Issue Report

License Number	Customer
CBD1400351	Nexteer Automotive Corporation Package: GM Global A / FR - ECU Product "Steering Systems" Micro: R7F701311 Compiler: GHS 2013.5.5

Maintenance Expiry Date

2015-09-01

SIP Delivery Date	SIP Version
2015-10-23	13.00.00

SLP	Delivery Number
GM Global A / FR	D03

Report Creation Date

2015-10-23

Contact

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

Table of Contents

1. Introduction

1.1 Resolving Issues

Reported issues are not necessarily fixed automatically by the next update delivery. If some of the reported issues shall be fixed, please contact Vector to establish an agreement about issues that shall be fixed in upcoming deliveries. Please note that Vector may fix additional 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:

- **Runtime Issues without Workaround:** Runtime issues without a workaround require an update of the basic 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 basic 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 customer as the basic software usage and its configuration is not known by Vector. The risk of change has also to be taken into account.
- **Compiler Warnings:** As a service we report the known compiler warnings. The occurrence of a compiler warning may depend on the used configuration and compiler settings.
- **Apparent Issues:** Apparent issues are detected immediately when using the basic software. If an issue does not show up while working with the basic software the ECU project is not affected by the issue. Apparent issues may or may not have workarounds.

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 Runtime Issues without Workaround

The lists contain issues that have been detected since the last report and which could not be excluded based on the use-cases defined in the questionnaire (see chapter ‘New Issues for Information’).

ESCAN00076676	'initializing' : truncation from 'SomeBigDataType' to 'SomeSmallerDataType'
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	4.00.01

Problem Description:
What happens (symptoms):
---
a compiler warning like the following occurs:
warning C4305: 'initializing' : truncation from 'SomeBigDataType' to 'SomeSmallerDataType'
when you are using the code

Nevertheless, the consequence for the ECU is unpredictable.

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 generated index based data types which are NOT changeable at postbuild time.
AND
the code has been generated by Com or IpduM or PduR or LdCom or BswM.

Resolution Description:
Workaround:

Il_AsrComCfg5: Configure ComMinimizeNumericalDataTypes to NONE.
Il_AsrIpduMCfg5: Configure IpduMMinimizeNumericalDataTypes to NONE.
Gw_AsrPduRCfg5: Configure PduRMinimizeNumericalDataTypes to NONE.
Il_AsrLdCom: No workaround available.
SysService_Asr4BswMCfg5: No workaround available.
Tp_Asr4TpLin: Not affected.
If_Asr4IfLin: Not affected.
If_AsrIfCan: Not affected.
SysService_Asr4EcuM: Not affected.
Ccl_Asr4ComMCfg5: Not affected.
Cdd_AsrCddCfg5: Not affected
Cdd_AsrCddCfg5: Not affected
EcuC_AsrEcuC: Not affected

Resolution:

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

ESCAN00079240	Undefined ECU behavior due to invalid index access in 0:* and 1:* Relations
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	5.00.01

Problem Description:

ESCAN00079240	Undefined ECU behavior due to invalid index access in 0:* and 1:* Relations

What happens (symptoms):

Undefined ECU behavior due to invalid index access.

A loop in the code executes code for indexes but the intention is to execute no code.
The issue is PERMANENT and the problem can produce DET errors and a suspicious long runtime of code.

e.g.

typedef uint16 Com_DefRxSigInfoEndIdxOfDefRxPduInfoType;
typedef uint8 Com_DefRxSigInfoStartIdxOfDefRxPduInfoType;

#define COM_NO_DEFRXSIGINFOENDIDXOFDEFRXPDUINFO 65535U
#define COM_NO_DEFRXSIGINFOSTARTIDXOFDEFRXPDUINFO 255U

Com_DefRxSigInfoEndIdxOfDefRxPduInfoType idxRxSigInfo =
Com_GetDefRxSigInfoStartIdxOfDefRxPduInfo(idxRxPduInfo);
for(; idxRxSigInfo < Com_GetDefRxSigInfoEndIdxOfDefRxPduInfo(idxRxPduInfo); idxRxSigInfo++)

the loop executes code from idxRxSigInfo 255 to 65535

When does this happen:

The issue occurs always and immediately.

In which configuration does this happen:

Any configuration where the following multiline regex matches 1 or more times in the generated files:
/typedef\s([A-Za-z0-9_]+)\s+([A-Za-z0-9_]+)(?:Start|End)IdxOf([A-Za-z0-9_]+Type)[\s\S]+typedef\s+(?!\1)([A-Za-z0-9_]+)\s+\2(?:Start|End)IdxOf\3/gm

affected modules if the regex matches:
Ccl_Asr4SmLin@GenTool_GeneratorMsr[3.00.00]
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.02]
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.01]
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.00]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.03]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.02]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.01]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.00]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.02]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.01]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.00]
Il_AsrLdCom@GenTool_GeneratorMsr[1.00.00]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.02]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.01]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.00]
SysService_Issm_Volvo_ab@GenTool_GeneratorMsr[1.01.00]

not affected modules:
all versions of Cp_AsrXcp@GenTool_GeneratorMsr
all versions of Cdd_AsrCddCfg5@GenTool_GeneratorMsr
all versions of EcuC_AsrEcuC@GenTool_GeneratorMsr

ESCAN00079240	Undefined ECU behavior due to invalid index access in 0:* and 1:* Relations

Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr[3.00.00]
If_Asr4IfLin@GenTool_GeneratorMsr[3.00.00]
If_AsrIfCan@GenTool_GeneratorMsr[3.03.00]
Tp_Asr4TpLin@GenTool_GeneratorMsr[3.00.00]
SysService_Asr4EcuM@GenTool_GeneratorMsr[3.00.02]
SysService_Asr4EcuM@GenTool_GeneratorMsr[3.00.01]
SysService_Asr4EcuM@GenTool_GeneratorMsr[3.00.00]

Resolution Description:
Workaround:
---
for
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.02]
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.01]
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.00]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.03]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.02]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.01]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.00]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.02]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.01]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.00]
is a workaround available. Configure <MSN>MinimizeNumericalDataTypes to NONE in <MSN>General.

for
Ccl_Asr4SmLin@GenTool_GeneratorMsr[3.00.00]
Il_AsrLdCom@GenTool_GeneratorMsr[1.00.00]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.02]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.01]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.00]
SysService_Issm_Volvo_ab@GenTool_GeneratorMsr[1.01.00]
other modules is no workaround available, but the condition is generated quite rarely. Check the generated code with:
/typedef\s([A-Za-z0-9_]+)\s+([A-Za-z0-9_]+)(?:Start|End)IdxOf([A-Za-z0-9_]+Type)[\s\S]+typedef\s+(?!\1)([A-Za-z0-9_]+)\s+\2(?:Start|End)IdxOf\3/gm

Resolution:

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

ESCAN00079274	Undefined ECU behavior due to invalid index access in optional indirections pointing to IConstStructWithUnusedIndexes
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	5.00.01

Problem Description:
What happens (symptoms):

Undefined ECU behavior due to invalid index access.
The code code executes code for indexes but the intention is to execute no code.
The issue is PERMANENT and the problem can produce DET errors and a suspicious long runtime of code.

When does this happen:

The issue occurs always and immediately.

In which configuration does this happen:

No module is up to now affected.

not affected modules:
all versions of Cp_AsrXcp@GenTool_GeneratorMsr
all versions of Cdd_AsrCddCfg5@GenTool_GeneratorMsr
all versions of EcuC_AsrEcuC@GenTool_GeneratorMsr
Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr[3.00.00]
Ccl_Asr4SmLin@GenTool_GeneratorMsr[3.00.00]
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.02]
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.01]
Gw_AsrPduRCfg5@GenTool_GeneratorMsr[5.00.00]
If_Asr4IfLin@GenTool_GeneratorMsr[3.00.00]
If_AsrIfCan@GenTool_GeneratorMsr[3.03.00]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.03]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.02]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.01]
Il_AsrComCfg5@GenTool_GeneratorMsr[4.00.00]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.02]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.01]
Il_AsrIpduMCfg5@GenTool_GeneratorMsr[4.00.00]
Il_AsrLdCom@GenTool_GeneratorMsr[1.00.00]
Tp_Asr4TpLin@GenTool_GeneratorMsr[3.00.00]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.02]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.01]
SysService_Asr4BswMCfg5@GenTool_GeneratorMsr[3.00.00]
SysService_Asr4EcuM@GenTool_GeneratorMsr[3.00.02]
SysService_Asr4EcuM@GenTool_GeneratorMsr[3.00.01]
SysService_Asr4EcuM@GenTool_GeneratorMsr[3.00.00]
SysService_Issm_Volvo_ab@GenTool_GeneratorMsr[1.01.00]

Resolution Description:

ESCAN00079274	Undefined ECU behavior due to invalid index access in optional indirections pointing to IConstStructWithUnusedIndexes

Workaround:

No workaround available.

