurations in BSW Management in DaVinci Configurator

AND

EcuM is configured as Postbuild Loadable or Postbuild Selectable

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.

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.

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:	4.11.00

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.

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.

ESCAN00094765 In VTT Dual Target projects measurement points of type ExecutionTime not supported

Component@Subcomponent: Monitoring_Asr4Rtm@Implementation

First affected version: 3.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Measurement points of type ExecutionTime will report incorrect results.

When does this happen:

When using RTM in a VTT Dual Target project and RTM is used in the Virtual Target.
The RTM generator takes into account the number of interrupts of the real target OS.
The number of interrupts of the virtual target OS are not synchronized and are fixed by the VTT MCAL modules.
Hence usually the number of interrupts for the virtual target differs from the number of interrupts of the real target.
The number of interrupts are needed for the ExecutionTime feature.

In which configuration does this happen:

VTT Dual Target projects with RTM, configured measurement points with type ExecutionTime and with different number of interrupts of virtual and real target.

Resolution Description:

Workaround:

Workarounds:
1. Set all MP types to ResponseTime OR
2. Create dummy ISRs in the real OS
3. Remove ISRs in the VTT OS if possible

Resolution:

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

ESCAN00094810	**"Unexpected Error" in DaVinci Configurator GUI**
Component@Subcomponent:	If_AsrIfFee@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	4.00.05

Problem Description:

What happens (symptoms):

DaVinci Configurator GUI show a message: "An unexpected error has occured. ..."
There is no need to restart DaVinci Configurator.

When does this happen:

It happens, when the "Optimize FEE" assistant was started.

In which configuration does this happen:

It happens if the FEE configuration is inconsistent regarding its definition (bswmd), i.e. (live) validation reported errors in configuration:
AR-ECUC02008 Mandatory parameter <Fee...> is missing in <FeeContainerName>

Resolution Description:

Workaround:

Make sure, that validation does not report any errors before starting the assistant.
If the error message occurred, it is also sufficient to solve validation errors, and then retry.

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.

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 plattform 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.

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.

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.

ESCAN00096007	**IoHwAb - Init Values not configurable for complex data type (e.g Array, structure...)**
Component@Subcomponent:	EcuAb_AsrIoHwAb@GenTool_GeneratorMsr
First affected version:	6.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

No Init-Value for configured complex Data Type could be set.

When does this happen:

----

when a data type is created by the user and he wants to set a Init Value on the port prototype

In which configuration does this happen:

Every configuration with IoHwAb and with Data Types != from base types

Resolution Description:

Workaround:

Currently, for Arrays and Records no initial values can be configured. So a user has to connect a port with a data element referencing an array or a record always with the counterpart in an application within the DaVinci Developer.

Resolution:

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

ESCAN00096243 Compiler warning: WdgM uninitialized variable "aperiodic_local_status_programflow"

Component@Subcomponent: SysService_Asr4WdM@Implementation

First affected version: 5.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

For the compiler the local variable "aperiodic_local_status_programflow" seems to be used uninitialized within the API "BuildEntitiesStatus". Within this function this variable is forwarded as reference to function "CheckProgramFlowViolation". The compiler rises a warning that this variable is possibily uninitialized.

For example:

```
Compiling file: ../../../external/BSW/WdgM/WdgM.c
1 errors, 0 warnings
ctc E507: ["../../../external/BSW/WdgM/WdgM.c" 1317/39] variable
"aperiodic_local_status_programflow" is possibly uninitialized
1 errors, 0 warnings
..\..\MakeSupport\cmd\make: *** [obj/WdgM.o] Error 1
```

This is normally a warning, but it is an error with the compiler option: -Wc--warnings-as-errors=507

When does this happen:

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

Tasking Compiler with version: v6.0r1p2

AND

compiler option: "-Wc--warnings-as-errors=507"

In which configuration does this happen:

Always

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 open item TCVX-42261 - False positive warning "W507 variable is possibly uninitialized" with constant propagation optimization switched off - see release notes https://issues.tasking.com/?project=TCVX&version=v6.0r1p2

Resolution Description:

Workaround:

Remove compiler option "-Wc--warnings-as-errors=507"

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.

ESCAN00097063 Auto-Configuration Communication Control: Tx PDU-Groups are not assigned to a channel and can not be selected

Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	13.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Some Tx PDU-Groups are listed at the end of the communication control view and are marked in grey as not available with the following annotation:

"Channel of corresponding Group PDUs could not be determined"

When does this happen:

Always during execution of the Auto-Configuration Communication Control.

In which configuration does this happen:

In Configurations with at least one Tx Pdu Group, in which all of the Tx Pdus have different global Pdu References in parameters /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu and /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRSrcPdu.

Resolution Description:

Workaround:

Control of Tx PDU-Groups has to be configured with manual BswM Rules.

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_ExpiredWakeups
EcuM_PendingCheckWakeups
EcuM_PendingWakeups

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.

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.

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.

ESCAN00097385 TcpIp calls <Up>_TcpIpEvent() callback twice if TCP listen socket is closed

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The TcpIp calls the upper layer callback <Up>_TcpIpEvent() twice for the same event.

Usually this does not have a negative effect when using SoAd as upper layer.
But in SoAd versions of MSR4-R20 or higher a Det is reported by the SoAd.

When does this happen:

The issue occurs if a TCP listen socket is closed via TcpIp_Close(Abort = FALSE)

In which configuration does this happen:

All configurations using TCP listen sockets.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00097410 Compiler error: error C2065: 'UdpNm_CancelTransmit' : undeclared identifier

Component@Subcomponent:	Nm_AsrNmUdp@Implementation
First affected version:	1.00.00
Fixed in versions:	4.00.00

Problem Description:

What happens (symptoms):

The following error is issued by the compile:

error C2065: 'UdpNm_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 "UdpNm"
AND
- /MICROSAR/PduR/PduRBswModules/PduRCancelTransmit is ON

Resolution Description:

Workaround:

- Provide a User Config file with following content:

```
extern Std_ReturnType UdpNm_CancelTransmit(PduIdType TxPduId);
```

and add it's path to the following parameter in DaVinci Configurator 5

/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmUserConfigFile

- Generate UdpNm component in DaVinci Configurator 5.

Provide the following code in an application file:

```
#include "UdpNm.h"

Std_ReturnType UdpNm_CancelTransmit(PduIdType TxPduId)
{
 return E_OK;
}
```

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:	This ticket is not considered for fixing.

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.

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:

ESCAN00098148 Description for interrupt configuration is missing

Component@Subcomponent:	DrvEth__coreAsr@Doc_TechRef
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Provide a better description on how the interrupt service routine's and IRQ-handler's configuration steps have to be performed.

When does this happen:

While reading the Technical Reference

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.

ESCAN00098260 Erroneous validation message "CanIfMultipleBasicCANTxObjects is not required"

Component@Subcomponent:	If_AsrIfCan@GenTool_GeneratorMsr
First affected version:	4.02.00
Fixed in versions:	4.11.00

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.

ESCAN00098356 EthSwt_30_Sja1105PQRS_GetSwitchManagementIns might return incorrect port index

Component@Subcomponent:	DrvEthSwitch_Sja1105PQRSAsr@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

EthSwt_30_Sja1105PQRS_GetSwitchManagementInstance() currently returns the first switch that is found having a specific Ethernet controller. When using this switch instance to transform the port index for e.g. an cascaded switch instance it will lead to wrong port information in case the two switches are configured different with respect to their hardware port, configuration port mapping.

When does this happen:

The issue is present if there are more than 1 switch used and if not all ports are configured.

In which configuration does this happen:

The issue is present in all configurations.

Resolution Description:

Workaround:

No workaround available.

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.

ESCAN00098775 Error in CANoe due to not updated DLT system variables in XCP configuration

Component@Subcomponent:	Monitoring_AsrDlt@GenTool_GeneratorMsr
First affected version:	1.03.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

An error in CANoe occurs if one of following system variables of DLT are activated:
- Dlt_NonVerboseMessageId
- Dlt_DetErrorCode
- Dlt_VerboseMessageData

Hint:
During the a2l update process warnings are reported due to this variables ("<WARNING> Messgröße 'Dlt_DetErrorCode': Adresse nicht aktualisiert").

When does this happen:

By activation of the variables in the XCP/CCP window of CANoe.

In which configuration does this happen:

The "/MICROSAR/Dlt/DltGeneral/DltComLayer" is set to "DLTCOM"

Resolution Description:

Workaround:

No workaround available.

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.

ESCAN00099124	Compiler error: ctc E208: ["..\..\..\external\bsw\can\Can.h" 1093/1] syntax error – token ";" inserted before "*"
Component@Subcomponent:	DrvCan_TricoreMcanAsr@Implementation
First affected version:	3.02.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compile Error occur:

ctc E208: ["..\..\..\external\bsw\can\Can.h" 1093/1] syntax error – token ";" inserted before "*"
ctc I805: ["..\..\..\external\bsw\can\Can.h" 1093/1] start of current external 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:

In every configuration

Resolution Description:

Workaround:

----

add this define with a user config file:

#define CAN_V_NONE

Resolution:

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

ESCAN00099125 Compiler error/warning: C4013, C2065, C2109, C4244

Component@Subcomponent:	Nm_AsrNmUdp@Implementation
First affected version:	2.00.00
Fixed in versions:	4.00.00

Problem Description:

What happens (symptoms):

The following errors and warnings are issued by the compiler:

```
error C4013: 'UdpNm_VCfgGetActiveWakeupBitEnabled' undefined; assuming extern returning int
error C2065: 'UdpNm_TxPduBuffers' : undeclared identifier
error C2109: subscript requires array or pointer type
error C4013: 'UdpNm_VTransmitPduCyclic' undefined; assuming extern returning int
error C2065: 'UdpNm_NmTxPduId2ChannelId' : undeclared identifier
error C4013: 'PduR_UdpNmRxIndication' undefined; assuming extern returning int
error C4013: 'UdpNm_VCfgIsAllNmMessagesKeepAwake' undefined; assuming extern returning int
warning C4244: 'return' : conversion from 'int' to 'Std_ReturnType', possible loss of data
```

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 where Passive Mode is enabled AND at least one of the following features is enabled:

ActiveWakeupBit Enabled (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmActiveWakeupBitEnabled)
OR
CoordinatorSyncSupport (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmCoordinatorSyncSupport)
OR
(UlImmediateTxSupport (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmUlImmediateTxSupport) AND ComUserDataSupport (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmComUserDataSupport))
OR
PnEnabled (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmPnEnabled)

- PassiveMode path (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmPassiveModeEnabled)

Resolution Description:

Workaround:

As a workaround the listed features shall not be enabled when PassiveMode is enabled:

- ActiveWakeupBit Enabled (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmActiveWakeupBitEnabled)
- CoordinatorSyncSupport (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmCoordinatorSyncSupport)
- UlImmediateTxSupport (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmUlImmediateTxSupport) AND ComUserDataSupport (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmComUserDataSupport)
- PnEnabled (/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmPnEnabled)

ESCAN00099172 Linker Errors: Unresolved external symbol

Component@Subcomponent:	Diag_AsrSwcVDem42@Make
First affected version:	1.00.00
Fixed in versions:	1.01.00

Problem Description:

What happens (symptoms):

Linker throws many errors like:

Rte.obj : error LNK2019: unresolved external symbol
_VDem42_ERRH_Fpl_SG_msg_PMM_DemCallbackEventStatusChanged_CallbackEventUdsStatusCha
referenced in function
Rte_Call_DemMaster_0_CBEventUdsStatusChanged_ERRH_Fpl_SG_msg_PMM_DemCallback
EventStatusChanged_CallbackEventUdsStatusChanged

When does this happen:

At compilation time.

In which configuration does this happen:

In all configurations using vDem42 and having at least one service port of vDem42 connected.

Resolution Description:

Workaround:

Workaround 1:
Correct vDem42 make file "vDem42_rules.mak" by removing '#' from line:
#GENERATED_SOURCE_FILES += $(GENDATA_DIR)\vDem42_Cfg.c.

Workaround 2:
Add the "$(GENDATA_DIR)\vDem42_Cfg.c" file manually to your make support environment.

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 in 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.

ESCAN00099290 Partial Networking is not available if Passive Mode is selected

Component@Subcomponent:	Nm_AsrNmUdp@Implementation
First affected version:	2.01.00
Fixed in versions:	4.00.00

Problem Description:

What happens (symptoms):

The following errors and warnings are issued by the compiler:

```
error C2065: 'UdpNm_TxPduBuffers' : undeclared identifier
error C2109: subscript requires array or pointer type
error C4013: 'PduR_UdpNmRxIndication' undefined; assuming extern returning int
error C4013: 'UdpNm_VCfgIsAllNmMessagesKeepAwake' undefined; assuming extern returning int
```

When does this happen:

During compile time.

In which configuration does this happen:

In configurations where Passive Mode is enabled AND Partial Networking is also enabled.

/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmPassiveModeEnabled
AND
/MICROSAR/UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmPnEnabled

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00099352 ConsistencyRT00002 - Error at validator runtime: CanIfTxBufferSupportValidator

Component@Subcomponent:	If_AsrIfCan@GenTool_GeneratorMsr
First affected version:	4.06.01
Fixed in versions:	4.11.00

Problem Description:

What happens (symptoms):

The following error occurs in CFG5:

```
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.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: java.lang.IllegalArgumentException: The passed instance element (/ActiveEcuC/Can/
CanConfigSet/CN_PB_9dcd9cfb_Tx (DefRef: /MICROSAR/Can_CanoeemuCanoe/Can/CanConfigSet/
CanHardwareObject)) is not unique in the ModelTraverser InstanceTree.
You can only pass elements which are unique in the InstanceTree, or you have to use the
selectxxxAsSubView() methods.
Please enable debug log level to see more details.
We are sorry, but due to this internal error, code generation of /MICROSAR/CanIf, /[ANY]/Can has
to be blocked. As a workaround, you may try to restart DaVinci Configurator. Otherwise, please
call Vector for support.
 /ActiveEcuC/Can
 /ActiveEcuC/CanIf
```

When does this happen:

During configuration

In which configuration does this happen:

The same Can/CanConfigSet/CanHardwareObject is referenced by multiple CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg/CanIfHthIdSymRef

Resolution Description:

Workaround:

Referencing the same CanHardwareObject from multiple CanIfHthCfgs is a mis-configuration - delete the duplicated CanIfHthCfgs and reload the configuration to avoid the exception.

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:	This ticket is not considered for fixing.

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.

ESCAN00099474	**a2l: Parameter MAX_ODT_ENTRY_SIZE_DAQ fixed to 7**
Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	This ticket is not considered for fixing.

Problem Description:

What happens (symptoms):

The a2l parameter MAX_ODT_ENTRY_SIZE_DAQ 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.

ESCAN00099481	FlsPageSize configured to 0 leads to an exception in FEE validator
Component@Subcomponent:	If_AsrIfFee@GenTool_GeneratorMsr
First affected version:	3.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

FEE validator throws an exception (ConsistencyRT00002) and FEE cannot be generated. The user gets an error message that this is an internal implementation error and that vector support shall be contacted.

Full Error Message:

"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.If_AsrIfFee.validation.FeePageSizeCheck Validator-Description:Checks, if the Fls page size for write jobs is an integral multiple of all used sector's Fls page sizes. Further runtime errors of this validator won't be reported in the UI. Ex:java.lang.ArithmeticException: /by zero
We are sorry, but due to this internal error, code generation of /MICROSAR/Fee,/[ANY]/Fls has to be blocked. As a workaround, you may try to restart DaVinci Configurator. Otherwise, please call Vector for support."

When does this happen:

It happens always if the configuration is as described in the following:

In which configuration does this happen:

It happens if /ActiveEcuC/Fls/FlsConfigSet/FlsSectorList/FlsSector/FlsPageSize is by accident configured to 0, which is not a valid configuration.
Some Fls drivers don't allow configuring the FlsPageSize to 0 anyway, the issue is not possible to occur then.

Resolution Description:

Workaround:

Configure FlsPageSize properly (0 is not valid)!

ESCAN00099525 CanTpEnableSynchronousTransmit cannot be used with non MICROSAR Components

Component@Subcomponent:	Tp_Asr4TpCan@Doc_TechRef
First affected version:	3.01.00
Fixed in versions:	3.03.00

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.

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.

ESCAN00099814 Wrong references to CanTp_Cfg.c exist

Component@Subcomponent:	Tp_Asr4TpCan@Doc_TechRef
First affected version:	3.00.00
Fixed in versions:	3.02.00

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.

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.

ESCAN00100067	Validator: IllegalStateException occured when DID Control Mask Size is empty for external control mask
Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	9.02.00
Fixed in versions:	11.04.00

Problem Description:

What happens (symptoms):

Dcm generator validation issues the following message:
[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:UnfilteredInvariantProjectModelView
This is not a configuration problem but an internal implementation error. Please contact Vector for support.
Validator-Class: com.vector.cfg.model.swctemplates.internal.validation.SwcValidation Validator-Name: Dcm Swc description validator (/MICROSAR/Dcm) Validator-Description:com.vector.cfg.model.swctemplates.internal.validation.SwcValidation
Further runtime errors of this validator won't be reported in the UI.
Ex: java.lang.IllegalStateException: Missing size of EXTERNAL control mask of IO-DID: com.vector.cfg.gen.Diag_Asr4DcmCore.cm.didManager.did.impl.DidObjectStaticGeneric@235df359.
We are sorry, but due to this internal error, code generation of all modules has to be blocked. As a workaround, you may try to restart the tool. Otherwise, please call Vector for support.

When does this happen:

When the control mask for an IO DID (/Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidAccess/DcmDspDidControl/DcmDspDidControlMask) is switched from DCM_CONTROLMASK_INTERNAL to DCM_CONTROLMASK_EXTERNAL while the IO DID control mask size (Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidAccess/DcmDspDidControl/DcmDspDidControlMaskSize) is empty.

In which configuration does this happen:

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

Resolution Description:

Workaround:

Safe the configuration, close the DaVinci Developer and reopen the configuration.
Hint: Set first the DID Control Mask Size to the correct value before switching the DID Control Mask to DCM_CONTROLMASK_EXTERNAL.

ESCAN00100161 Service 0x2C: Unnecessary RAM usage

Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.02.00
Fixed in versions:	11.00.00

Problem Description:

What happens (symptoms):

The RAM usage is significantly increased.

When does this happen:

Always and immediately

In which configuration does this happen:

- Service 0x2C is supported (in Dcm_Cfg.h: #define DCM_SVC_2C_SUPPORT_ENABLED == STD_ON)

AND

- Subfunction 0x01 of service 0x2C is suppported (in Dcm_Cfg.h: #define DCM_SVC_2C_01_SUPPORT_ENABLED = STD_ON)

AND

- Subfunction 0x02 of service 0x2C is suppported (in Dcm_Cfg.h: #define DCM_SVC_2C_02_SUPPORT_ENABLED = STD_ON)

AND

- At least one DDDID in range 0xF200 to 0xF3FF is configured

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00100457	Compiler error: Missing parameter in function Dem_DcmGetDTCOfOBDFreezeFrame
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	7.02.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compilation error due to wrong prototype of DEM API "Dem_DcmGetDTCOfOBDFreezeFrame".

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:

- Non-MICROSAR DEM is used

AND

- DEM APIs for AR 4.3.0 are enabled (in Dcm_Cfg.h: #define DCM_DEM_API_430_ENABLED == STD_ON)

AND

- Service 0x02 is supported (in Dcm_Cfg.h: #define DCM_SVC_02_SUPPORT_ENABLED == STD_ON)

Resolution Description:

ESCAN00100457 Compiler error: Missing parameter in function Dem_DcmGetDTCOfOBDFreezeFrame

Workaround:

Use a user-configuration file to override and adapt the API to match the expected prototype.

1) Create a user-configuration file (e.g. dcm.cfg). Refer to the DCM technical reference on how to use such a file.
2) Add the following content to it:

```
/* WA_ESCAN00100457_BEGIN: */
#if defined (DCM_SOURCE)
extern Std_ReturnType Dem_DcmGetDTCOfOBDFreezeFrame_Wrapper(uint8 FrameNumber,
uint32* DTC);
# define Dem_DcmGetDTCOfOBDFreezeFrame Dem_DcmGetDTCOfOBDFreezeFrame_Wrapper
#endif
/* WA_ESCAN00100457_END: */
```

3) In an application C-source file (e.g. Appl<SourceFile>.c) add following code:

```
#include "Dcm.h"

/* WA_ESCAN00100457_BEGIN: */
/**
 * (AR <= 4.2.1) to (AR >= 4.2.2) wrapper Dem_DcmGetDTCOfOBDFreezeFrame_Wrapper
 */
Std_ReturnType Dem_DcmGetDTCOfOBDFreezeFrame_Wrapper(uint8 FrameNumber, uint32* DTC)
{
 return DcmGetDTCOfOBDFreezeFrame(FrameNumber, DTC, );
}
/* WA_ESCAN00100457_END: */
```

Resolution:

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

ESCAN00100613 Compiler error: Declaration of CanIf_ControllerModeIndication() is missing

Component@Subcomponent:	If_AsrIfCan@Implementation
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compile error "CanIf_ControllerModeIndication is not declared" 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:

When using a Can Driver, which does not include CanIf.h.
Doesn't occur with Vector Can Driver.

Resolution Description:

Workaround:

Include CanIf.h or copy declaration from CanIf_ControllerModeIndication() into any file, which is included by the Can.c

Resolution:

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

ESCAN00100659 Template functionality can not reference items of 3rd Party MCAL implementations

Component@Subcomponent:	EcuAb_AsrIoHwAb@Description
First affected version:	4.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

IoHwAbSRUsedMcalEntity in IoHwAbSRPortPrototype can not be referenced / chosen if it is a 3rd party MCAL item

When does this happen:

----

always and immediately

In which configuration does this happen:

----

when template functionality is wanted to be used and a 3rd party MCAL item needs to be referenced

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00100716	Compiler error: identifier ComM_UserHandleArrayType_<channel_name> is undefined
Component@Subcomponent:	Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr
First affected version:	3.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Typical compiler error explanations may be:
\Appl\GenData\ComM_Lcfg.c(452) : error: identifier
"ComM_UserHandleArrayType_<chanel_name>" is undefined COMM_LOCAL
VAR(ComM_UserHandleArrayType_<chanel_name>, COMM_VAR_NOINIT)
ComM_FullComRequesters_<chanel_name>;
\Appl\GenData\ComM_Lcfg.c(646) : error: function
Rte_Write_ComM_CR_<chanel_name>_fullComRequestors declared implicitly
(void)Rte_Write_ComM_CR_<chanel_name>_fullComRequestors(&ComM_FullComRequesters_CR_<

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/ComM/ComMGeneral/ComMPncSupport is enabled
AND
at least one ComM channel with
- /MICROSAR/ComM/ComMConfigSet/ComMChannel/ComMFullCommRequestNotificationEnabled is enabled
AND
- all ComM users mapped to this channels are also mapped to at least one Partial Network.

Resolution Description:

Workaround:
----
1) If optional ComM_CurrentChannelRequest functionality is not needed, disable it by setting /ComM/ComMConfigSet/ComMChannel/ComMFullCommRequestNotificationEnabled to disabled.

otherwise

2) Create a new ComM user and map it to the channel that has /MICROSAR/ComM/ComMConfigSet/ComMChannel/ComMFullCommRequestNotificationEnabled enabled.
Notes:
- this user has to be a "channel-only" user, do not map it to any Partial Network.
- ComM_CurrentChannelRequest functionality only considers "channel-only" users, refer to the Technical Reference chapter "Sender Receiver Interface" -> "ComM_CurrentChannelRequest".

Resolution:

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

ESCAN00100736 MSSV terminates unexpectedly if header files are included multiple times

Component@Subcomponent:	Elisa__core@Application
First affected version:	1.08.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

MSSV terminates unexpectedly.

When does this happen:

When a header file is included multiple times.

In which configuration does this happen:

Any Configuration where header files are included multiple times, resulting in an extremely large size of the code.
In the configuration where the problem was found, MemMap.h included Os_MemMap.h

Resolution Description:

Workaround:

Use a define (e.g. Vector_MSSV_Test) to switch off the content of MemMap.h.
Set this define when running MSSV, so that the code is deactivated for the analysis.

ESCAN00100751	Generation aborted due to not executed synchronization of swc description
Component@Subcomponent:	Monitoring_AsrDlt@GenTool_GeneratorMsr
First affected version:	1.03.00
Fixed in versions:	3.00.00

Problem Description:

What happens (symptoms):

When the configuration parameter "/MICROSAR/Dlt/DltGeneral/DltMainFunctionCycleTime" is changed, the expected "Full Swc description build from the Ecu configuration"-solving action does not appear.

Instead nothing happens. When the generation is started, the RTE fails to calculate, validate and generate.

The error RTE13023 occurs.

Usually the error message with solving action should be reported: DLT94990.

When does this happen:

Generation is started.

In which configuration does this happen:

All configurations.

Resolution Description:

Workaround:

----

1. Close the DaVinci Configurator 5
2. Remove the file Dlt_swc.arxml (usually in .\Config\ServiceComponents)
3. Start the DaVinci Configurator 5 project
4. Execute the solving action of Error "DLT94990" to make a "Full Swc description build from the Ecu configuration"
5. Execute solving actions of up-coming errors (maybe another restart of DaVinci Configurator 5 is required).

Resolution:

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

ESCAN00100796 Compiler error: Invalid assignment for ModeSwitchEventMask0

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

Compilation fails because Rte*.c contains following code that does not compile

ModeSwitchEventMask${ID} |= ;

or

ModeSwitchEventMask${ID} is assigned with |= although the value was not initialized before

or

Rte_Start uses Rte_ModeEntryEventMask variables that do not exist

```
 (void)SetEvent(Task,Rte_ModeEntryEventMask_ID[Mode]); /* PRQA S 3417 */ /* MD_Rte_Os */
```

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 all of the following conditions evaluate to true:
- the configuration contains runnables with on entry, on exit or on transition mode switch triggers
- the configuration contains multiple partitions
- the mode switch triggered runnables are mapped to a different partition than the mode provider
- no mode disablings are configured or different cores use different mode mode switch trigger types

Resolution Description:

Workaround:

Configure a runnable with mode disabling dependency.
Use the same trigger types (entry, exit, transition) for all cores

Resolution:

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

ESCAN00100799 Compiler error: Memset used with float initializer in RTE Analyzer stubs

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.18.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

Analysis with RTE Analyzer fails with

error: implicit conversion from 'float' to 'int' changes value from XXXX to 0

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 all of the following conditions evaluate to true:

- the configuration contains implicit inter runnable variables
- the data type is a float data type
- the initializer is different from 0

Resolution Description:

Workaround:

Adapt the analyzer stubs to call memset with 0 as initializer value.
The stubs are not part of the generated code that will be executed in the ECU.

Resolution:

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

ESCAN00100828 Null pointer exception when component for a data mapping is missing

Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	4.20.95, 4.20.00

Problem Description:

What happens (symptoms):

Generation fails with a null pointer exception.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the configuration contains data mappings for not existing SWCs.

Resolution Description:

Workaround:

Remove data mappings for all removed components.

Resolution:

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

ESCAN00100931	**Null pointer exception when ISignal has no data def props**
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.08.00
Fixed in versions:	4.20.95, 4.20.00

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 ISignals do not reference data def props.

Resolution Description:

Workaround:

Add empty data def props to the ISignals

```
<I-SIGNAL>
 <SHORT-NAME>SG04</SHORT-NAME>
 <LENGTH>32</LENGTH>
 <NETWORK-REPRESENTATION-PROPS>
 <SW-DATA-DEF-PROPS-VARIANTS>
 <SW-DATA-DEF-PROPS-CONDITIONAL>
 </SW-DATA-DEF-PROPS-CONDITIONAL>
 </SW-DATA-DEF-PROPS-VARIANTS>
 </NETWORK-REPRESENTATION-PROPS>
 <SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL">/Communication/SystemSignals/SG04</
SYSTEM-SIGNAL-REF>
</I-SIGNAL>
```

Resolution:

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

ESCAN00100981 Compiler error: Unknown Identifier Dcm_ExternalSetNegResponse/ Dcm_ExternalProcessingDone

Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	11.00.00

Problem Description:

What happens (symptoms):

Compiler error: Unknown Identifier Dcm_ExternalSetNegResponse/Dcm_ExternalProcessingDone

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:

- No service is implemented externally

AND

- One or more sub-services are implemented externally

Resolution Description:

Workaround:

Use a user-configuration file to manually change the switch DCM_DIAG_EXTERN_SVC_HANDLING_ENABLED to STD_ON.