Resolution:

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

ESCAN00085552	Missing Data after DTC re-occurrence

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

Problem Description:
What happens (symptoms):

(Typically) Some extended data records are not reported for a DTC.
The missing records are statistic values like number of failed cycles, maximum FDC and similar values.

When does this happen:

After an event memory entry used as aging counter is used to store an active (confirmed and failed) DTC.

To use event entries as aging counter, the event memory must be completely filled, and the number of confirmed DTCs has to be higher than the number of memory entries.
In this case, entries freed due to aging (or ClearSingleDTC) are re-used as aging counter for DTCs that are not stored but confirmed, and are reported 'tested passed'.

If a DTC holding such an entry is reported 'failed' again, the extended records are not re-recorded. This situation will last until the DTC in question has been removed from memory (aging, clear DTC, or displacement)

In which configuration does this happen:

Configurations with DemResetConfirmedBitOnOverflow == FALSE
AND
DemEventDisplacementStrategy != DEM_DISPLACEMENT_NONE

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00085612 Out of bounds memory write access

Component@Subcomponent: Diag_Asr4Dem@Implementation
First affected version: 6.00.00
Fixed in versions:

Problem Description:
What happens (symptoms):

The Dem accesses (writes) invalid memory.
Invalid data is read, interpreted as pointer, and dereferenced for write access.

When does this happen:

When reporting an Event with AgingTarget == 0 as 'passed', while the event is not active.
If this event is later reported 'failed', the issue is triggered.

In which configuration does this happen:

Option DemAgingRetainEnvironmentalData is enable (DEM_CFG_AGING_RETAIN_MEMORY_ENTRY == STD_ON)
AND
Aging Target 0 is used for some events (DemAgingCycleCounterThreshold == 0)

Resolution Description:
Workaround:

Instead of using AgingCycleCounterThreshold == 0, you can delete the DTC (API Dem_ClearDTC) once it tests OK.
If the setting of DemClearDTCLimitation does not include clearing single DTCs, there is no workaround.

Resolution:

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

2.2 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 customer as the basic software usage and its configuration is not known by Vector. Thereby the risk of change has also to be taken into account.

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 5 issues. This ticket is a general information and not an issue.

Issues of the DaVinci Configurator 5 tool are not part of the active issue reporting (i.e. this report). The DaVinci Configurator 5 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 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.

ESCAN00080492	Fee cannot read data
Component@Subcomponent:	If_AsrIfFee@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	3.01.01

Problem Description:

What happens (symptoms):

Fee cannot read data, which previously have been written successfully.
It delivers result MEMIF_BLOCK_INVALID.

When does this happen:

It happens once the ECU was restarted after data have been written (either intended or unintended).
Writes succeed; read requests issued before reset do deliver data as expected.

Since this issue almost completely depends on configuration, this issue is easy to reproduce.

In which configuration does this happen:

It happens for all blocks within a partition whose link table ends (the dynamic data area starts) beyond the first fourth of a sector.

This end offset is: Sector Header + Link-TableSize = ALIGN(8Bytes, AddressAlignment) + ALIGN(MIN(MAX(BlockId) + 1, MaxLinkTableSize) * ALIGN(8Bytes, WriteAlignment), AddressAlignment).

ALIGN(a, b) --> round up a to next integral multiple of b.

Actually it is intended to reject such configurations, i.e. the configuration tool should issue an error message. But such validation is missing.

Resolution Description:

Workaround:

Manually make sure that LinkTable size does not exceed one fourth of the smaller logical sector. Use the formula given in description.

If necessary, re-assign a partion's blocks' IDs. If that's not possible, reduce FeeMaxLinkTableSize. The latter option affects all partitions, and furthermore may result in performance penalties (FEE's job processing time, number of MainFunction calls) due to increased Block Search efforts.

Resolution:

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

ESCAN00081636	Out-of-Flash Read access in FBL mode
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	8.00.00
Fixed in versions:	8.00.11

Problem Description:

What happens (symptoms):

FEE may issue Fls_Read requests out of bounds.

Depending on system/system configuration this may result in a Exception, or a DET-Error reported by Fls.
It also may remain undetected, in such case no misbehavior occurs at all.

When does this happen:

It may happen during run-time, if FEE is performing a sector switch and a Flash sector is completely filled - chunks were allocated up to the last possible address.

In which configuration does this happen:

It happens only if FEE was configured in FBL mode ("incomplete Configuration"), or - in case DataConversion Feature is available - if a data conversion process was started.

Additionally it only becomes an issue, if the first 8bytes after a sector end are not accessible by Fls (Fls_Read).
If both sector are consecutive, this may only be the case for the upper sector.

Resolution Description:

Workaround:

In general there are four possible workarounds.
1. Disable FLS’s Development error detection. In this case it needs to be ensured that system does not run into an exception (e.g. Illegal Access). This is possible if the address space beyond DFlash is valid, and can be read safely (without side-effects).
2. Enable Fls’ Development Error detection, but make sure, SW keeps running. If necessary, filter for FLS'S module ID, Fls_Read's API ID and for error codes FLS_E_PARAM_ADDRESS and FLS_E_PARAM_LENGTH.
(It is not sufficient to filter for error codes only, because typically their numeric values are not unique across different modules.
3. Decrease FEE’s upper sector size (by one "Fee Address Alignment" entity). However, this change results in incompatible Flash layouts, i.e. existing data may get lost.
4. Change FLS’s sector configuration; make sure FEE does not access the last Fls Sector. This may not be possible with all platforms/Fls configurations.

Resolution:

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

ESCAN00085093	Wrong return value for unconnected Rte_Read
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

Rte_Read API returns RTE_E_OK instead of RTE_E_UNCONNECTED for unconnected ports.

When does this happen:

Always during runtime.

In which configuration does this happen:

This happens when a port is unconnected and when API usage by address and usage of indirect API is not enabled for the port.
Moreover this only happens for source code SWCs.

Resolution Description:

Workaround:

Enable API usage by address for the unconnected port.

Resolution:

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

ESCAN00085191	EIINT exception table is not aligned on 512Byte boundary
Component@Subcomponent:	Os_PlatformRh850Gen6@Generator
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

---

1) An unhandled core exception is called instead of the configured core exception handler
2) The CPU branches to program code which is mapped before the core exception table and the result is unpredictable
3) An unhandled EIINT exception is called instead of the configured EIINT interrupt handler
4) The CPU branches to program code which is mapped before the EIINT exception table and the result is unpredictable

When does this happen:

This happens during runtime when a core exception or an EIINT interrupt occurs.

In which configuration does this happen:

This happens in all configurations.

Resolution Description:

Workaround:

Add 512Byte alignment to linker file:

Example

```
SECTIONS
{
 .ROM_start :>ROM_0000

 .osExceptionVectorTable_c0 (align 512) :>.

 .osEIINTVectorTable_c0 (align 512) :>.

 .osExceptionVectorTable_c1 (align 512) :>.

 .osEIINTVectorTable_c1 : (align 512) >.
```

Resolution:

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

ESCAN00085391	DEM95101 "Non-optimal handle ID configuration" info message hints an issue in the configuration
Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	7.00.00

Problem Description:

What happens (symptoms):

- DaVinci Configurator5 shows the [Info] message in its validation view (for multi variant projects: in the tab "General"):

DEM95101 Non-optimal handle ID configuration. (1 message)
DEM95101 Non-optimal handle ID configuration. A more optimal configuration with 0 instead of 4 gaps could be used. An optimized handle ID configuration can reduce the amount of used hardware resources.
Recalculate all handle ID values.

- In the "Basic Editor" when showing the Dem/DemConfigSet/DemEventParameter in the grid view (sort by column "EventId"), the sequence of ID values has gaps for example ID values 1-5 and 7-20 exist but "6" is missing.

- In the generated code, the size of table Dem_C_EventTable[] in file Dem_Lcfg.h is smaller than the largest EventId value in the symbolic names (see #defines "Event IDs [symbolic name value]" in Dem_Lcfg.h.)

- During runtime, some DTCs are not set as expected, others unexpectedly change their state.
- With active feature "Development Error Detect" you get DET calls.
- Without feature "Development Error Detect" you see unexpected behavior.

When does this happen:

Always and immediately

In which configuration does this happen:

All configurations, where the sequence of EventIDs has gaps.

Resolution Description:

Workaround:

Either trigger in DaVinci Configurator5 the button "Solve All" to execute all preferred solving actions
or in the Validation result DEM95101 execute the solving action "Recalculate all handle ID values".

Resolution:

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

ESCAN00085471	Rte_Read / Rte_IStatus don't return RTE_E_NEVER_RECEIVED when the first received values are invalid and if the init value equals the invalid value
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

Rte_Read / Rte_IStatus don't return RTE_E_NEVER_RECEIVED when the first received values are invalid and if the init value equals the invalid value in the configuration.
In that case the return value is RTE_E_INVALID

When does this happen:

This happens before the first valid value was received and after the first invalid value was received.
When the first valid value was received the return values of Rte_Read and Rte_IStatus are correct again.

In which configuration does this happen:

- when invalidation is configured for a data element and the invalidation handling is configured to "keep".
- handle never received is configured to true.
- the invalid value equals the initial value

Resolution Description:

Workaround:

Do not use the configuration described above.

or implement the "never received" mechanism manually e.g. by usage of an "on data reception" triggered runnable and a call of Rte_Read / Rte_IRead inside this runnable. If the received value is no longer equal with the initial value a valid value was received and the manual implemented "never received flag" can be cleared.

Resolution:

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

ESCAN00085604	Statistic values 'Failed Cycles counter', 'Consecutive Failed Cycles Counter' do not update correctly
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	4.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Statistic values 'Failed Cycles counter' and / or 'Consecutive Failed Cycles Counter' are not persistet into NV-Ram

Typically these counters are mapped to extended data records. The counter is reported correctly at first, but after ECU reset a lower value as expected is returned.

When does this happen:

When reporting an event as 'failed' which is currently in state 'not tested && failed'
(This state is typical after operation cycle restart) and then resetting the ECU.

In which configuration does this happen:

Configurations using
internal data element 'DEM_FAILED_CYCLES'
OR
internal data element 'DEM_CONSECUTIVE_FAILED_CYCLES'

AND
Option DemStatusBitStorageTestFailed is FALSE

Resolution Description:

ESCAN00085604	Statistic values 'Failed Cycles counter', 'Consecutive Failed Cycles Counter' do not update correctly

Workaround:

In some cases enabling Option DemStatusBitStorageTestFailed to TRUE might solve the issue.
This option has side effects, as all failed results are interpreted as occurrence.

Alternatively, integration code after Dem_Shutdown can verify if the respective counters have changed and report the blocks notification to the NvM.

Step 1:
During the ECU startup, after restoring the NV contents, create a copy the cycle counters from the Dem NV data:
for failed cycles: Dem_PrimaryEntry_0.FailedCycleCounter
 Dem_PrimaryEntry_1.FailedCycleCounter
 ...
 Dem_PrimaryEntry_<N>.FailedCycleCounter

for consecutive failed cycles:
 Dem_PrimaryEntry_0.ConsecutiveFailedCycleCounter
 Dem_PrimaryEntry_1.ConsecutiveFailedCycleCounter
 ...
 Dem_PrimaryEntry_<N>.ConsecutiveFailedCycleCounter

After Dem_Shutdown, check whether the copies still contain the values stored in Dem_PrimaryEntry_<N>
If the values differ, mark the block as modified (In Autosar systems using NvM_SetRamBlockStatus).

To simplify matching the block data and NV block IDs, you can use
Arrays Dem_Cfg_MemoryDataPtr[] and Dem_Cfg_MemoryBlockId[].

 ((Dem_Mem_EntryPtrType)Dem_Cfg_MemoryDataPtr[2])->FailedCycleCounter
 ((Dem_Mem_EntryPtrType)Dem_Cfg_MemoryDataPtr[3])->FailedCycleCounter
 ...
the corresponding NvM Block Ids are stored with the same index
 Dem_Cfg_MemoryBlockId[2]
 Dem_Cfg_MemoryBlockId[3]

The correct indices for memory blocks can also be taken from the configuration.
Primary memory:
 Dem_Cfg_MemoryPrimaryIndex()
Secondary memory:
 Dem_Cfg_MemorySecondaryIndex()

e.g.
```
uint8 i;
uint8 backup[Dem_Cfg_GlobalPrimarySize()];
for (i = 0; i < Dem_Cfg_GlobalPrimarySize(); ++i)
{
 backup[i] = ((Dem_Mem_EntryPtrType)Dem_Cfg_MemoryDataPtr[i +
Dem_Cfg_MemoryPrimaryIndex()])->FailedCycleCounter;
}
for (i = 0; i < Dem_Cfg_GlobalPrimarySize(); ++i)
{
 if (backup[i] != ((Dem_Mem_EntryPtrType)Dem_Cfg_MemoryDataPtr[i +
```

ESCAN00085604	Statistic values 'Failed Cycles counter', 'Consecutive Failed Cycles Counter' do not update correctly

```
Dem_Cfg_MemoryPrimaryIndex()])->FailedCycleCounter)
 {
 NvM_SetRamBlockStatus(Dem_Cfg_MemoryBlockId[i + Dem_Cfg_MemoryPrimaryIndex()],
TRUE);
 }
}
```

Resolution:

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

ESCAN00085687	Wrong event handling for Data Reception Triggers for NV data elements mapped to NvBlockDescriptors of complex type
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.01.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:
What happens (symptoms):

A runnable is triggered by a Data Reception Trigger although the data element referenced in the trigger was not written OR
a runnable is NOT triggered although the data element was written and a Data Reception Event should be set.

When does this happen:

This happens during runtime if the configuration is as described below.

In which configuration does this happen:

This happens if a runnable should be triggered on Data Reception of a NV data element that is mapped to a NvBlockDescriptor of complex type and there are additional NvBlockDataMappings for the whole complex type or for other elements.

Resolution Description:
Workaround:

If Data Reception Triggers for NV data elements mapped to NvBlockDescriptors of complex type are used,
only map the whole complex NvBlockDescriptor or the individual elements and do not use multi mapping for these blocks.

Resolution:

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

ESCAN00085738	Wrong initialization of NvBlock mirror buffer in Rte_NvMNotifyInitBlock function
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.08.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

The Rte_NvMNotifyInitBlock function in Rte.c contains a line

1. Rte_ = ; OR

2. Rte_<ComponentTypeName>_<NvBlockDescriptorName> = ; OR

3. the NvBlock mirror buffer is initialized with the RamBlockInitValue instead of the RomBlockInitValue.

When does this happen:

This happens during compilation (case 1 and 2) or during runtime (case 3) in the configurations described below.

In which configuration does this happen:

A NV block descriptor has a port assignment with the role "NvMNotifyInitBlock" and

1. the NV block descriptor has a NV block data mapping to an unconnected NV data port OR

2. the NV block descriptor has a NV block data mapping to a connected NV data port and the NV block descriptor is of a primitive type and uses a RomBlock, but has no RamBlockInitValue OR

3. the NV block descriptor has a NV block data mapping to a connected NV data port and the NV block descriptor is of a primitive type and uses a RomBlock and the RamBlockInitValue is different than the RomBlockInitValue.

Resolution Description:

Workaround:

Do any of the following:
- remove the mapping to the NV data port, if the port is unconnected
- connect the NV data port, if the port is unconnected
- remove the port assignment
- use the same init values for the RamBlock and the RomBlock, if the NV block descriptor is of primitive type

Resolution:

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

ESCAN00085877	Dynamic length signals cannot be received
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

A dynamic length signal is not received or received incompletely.