1) Create a user-configuration file (e.g. dcm.cfg). Refer to the DCM technical reference on how to use such a file.
2) Add the following content to it:

```
/* WA_ESCAN00100981_BEGIN: */
#if defined (DCM_DIAG_EXTERN_SVC_HANDLING_ENABLED)
# undef DCM_DIAG_EXTERN_SVC_HANDLING_ENABLED
# define DCM_DIAG_EXTERN_SVC_HANDLING_ENABLED STD_ON
#endif
/* WA_ESCAN00100981_END: */
```

Resolution:

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

ESCAN00100985 Compiler error: Undefined symbols XcpConf_XcpEventChannel_Dlt_EvtDet and similar

Component@Subcomponent:	Monitoring_AsrDlt@GenTool_GeneratorMsr
First affected version:	2.01.03
Fixed in versions:	3.00.00

Problem Description:

What happens (symptoms):

The compiler complains about undefined symbols
XcpConf_XcpEventChannel_Dlt_EvtDet
XcpConf_XcpEventChannel_Dlt_EvtDem
XcpConf_XcpEventChannel_Dlt_EvtMsg

If verbose messages are supported: XcpConf_XcpEventChannel_Dlt_EvtVMsg

When does this happen:

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

In which configuration does this happen:

Each configuration, in which DltOverXcp is used, i.e. /MICROSAR/Dlt/DltGeneral/DltComLayer is set to XCP AND
the XCP version is used where the container /MICROSAR/Xcp/XcpCmdConfig exists.

Resolution Description:

Workaround:

Add the following XCP event channels ():
- Dlt_EvtDet
- Dlt_EvtDem
- Dlt_EvtMsg
- Dlt_EvtVMsg (optionally required if verbose mode is enabled; doesn't harm if always added)

Resolution:

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

ESCAN00101005	Compiler error: "undeclared identifier EcuM_GlobalPBConfig_Ptr"
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	9.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The compiler throws the following error:

undeclared identifier "EcuM_GlobalPBConfig_Ptr"

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 EcuM is configured as PB-S

AND

an MCAL module is configured which do only support PB-L

AND

This MCAL module is initialized via an EcuM Driver Init List.

Resolution Description:

Workaround:

Don't use the parameter EcuMModuleService in the container EcuMDriverInitItem. Instead use the parameter AdditionalInitCode and pass the correct configuration pointer.

e.g.
Mcu_Init(&<ConfigPtrName>)

Resolution:

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

ESCAN00101074	Compiler error: Missing Rte_MemClr when LdCom TP is used
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

Compilation fails because the RTE calls a method Rte_MemClr 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 when the LDCOM TP API is configured and
- no timeout flags
- no update flags
- no never received flags
- no acknowledgement
are configured.

Resolution Description:

Workaround:

Configure any other feature that require Rte_MemClr, e.g.
- Tx Acknowledge Handling
- NeverReceived Handling
- IsUpdated Handling
- Mapped Client/Server communication
- Queued Sender/Receiver communication

Resolution:

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

ESCAN00101078 MSSV does not consider include paths not containing header files

Component@Subcomponent:	Elisa__core@Application
First affected version:	1.08.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

MSSV exits with a fatal error because an included file cannot be found in the include paths.

When does this happen:

When executing MSSV with included files (e.g. ".inc"-files) in directories which don't contain header files.

Resolution Description:

Workaround:

Put a dummy header file in included paths if needed.

Resolution:

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

ESCAN00101086 IndexOutOfBoundsException during Calculation

Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.19.00
Fixed in versions:	4.20.95, 4.20.00

Problem Description:

What happens (symptoms):

- When generating the RTE, a java.lang.IndexOutOfBoundException: Index: 0, Size: 0 is thrown.

-> Rte cannot be generated

When does this happen:

- Generation or validation of the RTE

In which configuration does this happen:

- projects with variants

Resolution Description:

Workaround:

- Deleting RteCache files, if the current checksum != ""
- Disable RteCache

Resolution:

- fixed in RteFileCache class getChecksum, by checking the return value of reading the checksum file

ESCAN00101123	Reset TCP SockStateNext after sending the ACK in state CloseWait
Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	1.00.00
Fixed in versions:	9.00.00

Problem Description:

What happens (symptoms):

TCP SockStateNext next is not reset to TCPIP_TCP_SOCK_STATE_INVALID after sending the ACK in state CloseWait.
This behaviour was detected during a code analysis for decision coverage improvement. No erroneous behaviour was detected!

When does this happen:

During the closure of a TCP connection.

In which configuration does this happen:

In all configurations using TCP

Resolution Description:

Workaround:

No workaround available nor necessary.

Resolution:

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

Error is resolved in file "TcpIp_Tcp.c" function "TcpIp_Tcp_VMainStateHandling".

ESCAN00101164 Compiler error: PduR_LoIfTxConfirmation undefined

Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	14.02.00
Fixed in versions:	15.03.00

Problem Description:

What happens (symptoms):

Compiler error "PduR_LoIfTxConfirmation undefined" 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:

- only precompile configuration
- At least one module which has /MICROSAR/PduR/PduRBswModules/PduRTxConfirmation activated
- No communication interface routing path with /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRTransmissionConfirmation activated

Resolution Description:

Workaround:

Configure a dummy communication interface routing path and activate /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRTransmissionConfirmation.

Resolution:

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

ESCAN00101183	**Compiler error: Missing parentheses for Rte_IrvIRead_ and Rte_IrvIWriteRef macros**
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

Compilation fails with an error message
"error: invalid type argument of `->'" or similar.

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 implicit inter-runnable variables of record or primitive type are configured and a Rte_IrvIWriteRef or Rte_IrvIRead_ macro is used directly in an expression:

E.g.
When one element of a record type shall be written.

```
Rte_IrvIWriteRef_a()->element = 0;
```

the compiler is not able to resolve the element because missing parentheses around the address operator let the compiler evaluate the element selection operator first.

```
#define Rte_IrvIWriteRef_a() \
 &(Rte_r)
```

Resolution Description:

Workaround:

Enapsulate the RTE API call in parentheses.
E.g.

```
(Rte_IrvIWriteRef_a())->element = 0;
```

Resolution:

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

ESCAN00101286 Missing MemMap functionality results in unexpected linkage

Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	2.02.00
Fixed in versions:	13.02.00

Problem Description:

What happens (symptoms):

Code is not linked as expected due to Local Function Prototypes missing MemMap functionality.

When does this happen:

Always and immediately

In which configuration does this happen:

Every configuration

Resolution Description:

Workaround:

Manually set the compiler so that all the code is mapped to a different memory section.

Resolution:

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

ESCAN00101345 Service 0x86: Consistency error due to a missing reference to DcmDsdServiceTable

Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	5.01.00
Fixed in versions:	11.04.00

Problem Description:

What happens (symptoms):

Consistency error is issued from the DaVinci Configurator due to a thrown exception.

When does this happen:

Once service 0x86 (response on event) is configured in a service table (DcmDsdServiceTable).

In which configuration does this happen:

- Service 0x86 is configured in DcmDsdServiceTable

AND

- DcmDsdServiceTable is not referenced by DcmDslProtocolRow

Resolution Description:

Workaround:

To prevent the appearance of the consistency error on the configurator GUI, configure DcmDslProtocolRow before configuring service 0x86 in DcmDsdServiceTable.

Or,

If the consistency error is already issued, configure DcmDslProtocolRow, save, close the project, and loaded again.

ESCAN00101354 Service 0x86: Consistency error due to an invalid DcmDsdSidTabServiceDispatcher value

Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	10.02.00
Fixed in versions:	11.04.00

Problem Description:

What happens (symptoms):

Consistency error is issued from the DaVinci Configurator due to a thrown exception.

When does this happen:

Once service 0x86 (response on event) is configured in a service table (DcmDsdServiceTable).

In which configuration does this happen:

- Service 0x86 is configured in DcmDsdServiceTable

AND

- 0x86 is configured as external service

AND

- The parameter DcmDsdSidTabServiceDispatcher is set to "false"

Resolution Description:

Workaround:

If the parameter DcmDsdSidTabServiceDispatcher is set to "false", then delete it.

ESCAN00101362	**Compiler error: C4305 - truncation from 'unsigned int' to PduR<IndexConstArray>Type**
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	14.01.00
Fixed in versions:	15.03.00

Problem Description:

What happens (symptoms):

Compile error C4305: truncation from 'unsigned int' to PduR_<IndexConstArray>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:

- EModuleConfigurationVariant = VARIANT_POST_BUILD_LOADABLE
- IndexConstArray with datatype EComStackDataTypeRep.PDUIDTYPE and invalid elements
- PduIdType = Uint8

Resolution Description:

Workaround:

Configure EcuC/EcucPduCollection/PduIdTypeEnum to UINT16.

Resolution:

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

ESCAN00101401	Generation error due to multiple Tx routing paths with same LinTpTxNSdu from different modules
Component@Subcomponent:	Tp_Asr4TpLin@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	6.00.01, 5.00.04

Problem Description:

What happens (symptoms):

Validator error LINTP01018 is thrown during generation:
LINTP01018: Corresponding RxNSdu for <TxNSdu> could not be retrieved
Generation is aborted.

When does this happen:

During generation.
The issue happens sporadically (50% ok, 50% failed) as the problem depends on internal DaVinci Configurator data model and in which order it provides the routing paths in the model access.

In which configuration does this happen:

There is more than one TxNSdu with the same NAD on the same LIN channel configured
AND
there are several PduR routing paths that have such TxNSdu as destination Pdu
AND
these PduR routing paths have different upper layer modules.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101415	RTE49999: When NvBlock SWCs are used and NvM_MainFunction is missing
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.01.00
Fixed in versions:	1.20.00, 1.20.95

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 the configuration contains NvBlock SWCs but does not contain a NvM_MainFunction runnable.

Resolution Description:

Workaround:

Configure a NvM_MainFunction runnable.

Resolution:

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

ESCAN00101416 Warning RTE1070: Missing parameter definition when NVM configuration uses the standard definition

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.12.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

RTE generator issues a warning
RTE1070: Missing parameter definition

When does this happen:

During generation.

In which configuration does this happen:

This happens when a 3rd party NVM is used.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101463	Application (CFG5) freezes when entering a large negative value for ComSignalLength
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	16.01.00

Problem Description:

What happens (symptoms):

Application (CFG5) freezes after editing signal length (enter a value less than Integer.MIN, e.g. -2147483649)

When does this happen:

During configuration time

In which configuration does this happen:

ComSignalType == UINT8_N
and
ComSignalLength < Integer.MIN

Resolution Description:

Workaround:

Enter valid value according to the BSMWD parameter description and restart the CFG5.

Resolution:

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

ESCAN00101493 Linker error: Unresolved external symbol PduR_SchM_<Enter|Exit>_PduR_PDUR_EXCLUSIVE_AREA_<X>

Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	14.00.00
Fixed in versions:	15.03.00

Problem Description:

What happens (symptoms):

The linker issues an error that the external symbol PduR_SchM_<Enter|Exit>_PduR_PDUR_EXCLUSIVE_AREA_<X> cannot be resolved.

When does this happen:

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

In which configuration does this happen:

Any configuration where more than one /PduR/PduRRoutingTables/PduRLock/PduRExclusiveArea containers are configured in a single-core project.

Resolution Description:

Workaround:

Configure one single 1 /PduR/PduRRoutingTables/PduRLock/PduRExclusiveArea in a single-core project.

Resolution:

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

ESCAN00101509 Compiler error: Unconnected Rte_Invalidate

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

RTE generator generates an Rte_Invalidate macro that does not compile
#define Rte_Invalidate_id() (Rte_id = , ((Std_ReturnType)RTE_E_OK))

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 a sender-receiver p-port and when the following conditions evaluate to true:
- the port is not connected
- measurement access is configured to read-write
- invalidation is enabled
- no init value is configured

Resolution Description:

Workaround:

Connect the port and configure an initial value.

Resolution:

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

ESCAN00101534	LinTp: Available Rx buffer queried again after PduR indicated an error
Component@Subcomponent:	If_Asr4IfLin@Implementation
First affected version:	1.01.01
Fixed in versions:	7.00.00

Problem Description:
What happens (symptoms):

The LinIf queries PduR the current amount of available reception buffer via PduR_LinTpCopyRxData() after a previous call of PduR_LinTpCopyRxData has returned BUFREQ_E_NOT_OK.

When does this happen:

A previous call to PduR_LinTpCopyRxData() for the Rx connection has reported too less size to store the next frame, and the subsequent call to PduR_LinTpCopyRxData() to query the current amount of available buffer (with LinTp_PduInfoPtr.SduLength = 0) is returned with BUFREQ_E_NOT_OK.
Then the LinIf will call PduR_LinTpCopyRxData() again to query the current amount of available buffer until the reception is finished.

In which configuration does this happen:

Any.

=====
NOTE:
This is no issue when using MICROSAR PduR because the MICROSAR PduR always returns BUFREQ_OK for PduR_LinTpCopyRxData() when requested with LinTp_PduInfoPtr.SduLength = 0.
=====

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00101549 RTE49999: When a runnable is disabled by multiple mode machines and when one of the mode machines is not connected

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.00, 1.20.95

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 a trigger of a runnable is disabled by two mode disablings and when one of the mode r-ports is connected to an active mode provider and the other mode r-port is not connected.

Resolution Description:

Workaround:

Connect the unconnected mode r-port to a mode p-port and configure a mode switch point.

Resolution:

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

ESCAN00101552 Port Interfaces StateRequest and CurrentMode are documented as only available for EcuM fixed

Component@Subcomponent:	SysService_Asr4EcuM@Doc_TechRef
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

According to the Technical Reference the Port Interfaces StateRequest and CurrentMode are only available for EcuM with fixed behavior.

But in fact both are available also for EcuM Flex.

When does this happen:

While reading the TechRef.

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.

ESCAN00101557	RTE49999: When mode disablings are used in a runnable with multiple triggers
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.95, 1.20.00

Problem Description:

What happens (symptoms):

RTE generator aborts with a RTE49999 error when mode disablings are used over core boundaries in a runnable with multiple triggers

When does this happen:

During generation.

In which configuration does this happen:

- Cross core mode communication.
- Runnable with multiple triggers and at least one of them has mode disablings.

Resolution Description:

Workaround:

Create a new runnable with one trigger that contains mode disablings.