When does this happen:

During runtime.

In which configuration does this happen:

When dynamic length signals are configured.

Resolution Description:

ESCAN00085877 Dynamic length signals cannot be received

Workaround:

Replace the RTE COM callbacks for dynamic length signals in the COM configuration with custom callbacks
that initialize the
Rte_Q_<ID>_length array with the maximum length of the dynamic length signal
and then call the original RTE callback.

```
#define RTE_CORE
#include "Rte_Type.h"

#include "Rte_Cbk.h"

#define RTE_Q_cpTxRxEcu2_piSRP1_ext_deArrayUint8 (1)


extern Rte_QRamInfoType Rte_QRamInfo[];

extern const Rte_QRomInfoType Rte_QRomInfo[];

extern uint16 Rte_Q_cpTxRxEcu2_piSRP1_ext_deArrayUint8_length[];


FUNC(void, RTE_CODE) Fix_Rte_COMCbk_sigExt_a3308dc4_In(void)
{
 P2VAR(Rte_QRamInfoType, AUTOMATIC, RTE_VAR_NOINIT) pRamI =
&(Rte_QRamInfo[RTE_Q_cpTxRxEcu2_piSRP1_ext_deArrayUint8]);
 P2CONST(Rte_QRomInfoType, AUTOMATIC, RTE_CONST) pRomI =
&(Rte_QRomInfo[RTE_Q_cpTxRxEcu2_piSRP1_ext_deArrayUint8]);

 if (pRamI->Rte_ElementCtr < pRomI->Rte_MaxElements)
 {
 uint8 writeCtr = pRamI->Rte_WriteCtr;
 if (writeCtr < (pRomI->Rte_MaxElements - 1U))
 {
 ++writeCtr;
 }
 else
 {
 writeCtr = 0U;
 }
 Rte_Q_cpTxRxEcu2_piSRP1_ext_deArrayUint8_length[writeCtr] = pRomI->Rte_BytesPerElement;
 Rte_COMCbk_sigExt_a3308dc4_In();
 }
}
```

Resolution:

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

2.3 Apparent Issues

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

Index

Index

Index

ESCAN00070350	Fee does not report FEE_CRITICAL_LEVEL to user-defined Error Callback
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	8.00.06
Fixed in versions:	

Problem Description:

What happens (symptoms):

FEE never invokes user-defined callback with error code FEE_CRITICAL_LEVEL, to warn about reached FSS threshold.

When does this happen:

It should happen, when FEE detects that Foreground Sector Switch Threshold was reached.

In which configuration does this happen:

It should happen, if a user-defined error callback was configured. (Otherwise, nobody would care.)

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00070637	Defines for DEM events not generated. This leads to multiple linker errors.
Component@Subcomponent:	MemService_AsrNvM@GenTool_GeneratorMsr
First affected version:	2.01.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

The linker throws errors like the following:

NvM_DemErrWriteProtected from NvM.o
NvM_DemErrReqFailed from NvM_Act.o
NvM_DemErrIntegrityFailed from NvM_Act.o
NvM_DemErrLossOfRedundancy from NvM_Act.o
NvM_DemErrQueueOverflow from NvM_Queue.o

==> The configuration container "NvmDemEventParameterRefs" is mandatory, if DEM events are used.

When does this happen:

Linking of the project.

In which configuration does this happen:

If DEM events are used
and
the container "NvmDemEventParameterRefs" is not added manually (per default, this container is missing)

Resolution Description:
Workaround:

Configure DEM and NVM for NVM's errors.
Initially, when setting up a new project, add a container instance "NvmDemEventParameterRefs" to NVM (in DaVinci CFG5 use the "Basic Editor".
Then, "create" all container's child parameters, except E_VERIFY_FAILED and E_WRONG_BLOCK_ID (NVM does not support the related features, at all)
(As long as they are existing, CFG5 is able to assist users in a basic way).

Resolution:

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

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.

ESCAN00074769	TechRef gives wrong information about minimum "Internal Buffer Size"
Component@Subcomponent:	If_AsrIfFee@Doc_TechRef
First affected version:	8.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

TechnicalReference_Asr_Fee, chapter 5.1.5.3 states that "Internal buffer size" must "be larger than largest 'Write Alignment' setting."

This is is not correct. "Internal buffer Size" may also be equal to partitions' largest "Write Alignment" setting.
However, it is correct that "Internal Buffer Size" depends on "Write Alignment" settings.

In the worst case a user could configure a larger "Internal buffer value than actually necessary", which does not cause any misbehavior.

Note, that DaVinci CFG's validation performs its checks correctly; and, if validation fails, the error message gives correct information.

When does this happen:
---
n/a

In which configuration does this happen:
---
n/a

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00075523	NVM's requests to FEE are rejected
Component@Subcomponent:	MemService_AsrNvM@Implementation
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

NVM's requests fail (NvM_GetErrorStatus delivers NVM_REQ_NOT_OK) because requests to FEE or EA (via MemIf) are rejected.
Additionally FEE/EA reports a development error to FEE - FEE_EA_BUSY_

When does this happen:

It happens during run-time. Basically it only depends on:
- Concrete implementation of FEE/EA (AUTOSAR _allows_ to reject requests if component is in state MEMIF_BUSY_INTERNAL, but it was left open to the implementation
- Whether and when the component enters the state "MEMIF_BUSY_INTERNAL". For an EA it could be unnecessary to use it at all; a FEE should use it when it is performing a "sector switch", decoupled from serving client's (NVM's) requests. Therefore write requests are necessary in order to fill the NV memory, making "internal operations" necessary at all.

Some implementation might enter state MEMIF_BUSY_INTERNAL immediately during execution if Fee_Init/Ea_Init. In such cases, this issue would be always observable.

In which configuration does this happen:

It may happen with a 3rd party FEE, depending on its implementation of MEMIF_BUSY_INTERNAL handling.
In this case it happens in every configuration.

Issue won't be fixed:
This issue resulted from an Issue in AUTOSAR: Behavior in case of MEMIF_BUSY_INTERNAL was left open to implementation of FEE - there is no standardized way for NVM to get knowledge about the way implemented in FEE.
See RfC #64962
There was a decision to keep only one variant: FEE shall _accept_ requests when it is BUSY_INTERNAL (it is responsible for cancellation / abort / suspend of pending internal operations).
That's the behavior NVM expects.
However, unfortunately they left exceptions - RfC #70170 was created.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00075933	Generator crashes when parsing an OIL file with an include directive with nonexistent absolute path
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	

Problem Description:

What happens (symptoms):

The generate crashes with an exception.

When does this happen:

---

during parsing the file.

In which configuration does this happen:

ALL

(identical to ESOS00004007)

Resolution Description:

Workaround:

If this happens (rare case), please open the OIL file with an editor and correct the path to the correct include path. The include file should typically be found in the generator path.

Resolution:

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

ESCAN00076481	Debugger refuses to read ORTI file
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	6.06.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The debugger is unable to provide information about the currently active task, its priority, the currently running Cat2 ISR, the currently performed OS API service and the last occurred errors (we currently do not know whether the issue is detected at parsing the ORTI file or later).

When does this happen:

This happens always and immediately at debug time when the customer tries to trace the currently active task or ISR.

In which configuration does this happen:

This happens only in configurations with ORTI enabled and depends on no further configuration details. This may happen only with a debugger from a certain vendor.

(this issue is identical to ESOS00004101)

Resolution Description:

Workaround:

On single core systems perform the following substitution manually in the ORTI file:
Replace:
"osConfigBlock.CcbAddress->"
By:
"osCtrlVarsCore0."
On multicore systems, there is currently no workaround.

Resolution:

The described behavior is to be corrected by modification of the code.

ESCAN00079086	Validation causes "Divide by Zero" exception being thrown
Component@Subcomponent:	If_AsrIfFee@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	3.01.01

Problem Description:
What happens (symptoms):

Validation causes "Divide by Zero" exception being thrown.
However, this exception is silently caught by DaViciCFG5; it does not result in user-visible messages.

Limitations or misbehavior resulting from this exception cannot be observed.