Resolution:

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

ESCAN00101590	Switch management can't be enabled although an EthSwt is present in the configuration
Component@Subcomponent:	SysService_AsrTSynEth@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	8.00.00

Problem Description:

What happens (symptoms):

The parameter /MICROSAR/EthTSyn/EthTSynGeneral/EthTSynEnableSwitchMgmt is automatically switches to 'false' altough an EthSwt is present

When does this happen:

At configuration time

In which configuration does this happen:

EthSwt is present in the configuration

Resolution Description:

Workaround:

At /MICROSAR/EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynSwitchManagementEthSwitchPortRef for at least one EthTSyn port

Resolution:

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

ESCAN00101662	Description of NvM write frequency is incorrect for API call Dem_RequestNvSynchronization()
Component@Subcomponent:	Diag_Asr4Dem@Doc_TechRef
First affected version:	4.00.00
Fixed in versions:	16.02.00

Problem Description:

What happens (symptoms):

In chapter "NVRAM Write Frequency" following description is missing in table:
"Depending on configuration calling API Dem_RequestNvSynchronization() will always trigger a NVRAM write for Admin Data and Debounce Data even if content has not changed."

When does this happen:

Always.

In which configuration does this happen:

Admin Data is always written for configuration:
/Dem/DemGeneral/DemNvSynchronizeSupport == TRUE
AND
/Dem/DemGeneral/DemUseNvm == TRUE

Debounce Data is always written for configuration:
/Dem/DemGeneral/DemNvSynchronizeSupport == TRUE
AND
/Dem/DemGeneral/DemUseNvm == TRUE
AND
at least one /Dem/DemConfigSet/DemEventParameter/DemEventClass/
DemDebounceAlgorithmClass/DemDebounceCounterBased/DemDebounceCounterStorage == TRUE

Resolution Description:

Workaround:

The correct information is:
Depending on configuration calling API Dem_RequestNvSynchronization() will always trigger a NVRAM write for Admin Data and Debounce Data even if content has not changed.

ESCAN00101673 Compiler error: Undefined symbol ECUM_LOCAL

Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	10.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler complains about undefined symbol ECUM_LOCAL. Compilation process finishes with errors.

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:

'ECUM_NUMBER_OF_CORES > 1'

Resolution Description:

Workaround:

Add the following lines to 'Compiler_Cfg.h':

```
#if !defined (ECUM_LOCAL)
# define ECUM_LOCAL static
#endif
```

Resolution:

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

ESCAN00101681	Service 0x2A: Generator issue due to inconsistent periodic transmission rates
Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	8.04.00
Fixed in versions:	11.04.00

Problem Description:

What happens (symptoms):

DCM generator issues the following message:

DCM90001 Fatal Generator (MICROSAR Dcm Generator) failure (1 message)
 DCM90001 Generation phase has reported an error. The file generation was aborted.
DCM90005 Generator (MICROSAR Dcm Generator) failure, because of an exception (1 message)
 DCM90005 Exception in Dcm generator during Generation encountered:
java.lang.IllegalArgumentException: Time constant (<value>) is too small for the selected task time (<value>)! /ActiveEcuC/Dcm

When does this happen:

During generation.

In which configuration does this happen:

- Service 0x2A is supported (in Dcm_Cfg.h: #define DCM_SVC_2A_SUPPORT_ENABLED == STD_ON)
AND
- Parameter "/Dcm/DcmConfigSet/DcmGeneral/DcmTaskTime" is greater than parameter ("/Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission/"):
 - DcmDspPeriodicTransmissionSlowRate
 OR
 - DcmDspPeriodicTransmissionMediumRate
 OR
 - DcmDspPeriodicTransmissionFastRate
 OR
 - DcmDspPeriodicTransmissionDelaySlowRate
 OR
 - DcmDspPeriodicTransmissionDelayMediumRate
 OR
 - DcmDspPeriodicTransmissionDelayFastRate

Resolution Description:

ESCAN00101681 Service 0x2A: Generator issue due to inconsistent periodic transmission rates

Workaround:

Set the parameters "/Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission/":
- DcmDspPeriodicTransmissionSlowRate
- DcmDspPeriodicTransmissionMediumRate
- DcmDspPeriodicTransmissionFastRate
- DcmDspPeriodicTransmissionDelaySlowRate
- DcmDspPeriodicTransmissionDelayMediumRate
- DcmDspPeriodicTransmissionDelayFastRate

to a value greater or equal to /MICROSAR/Dcm/DcmConfigSet/DcmGeneral/DcmTaskTime.

Resolution:

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

ESCAN00101701 Compiler error: can't take address of object when ServerArgumentImplPolicy UseArrayBaseType is configured for non array types

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.05.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:
What happens (symptoms):

The RTE generator generates code with duplicated & operators
e.g.

ClientQueue->Rte_Result = Method(&(&ClientQueue->config));

and the compiler reports that it cannot take the address of object.

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 client-server calls and when the ServerArgumentImplPolicy is set to useArrayBaseType
for data types that are not array types.

Resolution Description:
Workaround:

Remove the ServerArgumentImplPolicy

Resolution:

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

ESCAN00101706	**TxConfirmation function prototype not generated.**
Component@Subcomponent:	Cdd_AsrCddCfg5@GenTool_GeneratorMsr
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Function prototype for TxConfirmation is not generated although it should.
This may lead to compile errors, depending on the CDD implementation.

When does this happen:

During code generation.

In which configuration does this happen:

Configurations where some PDUs of a single Cdd contribution have a TxConfirmation Name configured in the CanIf/FrIf/LinIf and others don't.
It depends of the order of the configured/unconfigured names.

Resolution Description:

Workaround:

Configure the TxConfirmation name either for all or for none of the PDUs in one Cdd contribution, not just for a few of them.

Resolution:

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

ESCAN00101721 Contradictory solving actions for the meta data length between CanIf and other module (e.g. PDUR, J1939DCM)

Component@Subcomponent:	If_AsrIfCan@GenTool_GeneratorMsr
First affected version:	4.04.00
Fixed in versions:	4.11.00

Problem Description:

What happens (symptoms):

CanIfTxPduMetaDataLengthValidator tries to minimize meta data length, which is in conflict to other modules (e.g. PDUR, J1939DCM) leading to contradictory solving actions.

When does this happen:

During configuration

In which configuration does this happen:

----

meta data length is used
&&
meta data length is bigger than actually used value by CanIf (e.g. closed Mask)
&&
PreCompile Configuration

Resolution Description:

Workaround:

----

1. Execute all solving action (solve all).
2. Solve all open validation messages on parameter MetaDataLength from the affected global PDU - until only CANIF30004 "The meta data length of a Tx-PDU is oversized (it exceeds the dynamic part of the CAN identifier)." remains.
2. Set parameter MetaDataLength of the affected global PDU to UserDefined.
3. Ignore validation message from CanIf.

Resolution:

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

ESCAN00101731	**Strapping pin configuration isn't overriden by the driver for all parameters where it might be necessary on TJA1102**
Component@Subcomponent:	DrvTrans_Tja1100EthAsr@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The NXP TJA1102 Ethernet transceiver supports the ability to pre-configure specific options like
- Clock Mode
- LDO Mode
- Reverse MII mode for P0 and P1
- PHY enable for P0 and P1

during chip startup through strapping pins.

Most commonly the strapping pin configuration fits to the actual hardware layout for the target application but in some board layouts there might be a misconfiguration that must be corrected by software.
Currently the driver doesn't set the mentioned configuration options and the configuration parameters for the noted options are missing.

When does this happen:

During initialization of the Ethernet stack.

In which configuration does this happen:

Strapping pin configuration of TJA1102 doesn't fit to the actual target application.

Resolution Description:

ESCAN00101731 Strapping pin configuration isn't overriden by the driver for all parameters where it might be necessary on TJA1102

Workaround:

Use the Post-Transceiver-Init-Callout functionality to set the respective configuration options.

(Note: The actual use case might differ from the following, as well as the actual SMI-index which can be obtained from the hardware layout)

Example for enabling the PHY with SMI-index 1 (overwrite strapping configuration of PHY_EN):

```
{
 uint16 regVal;
 uint8 registerAddress = 28;
 uint8 smiIdx = 1;

 (void)Eth_30_<Platform>_ReadMii(<SNVofRelatedEthCtrl>, smiIdx, registerAddress, &regVal); /* read current register value */

 regVal |= 0x0001u; /* Set the PHY-enable bit to overwrite the strapping configuration */

 (void)Eth_30_<Platform>_WriteMii(<SNVofRelatedEthCtrl>, smiIdx, registerAddress, regVal); /* write back modified register value */
}
```

Resolution:

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

ESCAN00101798	Auto Configuration Ecu State Handling - Additional modes of a Mode Declaration Group are removed after an update
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	10.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The BswM shows a validation message like the following:

BSWM01010 Inconsistent Mode Declaration Groups! (2 messages)
 BSWM01010 Mode Declarations Group of BswMSwitchedMode(value=<USER_MODE>) (<null Object>) and BswMModeSwitchInterfaceRef(value=<USER_MODE>) (ESH_Mode) are inconsistent!

If a mode is added manually to a Mode Declaration Group (/MICROSAR/BswM/BswMConfig/BswMRteModeDclGroup/BswMRteMode), it is being removed after the corresponding Auto Configuration Ecu State Handling is applied again.

This is relevant for the following Mode Declaration Groups of Ecu State Handling:
- ESH_RunRequest
- ESH_Mode

When does this happen:

During the configuration with DaVinci Configurator in the BSW Management Editor in the following sequence:
- Configure Ecu State Handling is clicked
- Finish is clicked
- The Mode Declaration Groups ESH_RunRequest or ESH_Mode are extended by at least one other mode
- Configure Ecu State Handling is clicked once again
- Finish is clicked
- the dialog 'Manual Adaptions' does not pop up or it pops up but the change is not displayed
- Finish is clicked in the 'Manual Adaptions' dialog if it is displayed

In which configuration does this happen:

Any configuration using the Ecu State Handling Auto Configuration in BSW Management in DaVinci Configurator

AND

Mode Declaration Group ESH_RunRequest or ESH_Mode are available

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.

ESCAN00101829	PDUR90005 Exception in PduR generator during Validation encountered
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	14.00.00
Fixed in versions:	15.03.00

Problem Description:

What happens (symptoms):

The PduR does not generate because the validation does not complete with the following error message:

PDUR90005 Exception in PduR generator during Validation encountered

When does this happen:

The exception is issued by the tool during validation of the configuration in case the configuration is as described below.

In which configuration does this happen:

Any configuration, where the parameter of the following definition exists and all values are configured to false

/MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRMulticoreRoutingPath

and

not minimum one /MICROSAR/EcuC/EcucHardware/EcucCoreDefinition exists with a configured EcucCoreHwRef.

Resolution Description:

Workaround:

Delete in the single core use case all instances of /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRMulticoreRoutingPath.

Resolution:

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

ESCAN00101835 Compiler switch is generated erroneously for the Tasking compiler

Component@Subcomponent:	DrvCan_Mpc5700McanLl@GenTool_GeneratorMsr
First affected version:	3.13.00
Fixed in versions:	3.15.00

Problem Description:

What happens (symptoms):

The compiler switch "V_COMP_TASKING" is generated as "V_COMP_Tasking". The code that relies on the upper case switch will then not be active.

When does this happen:

At configuration time

In which configuration does this happen:

In all configurations where the selected compiler is Tasking

Resolution Description:

Workaround:

Provide the following code in the user config:

```
#if !defined( V_COMP_TASKING )
# define V_COMP_TASKING
#endif
```

Resolution:

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

ESCAN00101870 Null pointer exception when module description for LDCOM is missing

Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.04.00
Fixed in versions:	4.20.95, 4.20.00

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:

When the reference to the BSW-IMPLEMENTATION of LDCOM is missing in the LDCOM ECUC configuration.

The problem normally does not occur when a MICROSAR LDCOM is used with Configurator 5.

Activating the module automatically sets the reference.

Resolution Description:

Workaround:

Configure the reference to the BSW-IMPLEMENTATION in the LDCOM configuration.

Resolution:

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

ESCAN00101880	Compiler error: Compilation fails due to missing NvM.h
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.18.00
Fixed in versions:	1.20.00, 1.20.95

Problem Description:

What happens (symptoms):

Compilation fails because a runnable in Rte_<oa>.c passes a NV block handle to an NVM API while NvM.h is not included.

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 memory protection is used and when the NvRunnable is called from another task and when no dirty flags are configured.

Resolution Description:

Workaround:

Configure dirty flags.

Resolution:

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

ESCAN00101955 BswM does not support Mode Request R Ports with Target Variable Data Prototypes and Data Type Mapping Sets

Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The BswM throws the following error:

BSWM01020 Target Variable Data Type Prototype must have an Implementation Data Type with a Compu Method! Otherwise, selected Mode can not be resolved!
/ActiveEcuC/BswM/BswMConfig/BswMArbitration/SwcModeRequest_SftyL3a_sopt_successfull/BswMModeRequestSource/BswMSwcModeRequest
/ActiveEcuC/BswM/BswMConfig/BswMArbitration/SwcModeRequest_SftyL3a_sopt_successfull/BswMModeRequestSource/BswMSwcModeRequest[BswMSwcModeRequestPortInterfaceRef]

When does this happen:

After configuring a Mode Request R Port.

In which configuration does this happen:

In configurations with Mode Request R Ports which use a Target Variable Data Prototype [BswMSwcModeRequest/BswMSwcModeRequestModeDeclarationGroupPrototypeRe] which reference an Application Data Type and no Implementation Data Type, even if the Application Data Type is mapped via a Data Type Mapping Set.

Resolution Description:

Workaround:

Use Mode Declaration Groups via BswMSwcModeRequestModeDeclarationGroupPrototypeRef instead of the BswMSwcModeRequestPortInterfaceRef.

Resolution:

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

ESCAN00101968 Wrong value of define DIAG_ASR4DCM_VERSION

Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	10.00.00
Fixed in versions:	11.05.00

Problem Description:

What happens (symptoms):

The define DIAG_ASR4DCM_VERSION has a wrong value.
Actual:
# define DIAG_ASR4DCM_VERSION 0x0A04u

Expected:
# define DIAG_ASR4DCM_VERSION 0x1004u

When does this happen:

Every time

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.

ESCAN00102032	OsResources are not configurable for each EcuM Core
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The configuration tool will throw the following error message in case that for all cores a scheduler resource is configured:

ECUM01013 OS Resource configuration is not correct. (1 message)
ECUM01013 Core number of '<OSRESOURCE>' could not be determined. Make sure that OsResource is used by an OsTask which is referenced by an OsApplication with a Core Assignment!
/ActiveEcuC/EcuM/EcuMConfiguration/EcuMCommonConfiguration[2:EcuMOSResource]

As a consequence it is not possible to allocate the scheduler resource on a slave core while this core is running to a sleep phase. This means that other tasks can interrupt the task (e.g. BSW task) which is responsible to bring the slave Core to sleep.