When does this happen:

During configuration or during generation (command line, called by DaVinci CFG4) using DaVinci CFG5

In which configuration does this happen:

AddressAlignment or WriteAlignment is set to 0.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00079209	Counter macros <UNIT>*2TICKS and TICKS2*<UNIT> and TimerConstants use floating point operations with incorrect rounding
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	8.00.00
Fixed in versions:	9.01.06

Problem Description:

What happens (symptoms):

Two symptoms:
(1) A floating point library needs to be linked additionally (code size increase).
(2) The result value is always rounded downwards, no mathematical rounding.

When does this happen:

---

(1) In addition to the configurational preconditions, this happens only if the macros are used with variables as input values.
(2) See the description of configuration below.

In which configuration does this happen:

---

(1) This happens only if no hardware floating point support is used
(2) This happens only if the macro/TimerConstants computations lead to non-integer values.

Resolution Description:

Workaround:

Consider that rounding is not done mathematical.

Resolution:

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

ESCAN00079317	Generator does not check whether at least one APPMODE is assigned to each container TASK/ AUTOSTART=TRUE
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	

Problem Description:

What happens (symptoms):

A task, which is configured with AUTOSTART=TRUE, is not started. If this task is used as initial task (single entry point), the whole application will not start.

When does this happen:

This happens always and immediately when the OS is started.

In which configuration does this happen:

A task is configured with AUTOSTART=TRUE and no APPMODE is assigned.

Resolution Description:

Workaround:

The issue is simply that a wrong configuration is stays undetected. to work around this issue, simply correct the configuration.

Resolution:

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

ESCAN00079399	Linker error: '<Cdd>_Transmit' : undeclared identifier
Component@Subcomponent:	Cdd_AsrCddCfg5@Description
First affected version:	2.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

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

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

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.

ESCAN00079977	TechnicalReference does not help to solve installation problem
Component@Subcomponent:	Os_CoreGen6@Doc_TechRef
First affected version:	6.16.01
Fixed in versions:	

Problem Description:
What happens (symptoms):

The OS Generator refuses to generate files with the following message:
[Fatal Error: 0021] XML write error: Failed to CreateInstance on an XML DOM

Neither the message itself, nor the TechnicalReference help to solve the issue.

When does this happen:

This happens during OS generation.

In which configuration does this happen:

This issue is configuration independent.

Resolution Description:
Workaround:

Install MSXML 4.0 Service Pack 3.

Resolution:

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

ESCAN00080172	Compiler error: missing API SyncScheduleTable, StartScheduleTableSynchron and/or SetScheduleTableAsync
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	8.00.01
Fixed in versions:	

Problem Description:
What happens (symptoms):

The compiler/linker issues an error that one or more of the following functions are implicitly defined, do not have a prototype, or cannot be linked:
SyncScheduleTable
StartScheduleTableSynchron
SetScheduleTableAsync

When does this happen:

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

In which configuration does this happen:

This happens if in an OS SC1 or SC3 at least one schedule table is configured with OsScheduleTableSyncStrategy other than NONE.

Resolution Description:
Workaround:

No workaround necessary. The issue occurs only in cases where a non-supported or non-licensed feature is enabled in the configuration.

Resolution:

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

ESCAN00080185	Erroneous Os_rules.mak
Component@Subcomponent:	Os_Rh850Fs04@Make
First affected version:	1.00.00
Fixed in versions:	

Problem Description:
Os_rules.mak: osSysCallTable.c should be osSysCall.c

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00080528	IllegalStateException with Multiple FeatureDeactivationConditions
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	6.00.01

Problem Description:
What happens (symptoms):

The ComStackLib throws the following IllegalStateException
Exception in <MSN> generator during Generation encountered:
java.lang.IllegalStateException: The size of the <ElementInStructType>: <ElementName>(0) does not match to the size of the ConstStruct: <StructName> <NumberOfElementsInStructName>!

When does this happen:

Always and immediately at generation time if the configuration matches.

In which configuration does this happen:

Any configuration with a ConstStruct
AND
the structure element has multiple FeatureDeactivationConditions
AND
the last FeatureDeactivationCondition is not true (Note, FeatureDeactivationConditions are String ordered by the FeatureDeactivationCondition reason)

not affected modules:

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00080543	Configuration Errors in SectorSwitch ThresholdReserves, if in FBL mode
Component@Subcomponent:	If_AsrIfFee@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

Configuration (Validation) Errors are issued:
1. AR-ECUC02008 Invalid multiplicity (1 message)
AR-ECUC02008 Mandatory parameter FeeFssThresholdReserved is missing in FeePartitionConfiguration.
This error may also be issued for FeeBssThresholdReserved
2.Cfg00022 Missing parameter value (1 message)
Cfg00022 The value of parameter FeeFssThresholdReserved is missing or empty.
This error may also be reported for FeeBssThresholdReserved
3. FEE00049 The difference between both Sector Switch Thresholds should be greater than the largest configured block.
This error occurs most likely when adding new or resizing existing blocks.
4. FEE50004 BSS threshold reserve too big
5. FEE50005 FSS threshold reserve too big

This happens though both sector switch reserve values are not used in FBL config;.

When does this happen:

It happens when configuring BSW, when setting up a new project or when significantly changing an existing configuration (Partition parameters, add new blocks, heavily resize existing blocks).

In which configuration does this happen:

It happens, if FEE was configured with "Flash Bootloader Configuration" (FeeSpecificFeatures/FeeFblConfig) was set to TRUE.
The errors being issued would be correct, if "Flash Bootloader Configuration" was FALSE, i.e. the issue is that wrong values are not ignored (or at least: that errors' severities are not reduced).

When using comfort view ASR3 projects (i.e. with DaVinci CFG4) these parameters are not shown.

In ASR 4 projects all of these parameters should be configured using the Basic Editor, anyway. Additionally these validation errors are complemented with solving actions. Thus it should be less annoying.

Resolution Description:
Workaround:

Use "Basic Editor" to set Partitions' Foreground and Background Sector Switch reserve values. Just choose valid value - so that error messages disappear.
This way should be preferred over setting those parameters to "User Defined", turning validation errors to warnings.
The latter option may also affect other values/validations.

Resolution:

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

ESCAN00080663	"Insert Block" causes Access Violation being reported
Component@Subcomponent:	If_AsrIfFee@Description
First affected version:	8.02.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

DaVinci CFG4 displays an error dialog "Access violation at address 0x... in Module DaVinciCFG.exe ..."

When does this happen:

It happens during configuration of FEE, when attempting to insert a new block.
However, it only occurs, if FEE configuration view was just opened, and no existing FEE block was selected before clicking button "Insert Block".

In which configuration does this happen:

It happens, if MICROSAR NVM is also activated in configuration

Resolution Description:

Workaround:

Select an existing FEE block before clicking the "insert" button.

Resolution:

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

ESCAN00080731	Fee_ForceSectorSwitch might do nothing
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Fee_ForceSectorSwitch processing might do nothing, though it should.

Actually there are situations Fee_ForceSectorSwitcht actually does nothing, especially when the desired situation was already achieved: All most recent data instances are located within one logical sector; the other one is empty.
Additionally Fee_ForceSectorSwitch skips parti

However, under circumstances described below, FeeForceSectorSwitch unintentionally does nothing.

When does this happen:

It happens during run-time, in ECU; but it only happens, if the newer sector (of a partition) already contains data, which did not have been accessed, so far.
I.e. those data belong to Blocks which did not have been read during start-up (e.h. NvM_ReadAll).

In this case, though both logical sectors are in use, FEE does not do anything.

This might result in performance penalties, especially after SW (configuration) updates, because FEE remains unable do "fix" some issues (defective chunk-links, or too small link-tables).

Since very most data stored in NV memory are intended to be accessed, i.e. to be read, it is unlikely that this issue will actually be noticed at customer side.

In which configuration does this happen:

It happens if Fee_ForceSectorSwitch API was enabled.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00080742	OSTICKDURATION not defined
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compilation error is reported as there has no OSTICKDURATION been defined but used,

When does this happen:

The error is issued by the compiler during compilation of the code if the code uses OSTICKDURATION.

In which configuration does this happen:

This happens independent from the configuration.

Resolution Description:

Workaround:

Two alternatives:
1. Use the new macros OSSECONDSPERTICK_<CounterName> instead. These define the time of a counter tick as a floating point value in seconds while OSTCKDURATION defined it as an integer value in nanoseconds.
2. Create define macro OSTICKDURATION manually as an integer value which is equal to the return value of OS_TICKS2NS_<CounterName>(1). Mind that this value depends on the configuration parameter OsCounter/OsSecondsPerTick and needs adaptation after each change of that parameter. The macro OS_TICKS2NS_<CounterName> may be found in the generated file tcbpost.h. Direct usage of OS_TICKS2NS_<CounterName> for the definition of OSTICKDURATION could cause compilation problems of the RTE if the RTE uses OSTICKDURATION in a preprocessor condition where floating point operations are not allowed.

Resolution:

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

ESCAN00081556	NvM block's Datasets parameter is marked as erroneous but cannot be edited
Component@Subcomponent:	MemService_AsrNvM@GenTool_GeneratorMsr
First affected version:	3.03.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

DaVinci Configurator issues errors:
NVM01015 With management type "redundant" the count of blocks must be 2.
NVM01016 With management type "native" the count of blocks must be 1.

Parameter "Datasets" (NvMBlockDescriptor/NvMNvBlockNum) gets marked being erroneous, but it cannot be edited.
However, solving actions given along with error messages work as intended.

When does this happen:

It happens during configuration.

In which configuration does this happen:

It happens for NVRAM Blocks with a "Management Type" (NvMBlockDescriptor/ NvMBlockManagementType) with the value NVM_BLOCK_NATIVE or NVM_BLOCK_REDUNDANT, while the number of datasets ("Datasets", s.a.), but "Datasets" (s.a.) does not match the chosen type.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00081721	Extended version check is not available for MSR4
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	8.00.00
Fixed in versions:	8.01.00

Problem Description:

What happens (symptoms):

This code is not MSR4-conform and has to be removed:

```
# if !defined(V_SUPPRESS_EXTENDED_VERSION_CHECK)
# include "v_ver.h"
# endif
```

When does this happen:

Leads to compile error, because either V_SUPPRESS_EXTENDED_VERSION_CHECK nor v_ver.h exist within MSR4.

In which configuration does this happen:

---

all

Resolution Description:

Workaround:

Just globally #define V_SUPPRESS_EXTENDED_VERSION_CHECK
Either via compiler's command line (usually '-D') or in a central include file (such as Compiler_Cfg.h)

Alternatively create an empty file "v_ver.h" (in a directory that is contained in compiler's include path)

Resolution:

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

ESCAN00082117	Only on RH850 F1L: Stack overflow detection calls the ErrorHook instead of ProtectionHook
Component@Subcomponent:	Os_Rh850Fs04@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

On detection of stack overflow, the ErrorHook is called instead of ProtectionHook.
After ErrorHook the OS correctly calls ShutdownHook and requests the system shutdown.
The error code of ErrorHook and ShutdownHook is E_OS_STACKFAULT.

When does this happen:

During runtime when stack overflow is detected.

In which configuration does this happen:

This happens only on RH850 F1L in scalability classes SC3 and SC4.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00082482	Compiler error: Double definition of osTaskStack...
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	

Problem Description:

What happens (symptoms):

The generator produces to arrays with identical name. The array names both start on osTaskStack. The compiler states an error message as a result of this.

When does this happen:

The error is issued by the compiler during compilation of the operating system.

In which configuration does this happen:

This may happen only if two applications exist with one of them having the same name as the other one but extended by a number.

The reason is that task stack names are generated according to the following pattern:
osTaskStack<ApplicationName><Counter starting with 0>
Now assume one application was named "Appl" and another one "Appl1".
The first task stack of the application "Appl1" would get the name "osTaskStackAppl10".
The stacks of the application "Appl" would be called: "osTaskStackAppl0", "osTaskStackAppl1", "osTaskStackAppl2", ... , "osTaskStackAppl9", "osTaskStackAppl10"!
So the eleventh stack of the application "Appl" would have identical name as the first stack of the application "Appl1", which leads to the described issue.

Resolution Description:

Workaround:

Prevent application names which end on a number.

Mind that this workaround is stricter than necessary. Further suitable workarounds may be found by analysis of the configurations in which this issue occurs.

Resolution:

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

ESCAN00082600	The file <Mip>_Cbk.h is not generated for a CddSoAdUpperLayerContribution with only CddSoAdUpperLayerTxPdus
Component@Subcomponent:	Cdd_AsrCddCfg5@GenTool_GeneratorMsr
First affected version:	3.02.00
Fixed in versions:	5.00.00

Problem Description:

What happens (symptoms):

The file <Mip> Cbk.h is not generated.

When does this happen:

Always and immediately

In which configuration does this happen:

Any CddSoAdUpperLayerContribution where only CddSoAdUpperLayerTxPdus are configured.

Resolution Description:

Workaround:

Configure a CddSoAdUpperLayerRxPdu dummy PDU and implement the required APIs as empty Stubs.

Resolution:

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

ESCAN00083807	Compiler error: undefined preprocessing identifier DEM_FEATURE_NEED_USER_INDICATOR_CONTINUO resp. DEM_FEATURE_NEED_USER_INDICATOR_BLINKING
Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

Compiler error (here TexasInstruments compiler v5.1.1):
#195-D zero used for undefined preprocessing identifier
"DEM_FEATURE_NEED_USER_INDICATOR_BLINKING"

When does this happen:

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

In which configuration does this happen:

The configuration defines some (non-special) indicators which all are not used (not referenced) by any event:

The configuration contains some indicators (containers /Dem/DemGeneral/DemIndicator)
AND
Some of them are not used by any event as DemIndicatorRef

The "special" indicators do not trigger this issue: indicators referenced by either /Dem/DemGeneral/DemMILIndicatorRef, /Dem/DemGeneral/DemGeneralJ1939/DemAmberWarningLampIndicatorRef,
/Dem/DemGeneral/DemGeneralJ1939/DemProtectLampIndicatorRef, /Dem/DemGeneral/DemGeneralJ1939/DemRedStopLampIndicatorRef

Resolution Description:
Workaround:

Delete all unused /Dem/DemGeneral/DemIndicator containers.

Resolution:

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

ESCAN00084103	Compiler/Linker error: an array is empty in a Predefined Variant with Precompile Selectable
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	7.00.00

Problem Description:
What happens (symptoms):

The compiler/linker reports the error messages and the project cannot be builded.

In the case of a struct based array:
The typedef of the structure used in the typedef of the initialization structure typedef is not generated. Due to the missing typedef the compiler reports the warning "missing type specifier - int assumed." and consecutive the compiler error "syntax error : missing ';' before '*'"

In the case of a simple value based array:
The instance of the initialization structure references a not generated symbol. Due to this the compiler reports "'<SymbolName>' : undeclared identifier.

When does this happen:

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

In which configuration does this happen:

The module is in the configuration variant precompile selectable with predefined variants
AND
the optimization isOptimizeConstArrays2Define is activated.
AND
all values of the array are identical in all predefined variants.
AND
a array is empty in minimum one predefined variant.

Resolution Description:
Workaround:

If the module has a parameter like <MSN>OptimizeConstArrays2Define, set the parameter to false.
If the module has not the parameter and the switch is enabled always by the implementation, no workaround is available.

Resolution:

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

ESCAN00084471	BETA version - the BSW module has a feature with BETA state (FEAT-1093)
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.08.00
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-1093: Support of different strategies for writing NV data in Nv Block SWCs

To ensure that only productive code is used verify that:
- no cyclic or on data reception triggered runnables in a Nv Block SWC are used

Resolution Description:

ESCAN00084633	Error message "FEE90500" during generation of FEE
Component@Subcomponent:	If_AsrIfFee@GenTool_GeneratorMsr
First affected version:	3.01.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

The following errors occurs during generation:
FEE90500 The value 9223372036854775807 with comment (physical sector start address) is not in the range of the specified datatype UINT_32.

When does this happen:

During generation of BSW.

In which configuration does this happen:

If the related DrvFls does not have a FlsSectorList, which can only happen, if the underlying Fls defined FlsSectorList or its child, FlsSector, to be optional (which does not complain with AUTOSAR).

Currently it may happen only in conjunction with DrvFls_VttCanoe01Asr.