When does this happen:

Always when the BSW Core switches the ECU to a sleep mode.

In which configuration does this happen:

In Multicore configurations with an OS which do not provide the parameter /Os/OsApplication/OsApplicationCoreAssignment.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102127 Compiler error: Linker error: ltc E106: unresolved external

Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	2.19.00
Fixed in versions:	2.30.00

Problem Description:

What happens (symptoms):

After compilation the linker throws the error
ltc E106: unresolved external: <SymbolName> - (Os_Hal_Context_Lcfg.o) with Tasking tools
Os_Hal_Context_Lcfg.o:(.rodata.OS_CORE1_CONST+0x8c): undefined reference to
`<SymbolName>' with HighTec Gnu tools

When does this happen:

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

In which configuration does this happen:

When configuring an ISR to be a DMA interrupt
OsIsrInterruptMapping = /ActiveEcuC/Os/OsPublishedInformation/<Derivative>/DMA

Resolution Description:

Workaround:

Implement an empty category 2 ISR

```
ISR(<configured DMA ISR Name>)
{
}
```

Resolution:

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

ESCAN00102181 DVCfgRteGen terminates unexpectedly

Component@Subcomponent:	Rte_DaVinciBase@Generator
First affected version:	3.04.01
Fixed in versions:	4.20.00, 4.20.93

Problem Description:

What happens (symptoms):

DVCfgRteGen terminates unexpectedly

When does this happen:

During generation.

In which configuration does this happen:

When the model contains data reception triggers with a missing data element reference.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102223 Description of Service IDs in Technical Reference don't match Sercive IDs used in the implementation

Component@Subcomponent:	DrvEth__coreAsr@Doc_TechRef
First affected version:	2.01.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

The service IDs described in the Technical Reference don't match with the service IDs used in the Implementation.

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.

ESCAN00102235	Compiler error: Use of undeclared identifier TriggerDisableFlags
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

Compilation fails because Rte*.c accesses a TriggerDisableFlag that has not been 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:

This happens when all of the following conditions evaluate to true:
- the configuration contains multiple partitions
- there are mode disabled runnables
- all related mode disabled runnables are mapped to the same partition as the mode provider
- the task that performs the mode switch (the mode machine) is assigned to another partition
 (this is the case, for example, if a mode receiver with OnEntry-, OnTransistion- or OnExit triggered runnable exists on another partition)

Resolution Description:

Workaround:

Create a dummy runnable with mode disabled trigger in the partition to which the mode machine / mode switch task is assigned.

Resolution:

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

ESCAN00102287	Auto-Configuration Communication Control: No support of J1939TpTxDirectNPdu
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	12.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The BswM Auto-Configuration shows the following message when selecting a PDU Group which is not assigned to a J1939 Nm Node:

"Channel of corresponding Group PDUs could not be determined."

When does this happen:

When BswM Auto-Configuration Communication Control is executed.

In which configuration does this happen:

In J1939 configurations with J1939TP channels which refer to a global EcuC Pdu via the reference parameter J1939TpTxDirectNPdu [/J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxDirectNPdu].

Resolution Description:

Workaround:

Auto-Configuration wizard is not available, but the rules can be configured by the user manually.

Resolution:

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

ESCAN00102325 Compiler error: Client server communication uses undefined trusted function calls

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.03.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

Linker fails with an error message
"unresolved external symbol Call_Rte_TrustedCall_..."

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 all of the following conditions evaluate to true:
- the configuration contains multiple partitions and cores
- the configuration contains client server communication between two partitions on the same core with at least one untrusted partition.
- client and server are assigned to a different core than the BSW core.
- BSW runs in a trusted partition.

Resolution Description:

Workaround:

Add another client that additionally calls the server from a different core.

Resolution:

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

ESCAN00102364 Generator error RTE49999 when scaling of child constants fails

Component@Subcomponent:	Rte_Asr4@Generator
First affected version:	4.00.00
Fixed in versions:	4.20.00

Problem Description:

What happens (symptoms):

RTE generation aborts with an exception
RTE49999 The method or operation is not implemented. at
Interop.DVGenAPILib.IGenConstant.get_Name()

When does this happen:

During generation.

In which configuration does this happen:

This happens when the configuration contains initial values for which the conversion with factor and offset from the compu method of the application data type leads to an overflow exception.

Resolution Description:

Workaround:

Use an implementation data type or remove the compu method if it is not needed.
The initial value can then be specified directly as internal value.

Resolution:

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

ESCAN00102408 Compiler error: no member named Rte_ModeSwitchAck_

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.20.00

Problem Description:

What happens (symptoms):

Compilation fails because a task accesses a Rte_<OsApplName>_AckFlags 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:

CrossCore mode communication (Mode disablings, OnEntry, OnExit, OnTransition) with different port prototype names for sender and receiver port.

Resolution Description:

Workaround:

Rename the sender or receiver port so that they have the same port prototype name.

Resolution:

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

ESCAN00102438 Ambiguous description of error handling behavior in OS TechRef.

Component@Subcomponent:	Os_CoreGen7@Doc_TechRef
First affected version:	1.00.00
Fixed in versions:	2.25.00

Problem Description:

What happens (symptoms):

Documentation Issue in Chapter 3.13 Error Handling:

The last sentence can be misleading. The interrupts are disabled directly after the error occurs, not after the application has been informed abaout the occurence.

When does this happen:

Documentation issue.

In which configuration does this happen:

Documentation issue.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102445 PduR Mainfunction is not written into the internal behavior in Singlecore configurations

Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	14.00.00
Fixed in versions:	15.03.00

Problem Description:

What happens (symptoms):

The PduR Mainfunction is not written into the internal behavior. This way the user can not map the PduR Mainfunction to a task.

When does this happen:

Always if the below conditions match.

In which configuration does this happen:

All conditions must be true:

- Singlecore configurations (no /PduR/PduRBswModules/PduROsApplicationRef parameter exists)
- Mainfunction is used (/PduR/PduRGeneral/PduRMainFunctionPeriod is configured to any value)

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102467	Size of data computed by comfort view is lower then the actual needed size
Component@Subcomponent:	If_AsrIfFee@GenTool_GeneratorMsr
First affected version:	3.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The Partition Overhead Analysis in Memory Blocks Tab in Cfg5 comfort view shows wrong values.
Actually, it is expected that it computes the space that is needed if every block is written once so that the user knows how much space is still available.
Nevertheless, the computation is implemented wrong and does not involve the link table and the sector header, i.e. space that cannot be used by the user.
The user might think that he has more space than he actually has.

However, this cannot lead to faulty generated code. For all validation rules another function is used to determine the space and this function works properly.
It might only lead to wrong expectations and confusion for the user.

When does this happen:

It happens always, but the higher the number of configured blocks and the higher the alignments the bigger the difference of computed and real size gets.

In which configuration does this happen:

It happens for all configurations

Resolution Description:

Workaround:

Use Flash Endurance Excel sheet provided by Vector in order to compute size needed in flash.

Resolution:

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

ESCAN00102477	Compiler error: illegal zero-sized array // undeclared identifier
Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	12.01.00

Problem Description:

What happens (symptoms):

The compiler will issue compiler errors like:

```
1>..\..\..\..\external\BSW\Dcm\Dcm.c(1911): error C2229: struct
'DCM_SVC2CCONTEXTTYPE_TAG' has an illegal zero-sized array
1>..\..\..\..\external\BSW\Dcm\Dcm.c(1913): error C2229: struct
'DCM_SVC2CCONTEXTTYPE_TAG' has an illegal zero-sized array
1>..\..\..\..\external\BSW\Dcm\Dcm.c(27308): warning C4305: 'function' : truncation from
'unsigned int' to 'uint16_least'
1>..\..\..\..\external\BSW\Dcm\Dcm.c(27702): warning C4305: 'function' : truncation from
'unsigned int' to 'uint16_least'
1>..\..\..\..\external\BSW\Dcm\Dcm.c(27780): warning C4305: 'function' : truncation from
'unsigned int' to 'uint16_least'
1>..\..\..\..\external\BSW\Dcm\Dcm.c(34272): error C2065: 'Dcm_CfgDidMgrOpInfoDefine' :
undeclared identifier
1>..\..\..\..\external\BSW\Dcm\Dcm.c(34272): error C2109: subscript requires array or pointer
type
1>..\..\..\..\external\BSW\Dcm\Dcm.c(34338): error C2065: 'Dcm_CfgDidMgrOpInfoDefine' :
undeclared identifier
1>..\..\..\..\external\BSW\Dcm\Dcm.c(34338): error C2109: subscript requires array or pointer
type
1>..\..\..\..\external\BSW\Dcm\Dcm.c(37978): error C2065: 'Dcm_CfgDidMgrOpInfoDefine' :
undeclared identifier
1>..\..\..\..\external\BSW\Dcm\Dcm.c(37978): error C2109: subscript requires array or pointer
type
```

Hint: Root cause is a missing validator that checks that each dynamically defined DID has read access (see below for configuration details). There exists already a validator that checks that each dynamically defined DID has define access.

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 where all dynamically defined DIDs (/MICROSAR/Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidDynamicallyDefined set to TRUE) have additionally

- Define Access configued (/MICROSAR/Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidAccess/DcmDspDidDefine container available)

BUT NOT

- Read Access configured (/MICROSAR/Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidAccess/DcmDspDidRead container not available)

Resolution Description:

ESCAN00102477 Compiler error: illegal zero-sized array // undeclared identifier

Workaround:

Configure /MICROSAR/Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidAccess/DcmDspDidRead for each dynamically defined DID.

Resolution:

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

ESCAN00102541 RTE49999: When mode disablings are used over core boundaries and the receiver port is assigned to core 0

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.21.00

Problem Description:

What happens (symptoms):

RTE generator aborts with a RTE49999 error

When does this happen:

During generation

In which configuration does this happen:

- Cross core mode communication.
- Receiver mode port is assigned to core 0
- Sender mode port is assigned to core != 0

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102543	Validator limits SPI external buffer length to 256 bytes although more is needed
Component@Subcomponent:	DrvEthSwitch_Sja1105PQRSAsr@GenTool_GeneratorMsr
First affected version:	5.00.00
Fixed in versions:	5.00.02

Problem Description:

What happens (symptoms):

A validators issues that the SPI external buffer length must be set to 256 byte although the SJA1105PQRS driver needs way more.

When does this happen:

During configuration.

In which configuration does this happen:

Any configuration.

Resolution Description:

Workaround:

Set SPI external buffer length parameter to user-defined and set it to 512 bytes

Resolution:

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

ESCAN00102639 Compiler error: RTE calls IocSend functions that do not exist

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.04.00
Fixed in versions:	1.21.00

Problem Description:

What happens (symptoms):

Compilation fails because Rte_LdComRxIndication or Rte_COMCbk or Rte_Call functions
call IocSend_ functions that do not exist. However the OS generates functions IocSend__<digit>

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 all of the following conditions evaluate to true:
- the configuration contains client-server communication
- one server is connected to at least two clients
- one of the clients is on another ECU
- the BSW is located on a different core than the server
- the BSW runs nontrusted
- one of the clients is located in another nontrusted partition

Resolution Description:

Workaround:

Change the configuration so that all clients are located in the BSW partition
or create separated server ports for every client (the server ports can trigger the same runnable)

Resolution:

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

ESCAN00102682 RTE49999 when size field of variable length array uses implementation data type that is mapped to application data type with texttable compu method

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.04.00
Fixed in versions:	1.21.00

Problem Description:

What happens (symptoms):

Generation aborts with an error message
RTE49999 Non-compliant implementation data type.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the configuration contains an application array data type with variable length that is mapped to an implementation record data type with an integer size field and an array data field.
Moreover the integer data type for the size field also needs to be used in combination with a compu method.
This is the case when:
- the integer data type references a compu method directly
- it is mapped to an application data type with compu method
- it is used as target for a type reference in a complex data type that references a compu method

Resolution Description:

Workaround:

Create a type reference to the data type that is used for the size field.
Use this type reference for the size field.

Resolution:

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

ESCAN00102685	Compiler error: In file Os_Types_Lcfg.h: undeclared identifier
Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	2.31.00

Problem Description:

What happens (symptoms):

Compiler states an error message about undeclared identifier in Os_Types_Lcfg.h within the enum type definition of ResourceType in the line where the value of RES_SCHEDULER is 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:

This happens only in configurations with just one AUTOSAR core and attribute /MICROSAR/Os/OsOS/OsUseResScheduler set to true.
Additionally, the issue only occurs if the name of the res-scheduler resource of the AUTOSAR core does not match the naming rule:
 RES_SCHEDULER_<CoreName>
Where <CoreName> is the name of the master core.

Resolution Description:

Workaround:

On systems where the master core is the AUTOSAR core: Simply follow the naming rule as described at "In which configuration does this happen".
On systems where the master core is not the AUTOSAR core, this might have too strange implications. Consider to set /MICROSAR/Os/OsOS/OsUseResScheduler to false, remove the resource which is linked at /MICROSAR/Os/OsCore/OsCoreResScheduler of the AUTOSAR core and create a new resource, named RES_SCHEDULER which needs then to be assigned to all tasks.

Resolution:

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

ESCAN00102686 Compiler error: Missing variable declaration for receivers in other partitions when the BSW partition contains no sender-receiver communication

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.01.01
Fixed in versions:	1.21.00

Problem Description:
What happens (symptoms):

Compilation fails because a sender-receiver communication buffer is not generated.

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 only contains external communication and when the signal is received in a different partition than the partition that contains the COM module.

Resolution Description:
Workaround:

Create an internal sender-receiver communication in the BSW partition.

Resolution:

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

ESCAN00102707 Misleading error code during DAQ list allocation ERR_OUT_OF_RANGE

Component@Subcomponent:	Cp_Asr4Xcp@Implementation
First affected version:	1.00.00
Fixed in versions:	5.00.00

Problem Description:

What happens (symptoms):

The commands ALLOC_ODT and ALLOC_ODT_ENTRY are used to allocate the members of a DAQ list. Should on of the parameters ODT_COUNT or ODT_ENTRIES_COUNT be to big for the still available (configured) number of possible parameters the XCP slave returns an ERR_OUT_OF_RANGE (0x22u). The correct error code for this case would be ERR_MEMORY_OVERFLOW (0x30u) which instructs the XCP Master to retry DAQ allocation with other parameters.

When does this happen:

Always and immediately

In which configuration does this happen:

When DAQ is used.
/ActiveEcuC/Xcp/XcpCmdConfig/XcpDaqAndStim

Resolution Description:

Workaround:

No workaround required. The check for the described conditions works as intended, only the returned error code is misleading.