This issue will not be fixed in FEE, because it is caused by a deviation from AUTOSAR in FLS implementation.
Additionally it can (even: should be fixed) by a suitable Pre-/Recommended configuration of Fls driver.
The error message issued during generation also causes a compile time error (an #error directive it is generated into Fee_Lcfg.c).
This is considered sufficient for rare cases, an Fls implementation deviated from AUTOSAR.

Resolution Description:
Workaround:

FEE requires FlsSectorList and descendant FlsSector entries.
Manually create a FlsSectorList, and at least one FlsSector entry, to describe the properties of the Flash area to be used by FEE.

Note:
It depends on Fls's implementation, which entries are necessary, and which may be omitted.
Some implementations allow to omit unused sectors (especially Program Flash area(s)), but some require configuring ALL sectors.

Resolution:

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

ESCAN00084659	Compiler error due to missing include of NvM.h
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.08.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

Compilation fails because of undeclared identifiers since the NvMBlockDescriptor handles are used but not defined.

When does this happen:

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

In which configuration does this happen:

This happens if C/S Interfaces for Nv Block SWCs are used to call NvM APIs and no dirty flags are used. Furthermore all Server Runnables have to be realized as functions, e.g. because of asynchronous calls.

Resolution Description:

Workaround:

Use dirty flags or at least one synchronous call for a C/S operation of a Nv Block SWC.

Resolution:

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

ESCAN00084792	Compiler error: Wrong initializer for NvBlockSWCs in systems with OsApplications
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.01.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

Compile error because the NvBlockDescriptor buffer initialization is syntactical not correct.
e.g. Rte_<SWCName>_<NvBlockDescriptorName> = ;

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 NvBlockSWCs in systems with OsApplications are used and the configuration contains NvSignalPorts with partial NvDataMappings (only sub elements, like record elements, are mapped).

Resolution Description:

Workaround:

Create a new Nv PortInterface with the data type used for the effected NvBlockDescriptor.
Add a P-NvSignalPort with this PortInterface to the NvSWC.
Go to NvBlockDescriptor Tab and add an additional ReadAccess to the newly created element.
Add a R-NvSignalPort with same PortInterface to an other SWC and connect both ports.
Configure a ReadAccess for this element, too.

Resolution:

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

ESCAN00084816	Compiler error: struct has no member SnapshotData
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	6.02.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Function 'Dem_Data_PrestorageApplySRec' will not compile due to an undefined struct member 'SnapshotData'

When does this happen:

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

In which configuration does this happen:

Configuration using prestored FreezeFrames (DemMaxNumberPrestoredFF > 0)
AND
No SnapshotRecords (DemFreezeFrameClass) are configured

This would be possible e.g. for OBD configurations to prestore the OBD FreezeFrame data, while no UDS snapshot records are supported by the ECU.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00084822	Compiler error: struct has no member MaxDebounceValue, missing symbols DEM_ESM_TEST/SET_FDC_MAX
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	4.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Function 'Dem_Data_EntryInit' will not compile due to an undefined struct member 'MaxDebounceValue'

When time based debouncing is enabled:
Compiler warning for undefined symbol 'DEM_ESM_SET_FDC_MAX' in function Dem_Esm_TimeBasedCalculate.
Compiler warning for undefined symbol 'DEM_ESM_TEST_FDC_MAX' in function Dem_Esm_TimeBasedCommit.

When does this happen:

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

In which configuration does this happen:

Configuration using internal data element MAX_FDC_DURING_CURRENT_CYCLE
AND NOT also using internal data element MAX_FDC_SINCE_LAST_CLEAR

Resolution Description:

Workaround:

No workaround available

Resolution:

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

ESCAN00084863	API names of 3rd-party Wdg drivers may not be recognized correctly
Component@Subcomponent:	If_AsrIfWdTttechSub@Generator
First affected version:	3.03.09
Fixed in versions:	

Problem Description:
What happens (symptoms):

WdgIf generator issues the following warning during generation from DaVinci Configurator:

WDGIF01151 Could not determine platform for watchdog `Wdg_Impl` after searching all BSWMD files in `<SIP>\BSWMD` and all its subdirectories.
/ActiveEcuC/WdgIf

The WdgIf_Lcfg.c file contains default function names in the static constant structure, instead of the actual Wdg__SetMode and Wdg__SetTriggerCondition functions of the driver:

```
static const WdgIf_InterfaceFunctionsType generic_driver_functions =
 { Wdg_SetMode /* Wdg_SetMode */
 , Wdg_SetTriggerCondition /* Wdg_SetTriggerWindow */
 };
```

As a consequence, the code cannot be compiled correctly.

When does this happen:

This happens if the naming schema of the 3rd-party Wdg public functions does not follow the schema Wdg_<vendorId>_<HW_Name>_<serviceName>

Resolution Description:
Workaround:

Insert a #define inside of Compiler_Cfg.h to provide renamings of the actual Wdg functions, or manually adapt WdgIf_Lcfg.h after each generation.

Resolution:

ESCAN00084954	Compiler error: undefined symbol 'Dem_Esm_TestEventHealed'
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	9.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

Compiler warning for undefined symbol 'Dem_Esm_TestEventHealed' in function Dem_Esm_TestDtcHealed.
It will result in a linker error due to undefined function 'Dem_Esm_TestEventHealed' if this warning is ignored.

When does this happen:

When compiling the module in a configuration as mentioned below

In which configuration does this happen:

Aging after healing is set to 'age after DTCs have healed' (DEM_AGING_AFTER_HEALING)
AND
No Indicators are configured (so no DTCs need healing)
AND
Event Combination is enabled

Resolution Description:
Workaround:

Since no indicators are configured, the setting of DEM_AGING_AFTER_HEALING will not have an effect.
Set 'DemAgingAfterHealing' to DEM_AGING_WHILE_HEALING to circumvent the issue.

Resolution:

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

ESCAN00084967	Null pointer exception when complex data mapping exists for a primitive signal
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.08.00
Fixed in versions:	4.09.00

Problem Description:

What happens (symptoms):

A null pointer exception is reported.

When does this happen:

During generation

In which configuration does this happen:

This happens when a configuration contains a complex data element mapping for a primitive system signal.

Resolution Description:

Workaround:

Do not configure SenderRecRecordTypeMapping for primitive signals.

Resolution:

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

ESCAN00084971	Compile Error: Dem Namespace Separation is Broken
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	6.00.00
Fixed in versions:	8.00.00

Problem Description:
What happens (symptoms):

Several Compile Errors occur, because symbols do not have the correct separated namespace.
The Dem feature with separated namespaces is broken.

When does this happen:

At compile time.

In which configuration does this happen:

Any configuration with Diag_Asr4Dem@GenTool_GeneratorMsr .
Please note: There exists no Diag_Asr4Dem build with the affceted version range.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00084997	Access on the first configuration structure of PB-S is generated wrong
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	4.01.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

At compilation time the compiler throws the following error:

"bsw\sip\static\EcuM\EcuM.c", line 2112: error #20: identifier "EcuM_GlobalConfigRoot" is undefined
 ECUM_GLOBAL_CFG_PTR = ECUM_FIRST_CONFIG_STRUCTURE;

This is caused by an inconsistent generation of the EcuM, the access on the first configuration structure of PB-S is done by an array but the init structure itself is a struct and no array.

When does this happen:

During compilation of the module EcuM.

In which configuration does this happen:

Only in PB-S configurations with MCAL modules which do only support PB-L and the Cfg5 license does not support PB-L.

Resolution Description:
Workaround:

There are two possible workarounds:

1. Configure a user config file in the EcuM which redefines the ECUM_FIRST_CONFIG_STRUCTURE:

#undef ECUM_FIRST_CONFIG_STRUCTURE
#define ECUM_FIRST_CONFIG_STRUCTURE &EcuM_GlobalPcConfig[0]

2. Use another Cfg5 license which supports PB-S and PB-L

Resolution:

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

ESCAN00085138	DTCs not stored on FaultDetectionCounter Threshold
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	8.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

A DTC is not stored, although it should be

When does this happen:

When debouncing an event reaches the storage FaultDetectionCounter Threshold, the DTC is not stored as it should be.