Resolution:

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

ESCAN00102720	RTE generator creates invalid CompuMethods when a data type references a CompuMethod of category BITFIELD_TEXTTABLE
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.06.00
Fixed in versions:	1.21.00

Problem Description:

What happens (symptoms):

Measurement tool reports duplicate enteries of bitfield values in the generated a2l compu method

When does this happen:

During A2L update (measurement)

In which configuration does this happen:

This happens in configurations in which the measurement is active and compu method type is BITFIELD_TEXTTABLE

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102739	**Generator consistency issue: A S/R port cannot have dynamic length**
Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

DCM generator issues the following message:

ConsistencyRT00002 Consistency: <...> Ex: java.lang.IllegalStateException: A S/R port cannot have dynamic length!
We are sorry, but due to this internal error, code generation of all modules has to be blocked. As a workaround, you may try to restart DaVinci Configurator. Otherwise, please call Vector for support.

When does this happen:

During configuration.

In which configuration does this happen:

- At least 1 DID exists with dynamic length (/Dcm/DcmConfigSet/DcmDsp/DcmDspDataInfo/DcmDspDataFixedLength == FALSE)
AND
 - This DID supports S/R data access
 OR
 - This DID supports NV-Data access

Resolution Description:

Workaround:

Set parameter "/Dcm/DcmConfigSet/DcmDsp/DcmDspDataInfo/DcmDspDataFixedLength" for all DIDs with S/R or NV-Data access to TRUE. When suing S/R communication all DIDs and data elements shall have constant length (see TechnicalReference section "How to setup DCM for Sender-Receiver Communication" and sub-section "Implementation Limitations").

Resolution:

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

ESCAN00102769 MISRA deviations: conflicting redeclaration of Dem_SetOperationCycleState

Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	17.01.00

Problem Description:

What happens (symptoms):

There might be MISRA errors or warnings that Symbol 'Dem_SetOperationCycleState(uint8, Dem_OperationCycleStateType)' is redeclared and/or conflicts with the declaration in Dem/ Dem_API_Interface.h.

When does this happen:

MISRA analysis

In which configuration does this happen:

Happens only with some MISRA tools and configurations.

Resolution Description:

Workaround:

Error or warning can be ignored.

The port is defined with the port defined argument OperationCycleId of type Dem_OperationCycleIdType instead of type uint8.
As in Rte_Type.h, Dem_OperationCycleIdType is defined as uint8 via
‚typedef uint8 Dem_OperationCycleIdType'
this MISRA finding can be ignored.

In Rte_SwcDemSc.h the function is declared as follows:
FUNC(Std_ReturnType, RTE_<X>_APPL_CODE)
Dem_SetOperationCycleState(Dem_OperationCycleIdType parg0, Dem_OperationCycleStateType CycleState);

In Dem_API_Interface.h, the function is declared as follows.
FUNC(Std_ReturnType, DEM_CODE) Dem_SetOperationCycleState(uint8 OperationCycleId, Dem_OperationCycleStateType CycleState);

Resolution:

The port is defined with the port defined argument OperationCycleId of type uint8.

ESCAN00102807 Compiler error: RTE calls Rte_IocSend_ and Rte_IocReceive_ functions that do not exist

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

Compilation fails because the RTE calls Rte_IocSend_ And Rte_IocReceive_ functions 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 the combination of component, port and element name exceeds 128 characters.

Resolution Description:
Workaround:

Use short short names.

Resolution:

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

ESCAN00102825 Missing CanTpConnections and CanTpChannels

Component@Subcomponent:	GenTool_CsAsrLegacyDb2SystemDescr@Application
First affected version:	1.08.24
Fixed in versions:	

Problem Description:
What happens (symptoms):

The necessary CanTpConnections for OvTpIPdus are not created.

When does this happen:

During conversion

In which configuration does this happen:

In any configuration.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00102826 Max value of LinClockFrequency is too low.

Component@Subcomponent:	DrvLin_TricoreAsr@Description
First affected version:	5.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The live Validation returns with an error.

When does this happen:

If the customer selects a frequency higher than 50 MHz in the "Lin Clock Frequency" parameter.

In which configuration does this happen:

Every configuration.

Resolution Description:

Workaround:

Set value to user defined

Resolution:

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

ESCAN00102846 Validation and generation error with Postbuild-Selectable

Component@Subcomponent:	If_AsrIfSoAd@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Validation error:
- SOAD01504 SoAd PDU Route issue.
 <xy> references a global PDU, which isn't referenced by a propper SoAd upper layer BSW module reference
- SOAD01502 SoAd Socket Route Destination issue.
 <xy> references a global PDU, which isn't referenced by a propper SoAd upper layer BSW module reference

Generation error:
SOAD90005 Exception in SoAd generator during Generation encountered:
- java.lang.IllegalStateException: The size of the keys: (<xy>) of the ConstStruct: PduRoute does not match to the size of the ConstStruct !
 /ActiveEcuC/SoAd
- java.lang.IllegalStateException: The size of the keys: (<xy>) of the ConstStruct: SocketRoute does not match to the size of the ConstStruct !
 /ActiveEcuC/SoAd

When does this happen:

Issue happens on validation and/or generation time.

In which configuration does this happen:
----
1. SoAd/SoAdConfig/SoAdPduRoute/SoAdTxPduRef
references an
EcuC/EcucPduCollection/Pdu
which is referenced by a module with enabled Postbuild-Selectable and corresponding reference is variant depended.

and/or

2. SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest/SoAdRxPduRef
references an
EcuC/EcucPduCollection/Pdu
which is referenced by a module with enabled Postbuild-Selectable and corresponding reference is variant depended.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102871 Wrong validation of memory region end address alignment.

Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The DaVinci Cfg5 states an validation error:

OS02910 The region endaddress-1 must be a mulptiple of [x].
/ActiveEcuC/Os/OsMemoryProtection/Core0_Global_PS0DSPR0[OsMemoryRegionEndAddress]

where [x] is the hardware memory region granularity.

When does this happen:

This happens during the configuration of a OS memory region container.
/MICROSAR/Os/OsMemoryProtection/OsMemoryRegion

In which configuration does this happen:

This happens in all SC3 or SC4 configurations where the region end address parameter is configured as numerical value.
/MICROSAR/Os/OsMemoryProtection/OsMemoryRegion/OsMemoryRegionEndAddress

Resolution Description:

Workaround:

Define a linker label with the corresponding address and use it to configure the regions end address.

Resolution:

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

ESCAN00102873 Missing information in Technical Reference that WdgM supports only up to four cores in multi-core system

Component@Subcomponent:	SysService_Asr4WdM@Doc_TechRef
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The WdgM and the WdgIf can be used in multi-core system, but only with up to four cores. This information is missing in the technical reference. However, the parameter /MICROSAR/WdgM/WdgMConfigSet/WdgMMode/WdgMModeCoreAssignment (that is the processor core ID) has valid range [0,3] and if two parameters WdgMModeCoreAssignment are the same a validator shows an error message "The mode core assignment has an invalid value." Therefore, when trying to configure more than four cores, it will become apparent that this is not possible.

When does this happen:

The issue is missing information within Technical Reference for multi-core systems.

In which configuration does this happen:

The issue is missing information within Technical Reference for multi-core systems.

Resolution Description:

Workaround:

The missing information will become apparent during configuration: the parameter /MICROSAR/WdgM/WdgMConfigSet/WdgMMode/WdgMModeCoreAssignment (that is the processor core ID) has valid range [0,3] and if two parameters WdgMModeCoreAssignment are the same a validator shows an error message "The mode core assignment has an invalid value." Therefore, when trying to configure more than four cores, it will become apparent that this is not possible.

Resolution:

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

ESCAN00102887	CANTSYN90001 - generator Error thrown for postbuild configuration
Component@Subcomponent:	SysService_AsrTSynCan@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

CANTSYN90001
CANTSYN90001 Fatal Generator (MICROSAR CanTSyn Generator) failure (1 message)
CANTSYN90001 Generation phase has reported an error. The file generation was aborted.

When does this happen:

----

on generation phase

In which configuration does this happen:

----

when a CanTSYN Message is used by 2 different Variants

Resolution Description:

Workaround:

No workaround available.

Resolution:

Add a validator to detect postbuild variants and raise a warning.

ESCAN00102907 Compiler error: Section .OS_CORESTATUS_COREx_VAR_NOCACHE_N isn't included in the section map

Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	2.25.00
Fixed in versions:	2.31.00

Problem Description:

What happens (symptoms):

The compiler (linker) shows an error message like the following:

"... section .OS_CORESTATUS_COREx_VAR_NOCACHE_NOINIT from Os_Hal_Kernel_Lcfg.o isn't included in the section map."

Where "COREx" is replaced with a specific core ID. For example: "CORE1".

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 '/MICROSAR/Os/OsCore' has to be configured with '/MICROSAR/Os/OsCore/OsCoreIsAutosar' set to FALSE
and the vLinkGen is used to generate the project linker-script.

Resolution Description:

Workaround:

Create a new "vLinkGenVarGroupUserSection" within the OS_DATA_COREx (/MICROSAR/vLinkGen/vLinkGenMemLayout/vLinkGenLinkerSectionVarGroup) linker group.
Then add a compiler specific section definition to "Value". The following describes the string to be used for the GHS compiler, for other compilers the section definition might differ.

".OS_CORESTATUS_COREx_VAR_NOCACHE_NOINIT :>."

Where "x" is replaced by the corresponding logical core ID.

Resolution:

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

ESCAN00102912	Compiler error: 'IpV4_State' uses conflicting section name
Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	8.00.00
Fixed in versions:	10.00.00

Problem Description:

What happens (symptoms):

Compiler error due to different memory sections in source and header file for the variable IpV4_State.

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 differenty memory abstraction sections are used for initialized and uninitialized (zero initialized) data if IpV4.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102918 Compiler error: Multiple missing defines (e.g. ETHSWT_CONSTP2CONST)

Component@Subcomponent:	DrvEthSwitch_Sja1105PQRSAsr@Implementation
First affected version:	1.00.00
Fixed in versions:	5.00.02

Problem Description:

What happens (symptoms):

Compile complains about multiple missing defines related to compiler abstraction like:
ETHSWT_CONSTP2CONST
and other defines like
ETHSWT_INVALID_PORT_IDX
ETHSWT_PORT_SPEED_100_MBIT

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.

Resolution Description:

ESCAN00102918 Compiler error: Multiple missing defines (e.g. ETHSWT_CONSTP2CONST)

Workaround:

The following macros can be added to a header file, e.g. EthSwt_30_Sja1105PQRS_Types.h:

```
#ifndef ETHSWT_INVALID_SWITCH_IDX
# define ETHSWT_INVALID_SWITCH_IDX (0xFF)
#endif /* ETHSWT_INVALID_SWITCH_IDX */
#ifndef ETHSWT_INVALID_PORT_IDX
# define ETHSWT_INVALID_PORT_IDX (0xFF)
#endif /* ETHSWT_INVALID_PORT_IDX */

/*! Port speed */
#ifndef ETHSWT_PORT_SPEED_10_MBIT
# define ETHSWT_PORT_SPEED_10_MBIT ETHSWT_PORT_SPEED_10
#endif /* ETHSWT_PORT_SPEED_10_MBIT */
#ifndef ETHSWT_PORT_SPEED_100_MBIT
# define ETHSWT_PORT_SPEED_100_MBIT ETHSWT_PORT_SPEED_100
#endif /* ETHSWT_PORT_SPEED_100_MBIT */
#ifndef ETHSWT_PORT_SPEED_1000_MBIT
# define ETHSWT_PORT_SPEED_1000_MBIT ETHSWT_PORT_SPEED_1000
#endif /* ETHSWT_PORT_SPEED_1000_MBIT */

/*! Physical layer type */
#ifndef ETHSWT_PORT_PHYS_TYPE_BROAD_R_REACH
# define ETHSWT_PORT_PHYS_TYPE_BROAD_R_REACH ETHSWT_PORT_BROAD_R_REACH
#endif /* ETHSWT_PORT_PHYS_TYPE_BROAD_R_REACH */
#ifndef ETHSWT_PORT_PHYS_TYPE_BASE_T
# define ETHSWT_PORT_PHYS_TYPE_BASE_T ETHSWT_PORT_BASE_T
#endif /* ETHSWT_PORT_PHYS_TYPE_BASE_T */
#ifndef ETHSWT_PORT_PHYS_TYPE_RTPGE
# define ETHSWT_PORT_PHYS_TYPE_RTPGE ETHSWT_PORT_RTPGE
#endif /* ETHSWT_PORT_PHYS_TYPE_RTPGE */
#ifndef ETHSWT_PORT_PHYS_TYPE_X_MII
# define ETHSWT_PORT_PHYS_TYPE_X_MII ETHSWT_PORT_X_MII
#endif /* ETHSWT_PORT_PHYS_TYPE_X_MII */
#ifndef ETHSWT_PORT_PHYS_TYPE_INTERNAL
# define ETHSWT_PORT_PHYS_TYPE_INTERNAL ETHSWT_PORT_INTERNAL
#endif /* ETHSWT_PORT_PHYS_TYPE_INTERNAL */

/* PRQA S 0342,0881 GLUE_OPERATOR */ /* MD_MSR_19.13_0342 */
#ifndef ETHSWT_CONSTP2CONST
/*! Macro for better readability of compiler abstraction CONSTP2CONST */
# define ETHSWT_CONSTP2CONST(DataType, Derivative) \
 CONSTP2CONST(DataType, ETHSWT_30_##Derivative##_APPL_CONST,
ETHSWT_30_##Derivative##_APPL_CONST)
#endif /* ETHSWT_CONSTP2CONST */

#ifndef ETHSWT_CONSTP2VAR
/*! Macro for better readability of compiler abstraction CONSTP2VAR */
# define ETHSWT_CONSTP2VAR(DataType, Derivative) \
 CONSTP2VAR(DataType, ETHSWT_30_##Derivative##_APPL_CONST,
ETHSWT_30_##Derivative##_APPL_VAR)
#endif /* ETHSWT_CONSTP2VAR */

#ifndef ETHSWT_CONSTP2VAR_WITH_MEM
/*! Macro for better readability of compiler abstraction CONSTP2VAR */
# define ETHSWT_CONSTP2VAR_WITH_MEM(DataType, MemType, Derivative) \
```

ESCAN00102918 Compiler error: Multiple missing defines (e.g. ETHSWT_CONSTP2CONST)

```
 CONSTP2VAR(DataType, ETHSWT_30_##Derivative##_APPL_CONST,
ETHSWT_30_##Derivative##_##MemType)
#endif /* ETHSWT_CONSTP2VAR_WITH_MEM */

#ifndef ETHSWT_P2CONST
/*! Macro for better readability of compiler abstraction P2CONST */
# define ETHSWT_P2CONST(DataType, Derivative) \
 P2CONST(DataType, AUTOMATIC, ETHSWT_30_##Derivative##_CONST)
#endif /* ETHSWT_P2CONST */

#ifndef ETHSWT_P2VAR
/*! Macro for better readability of compiler abstraction P2VAR */
# define ETHSWT_P2VAR(DataType, Derivative) \
 P2VAR(DataType, AUTOMATIC, ETHSWT_30_##Derivative##_APPL_VAR)
#endif /* ETHSWT_P2VAR */

#ifndef ETHSWT_P2VAR_WITH_MEM
/*! Macro for better readability of compiler abstraction P2VAR */
# define ETHSWT_P2VAR_WITH_MEM(DataType, MemType, Derivative) \
 P2VAR(DataType, AUTOMATIC, ETHSWT_30_##Derivative##_##MemType)
#endif /* ETHSWT_P2VAR_WITH_MEM */

#ifndef ETHSWT_CONST
/*! Macro for better readability of compiler abstraction CONST */
# define ETHSWT_CONST(DataType, Derivative) \
 CONST(DataType, ETHSWT_30_##Derivative##_CONST)
#endif /* ETHSWT_CONST */

#ifndef ETHSWT_VAR
/*! Macro for better readability of compiler abstraction VAR */
# define ETHSWT_VAR(DataType, Derivative) \
 VAR(DataType, ETHSWT_30_##Derivative##_APPL_VAR)
#endif /* ETHSWT_VAR */

#ifndef ETHSWT_VAR_WITH_MEM
/*! Macro for better readability of compiler abstraction VAR */
# define ETHSWT_VAR_WITH_MEM(DataType, MemType, Derivative) \
 VAR(DataType, ETHSWT_30_##Derivative##_##MemType)
#endif /* ETHSWT_VAR_WITH_MEM */

#ifndef ETHSWT_FUNC
/*! Macro for better readability of compiler abstraction FUNC */
# define ETHSWT_FUNC(RetType, Derivative) \
 FUNC(RetType, ETHSWT_30_##Derivative##_CODE)
#endif /* ETHSWT_FUNC */
/* PRQA L:GLUE_OPERATOR */
```

Resolution:

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

ESCAN00102919 Compiler error: 'EthSwt_30_Sja1105PQRS_UpdateMCastPortAssignB uses conflicting section name

Component@Subcomponent:	DrvEthSwitch_Sja1105PQRSAsr@Implementation
First affected version:	1.00.00
Fixed in versions:	5.00.02

Problem Description:

What happens (symptoms):

Compiler complains:
'EthSwt_30_Sja1105PQRS_UpdateMCastPortAssignBuckets' uses conflicting section name

because EthSwt_30_Sja1105PQRS_UpdateMCastPortAssignBuckets is mapped to an INIT section although it is UNINIT data.

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/EthSwt_Sja1105PQRS/EthSwt/EthSwtGeneral/EthSwtUpdateMCastPortAssignmentApi == TRUE

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102924 Compiler error: RTE calls Rte_LdComTpRxReadLocksInit functions that do not exist

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.19.00
Fixed in versions:	1.21.00

Problem Description:
What happens (symptoms):

Compilation fails because Rte_InitMemory calls Rte_LdComTpRxReadLocksInit function 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 when all of the following conditions evaluate to true:
- the configuration contains multiple partitions
- the configuration uses the LdCom TP API
- there is no Rte_Read that uses LdCom TP and is called from multiple task contexts that can interrupt each other.

Resolution Description:
Workaround:

In each affected partition, configure an additional Rte_Read that uses LdCom TP
and is called from multiple task contexts that can interrupt each other.

Resolution:

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

ESCAN00102930 RTE49999: when unconnected server runnable is mapped to a task

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.21.00

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 an unconnected server runnable is mapped to a task.

Resolution Description:

Workaround:

Connect the server and create a client server call point.
It is not necessary to call the API.

Resolution:

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

ESCAN00102944 Compiler error: Missing CounterType

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compilation fails because an Rte_Call API uses an unknown data type CounterType.

```
# define Rte_Call_Os_Service_GetCounterValue(arg1)
(GetCounterValue((CounterType)SystemTimer, arg1)) /* PRQA S 3453 */ /* MD_MSR_19.7 */
```

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 the GetCounterValue and GetElapsedValue defines from the OS is called.

Resolution Description:

Workaround:

Add the CounterType data type to the included data type set of the SWC or
configure "Enable API usage by address" for the port.

Resolution:

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

ESCAN00102947	ConsistencyRT00002 - IllegalStateException in ComTxIPduTxConfirmationValidatorAs4
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	14.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

An illegal state exception is thrown in ComTxIPduTxConfirmationValidatorAs4: Optional.get() cannot be called on an absent value

When does this happen:

During live validation.

In which configuration does this happen:

- Pdu Fan out is configured (1:N Routing Path)
AND
- Minimum delay time is configured
AND
- PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRTransmissionConfirmation is not present for any of the related DestPdus.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00102960	RTE49999 when sender-receiver port is not connected (Windows 10)
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.14.00
Fixed in versions:	1.21.00

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 the configuration contains an unconnected sender-receiver port.

Resolution Description:

Workaround:

Connect the sender-reciver port and create a port access.
The API does not need to be called.

Resolution:

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

ESCAN00102968	COM90005 RuntimeException: Unknown SignalType UINT8_DYN
Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

During generation a RuntimeException is thrown:

COM90005 RuntimeException: Unknown SignalType UINT8_DYN of Signal...

When does this happen:

Always, during generation time.

In which configuration does this happen:

At least one Rx Signal is present with ComSignalType UINT8_DYN
AND
ComReceiveSignalMacroAPI is enabled

Resolution Description:

Workaround:

Disable ComReceiveSignalMacroAPI if possible.

Resolution:

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

ESCAN00102989 a2l: Sector plug&play info missing /end SECTOR tag

Component@Subcomponent:	Cp_Asr4Xcp@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

In the file Xcp.a2l the sector information is written in the form:
/begin SECTOR
...
/end SECTOR

Each /begin SECTOR should be closed by a matching /end SECTOR. This is not the case, only the last /end SECTOR is written.
As a result a Master tool will issue an a2l error.

When does this happen:

Always and immediately

In which configuration does this happen:

When Sector information is used.
/MICROSAR/Xcp/XcpCmdConfig/XcpProgramming/XcpSector

Resolution Description:

Workaround:

The file Xcp.a2l is not used in the ECU. It can be patched manually and stored under a different file name as work around.

Resolution:

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

ESCAN00103000 Compiler error: 'EthSwt_30_Sja1105PQRS_TsProcPool' and 'EthSwt_30_Sja1105PQRS_TsProcQueue' use conflicting section name

Component@Subcomponent:	DrvEthSwitch_Sja1105PQRSAsr@Implementation
First affected version:	1.00.00
Fixed in versions:	5.00.02

Problem Description:

What happens (symptoms):

Compiler complains:
'EthSwt_30_Sja1105PQRS_TsProcPool' uses conflicting section name
'EthSwt_30_Sja1105PQRS_TsProcQueue' uses conflicting section name

because EthSwt_30_Sja1105PQRS_TsProcPool and EthSwt_30_Sja1105PQRS_TsProcQueue are mapped to an INIT section although they are UNINIT data.

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/EthSwt_Sja1105PQRS/EthSwt/EthSwtConfig/EthSwtGlobalTimeSupport == TRUE

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00103030	**Configuration Element 'BswMActionListItem' is duplicated when derived**
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The element BswMActionListItem is duplicated in the DaVinciConfigurator 5.

At initial creation of the configuration the parameter BswMActionListItem is renamed to the name of the ActionListItemRef. After the next data base update the item is derived again with the original name. Therefore two equal items exist.

When does this happen:

When updating the data base of a configuration.

In which configuration does this happen:

If the parameter BswMActionListItemRef is derived and is not compliant with the the action name itself.

Resolution Description:

Workaround:

Remove the created parameters that are not needed manually.

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

Index

ESCAN00051574 [MSR4 only] Compiler warning: statement is unreachable

Component@Subcomponent:	SysService_AsrDet@Implementation
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warns for unreachable statement in API function Det_ReportError

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 with disabled "Enable Extended Debug Support" and DET_AUTOSARVERSION == 4

Resolution Description:

Workaround:

No workaround available.

Resolution:

The described issue is not resolved because there is no technical solution.

ESCAN00065890 Compiler warning: cast discards '__attribute__((noreturn))' qualifier from pointer target type

Component@Subcomponent:	DrvCan_Mpc5700McanLl@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_Mpc5700McanLl@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 occures 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.

ESCAN00067161 Compiler warning: conditional expression is constant

Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	1.00.00
Fixed in versions:	This ticket is not considered for fixing.

Problem Description:

What happens (symptoms):

Compiler warns for '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:

In all configurations with activated optimization "Optimize Const Arrays 2 Define".

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:

De-activate /MICROSAR/Com/ComGeneral/ComOptimizeConstArrays2Define

Resolution:

The described issue is not planned to be corrected.

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

ESCAN00068435 Compiler warning: narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned char

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

Problem Description:

What happens (symptoms):

- Compiler warns for narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned char

Warning occurs in following function:
FUNC(void, NVM_PRIVATE_CODE) NvM_QueueInit(void)

```
 ...
 NvM_JobQueue_at[index].PrevEntry = index - 1u;
```

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:

It happens in all configurations

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 MISRA 2004 - implicit conversion is allowed in this case. Additionally, it is obvious that actually no narrowing occurs (even a compiler could be capable of detection). Result of expression is always in range of [0,254].

Resolution Description:

Workaround:

Just ignore warning.

Resolution:

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

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:	This ticket is not considered for fixing.

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.

ESCAN00083411 Compiler warning: conditional expression is always true/false / statement not reached

Component@Subcomponent:	SysService_Asr4FiM@Implementation
First affected version:	2.00.00
Fixed in versions:	This ticket is not considered for fixing.

Problem Description:

What happens (symptoms):

---

Warnings during compilation of code:

"../../../external/BSW/FiM/FiM.c", line 691: warning (dcc:1606): conditional expression or part of it is always true/false
"../../../external/BSW/FiM/FiM.c", line 753: warning (dcc:1606): conditional expression or part of it is always true/false
"../../../external/BSW/FiM/FiM.c", line 756: warning (dcc:1522): statement not reached
"../../../external/BSW/FiM/FiM.c", line 691: 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:

---

Certain configurations of EventIds:
- depending on number of configured events
- sequence of configured events

Hint:

---

The compiler warning is known and has been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed to allow for dynamic runtime calibration of the event id list.

Resolution Description:

Workaround:

---

No workaround available.

ESCAN00084717 Compiler warning: Several warnings when the component is compiled with Visual Studio

Component@Subcomponent: Tp_AsrTpTcpIp@Implementation_IpV4

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

The compiler issues the following warnings in files with prefix "IpV4":

"unreferenced formal parameter"
"local variable is initialized but not referenced"
"conditional expression is constant"

When does this happen:

The warnings are issued by the compiler during compilation of the code.

In which configuration does this happen:

Visual Studio is used as compiler.
Occurrence of the Warnings depends on module configuration.

Resolution Description:

ESCAN00084718 Compiler warning: Several warnings when the component is compiled with Visual Studio

Component@Subcomponent: Tp_AsrTpTcpIp@Implementation_IpV6

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

The compiler issues the following warnings in files with prefix "IpV6":

"conditional expression is constant"
"unreferenced formal parameter"
"'+=' : conversion from 'int' to 'uint16', possible loss of data"

When does this happen:

The warnings are issued by the compiler during compilation of the code.

In which configuration does this happen:

Visual Studio is used as compiler.
Occurrence of the Warnings depends on module configuration.

Resolution Description:

ESCAN00088061	**BswM_Lcfg.c: warning: 'function' : conversion from 'const BswM_ImmediateUserStartIdxOfModeReqeustMappi 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:

ESCAN00089543 Compiler warning: dead assignment to "errorId" eliminated

Component@Subcomponent:	Nm_Asr4NmIf@Implementation
First affected version:	7.00.00
Fixed in versions:	This ticket is not considered for fixing.

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:	This ticket is not considered for fixing.

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.

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:

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:	This ticket is not considered for fixing.

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:

ESCAN00094232 Compiler warning: "conditional expression is constant"

Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	6.03.00
Fixed in versions:	This ticket is not considered for fixing.

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.

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:	This ticket is not considered for fixing.

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.

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.

ESCAN00098829	Compiler warning: conditional expression is constant
Component@Subcomponent:	Ccl_AsrSmEth@Implementation
First affected version:	4.00.00
Fixed in versions:	This ticket is not considered for fixing.

Problem Description:

What happens (symptoms):

The warning similar to the following is issued by the compiler:

ethsm.c(): 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:

All configurations with only one stack type (TCPIP / AVB only)

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.

Resolution:

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

ESCAN00099046 Compiler warning: unreferenced local variable/ parameter or conversion from 'uint*' to 'TcpIp_*IterType'

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The following warnings may be issued by the compiler depending on the configuration.

TcpIp_Ndp.c(6319): warning C4189: 'ndpConfigIdx' : local variable is initialized but not referenced
TcpIp_Ndp.c(5295): warning C4100: 'IpAddrCountPtr' : unreferenced formal parameter
TcpIp_Ndp.c(2671): warning C4189: 'ndpConfigIdx' : local variable is initialized but not referenced
TcpIp_IpV6.c(7122): warning C4100: 'IpCtrlIdx' : unreferenced formal parameter
TcpIp_IpV6.c(6555): warning C4189: 'ndpConfigIdx' : local variable is initialized but not referenced

The following warning is only issued if the type uint8_least or uint16_least is defined to a type smaller than uint32:

TcpIp_Arp.c(1611): warning C4242: 'function' : conversion from 'uint32' to 'TcpIp_ArpTableEntryIterType', 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:

TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig/TcpIpNdpNeighborUnreachabilityDetectionEnabled == FALSE
OR
TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig/TcpIpNdpInverseNaEnabled == FALSE
OR
TcpIp/TcpIpGeneral/TcpIpIpV6General/TcpIpDhcpV6ClientEnabled == FALSE
OR
TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpSlaacConfig/TcpIpNdpSlaacDadNumberOfTransmissions == FALSE

Hint:

The compiler warnings are known and have been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed due to 2018-08-01

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:	15.01.00

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.

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_CallTlFunction_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.

ESCAN00100160 Compiler warning: Order of volatile access is undefined in this statement

Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	13.00.00
Fixed in versions:	This ticket is not considered for fixing.

Problem Description:

What happens (symptoms):

A compiler warning is shown which indicates that order of volatile accesses are undefined.

For example in an IAR compiler for ARM 8.20: Warning[Pa082]: 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 which configuration does this happen:

Happens in all configurations.

Resolution Description:

Workaround:

The compiler warning can be ignored as the order of access of the respective volatile variables have no effect on the functionality.

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:	This ticket is not considered for fixing.

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.

ESCAN00100219	Compiler warning: Explicit Cast Discards Volatile Qualifier
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	6.02.00
Fixed in versions:	This ticket is not considered for fixing.

Problem Description:

What happens (symptoms):

The compiler states one or more of the following warnings:

In Dem_Mem_Implementation.h
line 1323: warning (dcc:1824): explicit cast from 'ptr-to-volatile Dem_Event_InternalStatusType' to 'Dem_DataPtrType' discards volatile qualifier
line 1436: warning (dcc:1824): explicit cast from 'ptr-to-volatile Dem_Cfg_MemoryCommitNumberType' to 'Dem_DataPtrType' discards volatile qualifier

In Dem_EnableCondition_Implementation.h
line 744: warning (dcc:1824): explicit cast from 'ptr-to-volatile Dem_Cfg_EnableConditionGroupStateType' to 'Dem_DataPtrType' discards volatile qualifier

When does this happen:

At compile time.

In which configuration does this happen:

All

Resolution Description:

Workaround:

Disable Compiler Warning.

Resolution:

Warning will not be fixed.

ESCAN00100871 Compiler warning: Variable is possibly uninitialized

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

Problem Description:

What happens (symptoms):

Compiler warns that variable is possibly uninitialized (W507 variable lNrc is possibly uninitialized).

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:

Configuration independent.

Resolution Description:

Workaround:

Ignore warning.

Resolution:

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

ESCAN00100903 Compiler warning: last line of Dem_EnableCondition_Interface.h ends without a newline

Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	13.03.00
Fixed in versions:	16.01.00

Problem Description:

What happens (symptoms):

Compiler warns about missing newline:

../../../external/bsw/dem/Dem_EnableCondition_Interface.h:698:120: error: no newline at end of file [-Werror,-Wnewline-eof]

When does this happen:

The warning is issued by the compiler during compilation of the code

In which configuration does this happen:

----

all

Resolution Description:

Workaround:

Ignore the warning

Resolution:

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

ESCAN00100904	Compiler warning: last line of Dem_Cfg_Definitions.h ends without a newline
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	15.04.00
Fixed in versions:	16.01.00

Problem Description:

What happens (symptoms):
```
---------------------------------------------------------------------
```
Compiler warns about missing newline:

../../../external/bsw/dem/Dem_Cfg_Definitions.h:5251:120: error: no newline at end of file [-Werror,-Wnewline-eof]

When does this happen:
```
---------------------------------------------------------------------
```
The warning is issued by the compiler during compilation of the code

In which configuration does this happen:
```
---------------------------------------------------------------------
```
all

Resolution Description:

Workaround:
```
---------------------------------------------------------------------
```
Ignore the warning

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

ESCAN00100905 Compiler warning: last line of Dem_MemState_Implementation.h ends without a newline

Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	15.05.00
Fixed in versions:	16.01.00

Problem Description:

What happens (symptoms):

Compiler warns about missing newline:

../../../external/bsw/dem/Dem_MemState_Implementation.h:1085:120: error: no newline at end of file [-Werror,-Wnewline-eof]

When does this happen:

The warning is issued by the compiler during compilation of the code

In which configuration does this happen:

----

all

Resolution Description:

Workaround:

Ignore the warning

Resolution:

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

ESCAN00101342 Compiler warning: unreferenced formal parameter

Component@Subcomponent:	Cp_XcpOnTcpIpAsr@Implementation
First affected version:	3.00.00
Fixed in versions:	4.00.00

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:

----

all configurations

Resolution Description:

Workaround:

Set V_ENABLE_USE_DUMMY_STATEMENT in a user.cfg file or a globally accessible include header.

Resolution:

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

ESCAN00101597 Compiler warning/MISRA deviation: MISRA-C:2012 Rule 8.2: Keyword void is missing for empty parameter list.

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

Problem Description:

What happens (symptoms):

The generated file IoHwAb_30.c contains a MISRA violation. The explicit void keyword for empty parameter list has been omitted. This message should occur in MISRA report or also as compiler warning. The warning has no impact on application.

When does this happen:

The warning is issued in a MISRA report or 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 no parameter are configured for an SRPortInterface or an CSPortInterface.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00101688	Compiler warning: conversion from 'const Dcm_CfgDidMgrAbstractOpRefType' to 'Dcm_CfgDidMgrDynDidHandleOptType', possible loss of data
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	5.00.00
Fixed in versions:	11.04.00

Problem Description:

What happens (symptoms):

A compiler warning for Dcm.c is issued: conversion from 'const Dcm_CfgDidMgrAbstractOpRefType' to 'Dcm_CfgDidMgrDynDidHandleOptType', 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:

- Service 0x2C is supported (in Dcm_Cfg.h: #define DCM_SVC_2C_SUPPORT_ENABLED == STD_ON)

AND

- There are more than 255 DID operations (e.g. a single DID may support read data, write data, read scaling data, IO control operations, etc.)

AND

- There are less than 256 dynamically defined DIDs configured

Resolution Description:

Workaround:

Ignore the warning since no real danger of data loss.

Resolution:

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

ESCAN00102090 Compiler warning: Tasking issues "ctc W555: [..] function declared with _Noreturn or __noreturn__ may return"

Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

A warning is issued
e.g.
ctc W555: ["D:\usr\development\Components\Os\Core\Os_CoreGen7\base\Implementation\trunk\kernel/Os_Hook.c" 490/1] function declared with _Noreturn or __noreturn__ may return
The affected function is "Os_HookReturn"

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 warning is issued if control flow simplification is disabled by the optimization strategy "F".

In which configuration does this happen:

The warning is issued independently of the configuration.

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 wrong interpretation of the code by the compiler.
The function does not return. So the compiler pragma __noreturn__ is correct.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

Hint:
If it is possible to use control flow simplification, the warning should not occur.
Use strategy "f" instead of "F" for the optimization options.

ESCAN00102097 Compiler warning: if statement does not affect control flow

Component@Subcomponent:	If_AsrIfSoAd@Implementation
First affected version:	12.00.00
Fixed in versions:	13.00.00

Problem Description:

What happens (symptoms):

Compiler warns for "if statement does not affect control flow".

In SoAd.c in SoAd_CopyTxData():
The code snippet below the affected if statement is always executed. In case if statement is false an invalid "sockIdx" is used.
This is an issue when input parameter "SocketId" is invalid. In configuration described below it is assumed that the input parameter values are always valid.
Therefore, this issue is a compiler warning which does not lead to runtime issues.

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:

Issue happens when SoAd/SoAdGeneral/SoAdSafeBswChecks and SoAd/SoAdGeneral/SoAdDevErrorDetect are disabled.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102317 Compiler warning: statement not reached/ unreachable code

Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	2.29.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

The compiler reports the following warning:
* statement not reached
* unreachable code

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 independently from the configuration. It has been observed with compiler versions Diab and Tasking.

Hint:

The compiler warning is known to occur at the places in the source code which are marked with the comment COMP_WARN_OS_UNREACHABLE_CODE.
At these places, it has been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed due to wrong interpretation of the code by the compiler.
Some compilers do not recognize no-return functions properly.

Resolution Description:
Workaround:

No workaround available.
The warning has been analyzed and can be accepted or suppressed my the users build environment.

Resolution:

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

ESCAN00102514 Compiler warning: explicit cast from 'ptr-to-volatile uint32' to 'uint32' discards volatile qualifier

Component@Subcomponent:	DrvEth_Tc3xxEthAsr@Implementation
First affected version:	0.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler throws warning "explicit cast from 'ptr-to-volatile uint32' to 'uint32' discards volatile qualifier".

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 if WindRiver DiabData version 5.9.6.2 is used.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00102824 Compile warning: "#186-D: pointless comparison of unsigned integer with zero"

Component@Subcomponent:	DrvCan__coreAsr@Implementation
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The compiler reports a warning "#186-D: pointless comparison of unsigned integer with zero".

There is no effect on the functionality.

When does this happen:

During compile time.

In which configuration does this happen:

No Tx Objects are configured.

Resolution Description:

ESCAN00102902 [Compiler Warning] Warning C4242 possible loss of data in TcpIp_Ndp.c

Component@Subcomponent:	Tp_AsrTpTcpIp@Implementation
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The following compiler warning is observed during compilation
1>..\..\..\..\..\SIP\external\BSW\TcpIp\TcpIp_Ndp.c(6925): warning C4242: '=': conversion from 'TcpIp_IpV6NeighborCacheEntryIterType' to 'uint32', possible loss of data is

When does this happen:

During compilation

In which configuration does this happen:

TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config is enabled

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

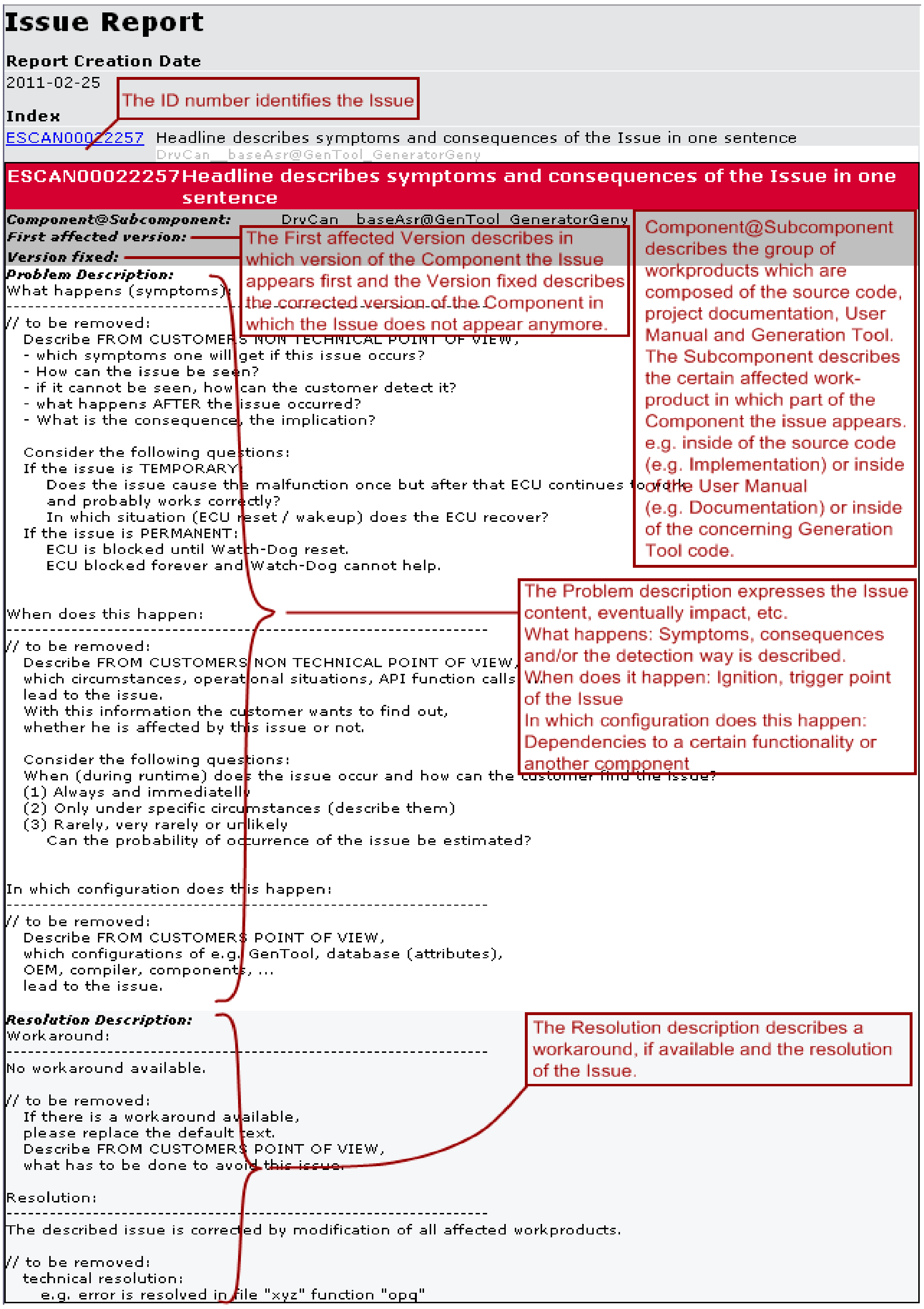
3. New Issues for Information

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

ESCAN00022257 Headline describes symptoms and consequences of the Issue in one sentence

DrvCan__baseAsr@GenTool_GeneratorGeny

ESCAN00022257Headline describes symptoms and consequences of the Issue in one sentence

Component@Subcomponent: DrvCan__baseAsr@GenTool_GeneratorGeny

First affected version:

Version fixed:

Problem Description:

```
What happens (symptoms):
-------------------------------
// 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?

   Consider the following questions:
   If the issue is TEMPORARY:
      Does the issue cause the malfunction once but after that ECU continues to
      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.


When does this happen:
----------------------------------------------------------------
// 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.

   Consider the following questions:
   When (during runtime) does the issue occur and how can the customer find the issue?
   (1) Always and immediatelly
   (2) Only under specific circumstances (describe them)
   (3) Rarely, very rarely or unlikely
      Can the probability of occurrence of the issue be estimated?


In which configuration does this happen:
----------------------------------------------------------------
// to be removed:
   Describe FROM CUSTOMERS POINT OF VIEW,
   which configurations of e.g. GenTool, database (attributes),
   OEM, compiler, components, ...
   lead to the issue.
```

Resolution Description:

```
Workaround:
----------------------------------------------------------------
No workaround available.

// 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.

Resolution:
----------------------------------------------------------------
The described issue is corrected by modification of all affected workproducts.

// to be removed:
   technical resolution:
     e.g. error is resolved in file "xyz" function "opq"
```

The ID number identifies the Issue

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.

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.

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

The Resolution description describes a workaround, if available and the resolution of the Issue.

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