In which configuration does this happen:

Configurations using only time-based de-bouncing

Resolution Description:
Workaround:

Enable Counter-Based de-bouncing for at least one event.
If necessary, you can create an event without DTC for this. Such an event will not change the observable behavior of the ECU.

Resolution:

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

ESCAN00085160	Compiler error: undefined symbol 'INewEventStatus'
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	9.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Function Dem_Esm_EventProcessHealing will not compile due to undefined symbol 'INewEventStatus'

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:

Option 'Aging After Healing' is enabled for all DTC (DEM_CFG_AGING_AFTER_HEALING_ALL_DTC == STD_ON)
AND
No Indicators are configured
AND
The healing target (DemIndicatorHealingCycleCounterThreshold) is >= 1 for at least one event

Resolution Description:

Workaround:

Configure at least one indicator.
If you attach it only to an event without DTC, the indicator will not be set and no bit7 will be visible on the bus.

Resolution:

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

ESCAN00085281	Com group signal settings are not synchronized in case of signal group degradation
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:
What happens (symptoms):

COM EcuC synchronization does not work for com group signals in case of signal group degradation.

When does this happen:

During validation/generation of RTE in case the configuration is as described below.

In which configuration does this happen:

This happens if
- signal groups are received over COM
- signal group degradation is used: signal group is received by multiple port prototypes with different interfaces (not all group signals are received by each port)

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00085371	Compiler error: Duplicated Rte_COMCbk in case of invariant signals in variant projects
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.05.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

Compilation fails, because the RTE contains a duplicated Rte_COMCbkRxTOut or COMCbkInv function.

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 rx signal that is not variant and that uses an alive timeout or invalidation.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00085447	generator exits unsuccessfully with no error message
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	

Problem Description:
What happens (symptoms):

The generator exits unsuccessfully with no error message. So Windows complains about the generator to not work anymore.

When does this happen:

This happens when start- and/or end-address of a peripheral region have been configured but no identifier yet. This may even happen when the user has not yet finished the configuration as the generator is internally used by the configurator to check the configuration cyclically. The problem disappears, once the identifier of the peripheral region has been set.

In which configuration does this happen:

This happens only in configurations with peripheral regions.

Resolution Description:
Workaround:

Configure the identifier of the peripheral region

Resolution:

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

ESCAN00085523	NullPointerException occurs at generation time and no data is generated
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	5.00.00
Fixed in versions:	6.00.02

Problem Description:

What happens (symptoms):

A NullPointerException occurs at generation time and no data is generated.

When does this happen:

At generation time.

In which configuration does this happen:

A non structured VAR Array has been instanciated with the API getVarArray(final String tagName, final int size).

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00085586	Rte_Read returns incorrect status for connected ports without read access
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.06.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

Rte_Read returns RTE_E_OK instead of RTE_E_NEVER_RECEIVED or RTE_E_INVALID if connection is optimized away.

When does this happen:

This happens before a value was received at all.

In which configuration does this happen:

- Never received handling is configured for element of receiver port.
- There is no write access configured for this element on sender side.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00085595	Compiler error: Wrong declarations in case of N:1 S/R communication with invalidation enabled and multiple OsApplications
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

Compilation fails, because
- IocWrite_Rte_RxInvalidate_* is called but not declared
- a struct in Rte_Type.h (Rte_*_RxInvalidateFlagsType) is generated without any members

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 case of N:1 sender/receiver communication with invalidation (Handle Invalid set to Keep) enabled and
multiple OsApplications. There must be at least two senders belonging to different OsApplications. If all
senders belong to the same OsApplication, there is no issue.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00085663	Unexpected Windows Error Message in case of wrongly configured Linked Resource
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	

Problem Description:

What happens (symptoms):

Windows states that the MICROSAR OS generator (gen<PlatformName>.exe) does not work anymore.

When does this happen:

This happens always and immediately at generation time when the configuration is erroneous as described below.

In which configuration does this happen:

The configuration contains a linked resource which has no link to another resource. This is an invalid configuration for linked resources.

Resolution Description:

Workaround:

Add a link to a valid resource to all resources with type 'LINKED'

Resolution:

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

ESCAN00085688	Rte_SwitchAck returns RTE_E_NO_DATA instead of RTE_E_TIMEOUT on timeout
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

If a timeout happens while waiting on the mode switch inside Rte_SwitchAck function, the return value is RTE_E_NO_DATA instead of RTE_E_TIMEOUT.

When does this happen:

Timeout on mode switch.

In which configuration does this happen:

- Mode switch acknowledgment point with wait point and timeout handling

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00085694	Rte_IStatus returns incorrect status
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00, 1.08.01

Problem Description:

What happens (symptoms):

Rte_IStatus returns RTE_E_UNCONNECTED instead of RTE_E_OK although it is connected

When does this happen:

Always during runtime.

In which configuration does this happen:

This happens when implicit communication is used and no RxTimeout or InvalidValue (Keep) or NeverReceived flag is configured.

Resolution Description:

Workaround:

No workaround available.

Please note that the issue only occurs if neither RxTimeout, InvalidValue (Keep) nor NeverReceived is configured.
In that cases the usage of Rte_IStatus makes less sense because it would return always the same return code.

Resolution:

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

ESCAN00085725	Rte VFB Trace hooks with client prefix(es) are not generated
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.03.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The hook function with client prefix does not appear in generated code.

When does this happen:

In configurations described below.

In which configuration does this happen:

- only the prefixed hook function is used, not the hook function itself and
- the corresponding Rte function can be implemented as a macro

Resolution Description:

Workaround:

Use the RTE API hook function without prefix in addition to a prefixed one.

Resolution:

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

ESCAN00085853	Compiler error: variable ret (Std_ReturnType of Rte_Feedback) is used before set
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00

Problem Description:

What happens (symptoms):

In Rte_Feedback the return value 'ret' is used before set.

When does this happen:

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

In which configuration does this happen:

This happens for external S/R communication with 'wait for Tx-Ack' and timeout > 0
when RteOptimizationMode is RUNTIME

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00085901	Rte_SwitchAck uses wrong / old VFB trace hook names starting with Rte_FeedbackHook
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00

Problem Description:

What happens (symptoms):

Rte_SwitchAck calls Rte_FeedbackHook_*_Start and Rte_FeedbackHook_*_Return instead of Rte_SwitchAckHook_*_Start and Rte_SwitchAckHook_*_Return

When does this happen:

In the configurations described below.

In which configuration does this happen:

Whenever Rte_SwitchAck api is generated.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

2.4 Compiler Warnings

As a service we also provide the known compiler warnings. The occurrence of a compiler warning may depend on the used basic software configuration and compiler settings.

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.

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.

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.

ESCAN00074793	Compiler warning: Condition is always constant
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	4.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warning 'Condition is always constant'

When does this happen:

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

In which configuration does this happen:

Configurations without DTCs
AND
Precompile configuration

Resolution Description:

Workaround:

The warning can be ignored

Resolution:

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

ESCAN00081825	Compiler warning: non-portable OS code compiled with --asm_warnings
Component@Subcomponent:	Os_PlatformRh850Gen6@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

When compiling the OS files, the compiler issues several warnings when compiling with the option --asm_warnings.
The warning is: "warning #1546-D: asm constructs are non-portable"

Those warnings can safely be ignored. The OS has non portable code inside and in case of inline assembly the compiler issues warnings if used with --asm_warnings

When does this happen:

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

In which configuration does this happen:
---
the warnings will be issued when compiling the OS with the compiler option --asm_warnings.

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed as the OS is non-portable.

Resolution Description:
Workaround:

It is possible to compile the OS without this option and to compile all other sources / modules with this option.
If OS is compiled with this option, then it is safe to ignore the issued warnings.

ESCAN00083396	Compiler warning: narrowing or signed-to-unsigned type conversion found
Component@Subcomponent:	SysService_AsrCrc@Implementation
First affected version:	3.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Warnings during compilation of code.
Compiler warning: narrowing or signed-to-unsigned type conversion found

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:

OBD2 usecase with MSR3, compiler DiabData 5.9.2.0

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00083410	Compiler warning: narrowing or signed-to-unsigned type conversion found
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	8.00.00
Fixed in versions:	8.01.00

Problem Description:
What happens (symptoms):

Warnings during compilation of code:
Compiler warning: narrowing or signed-to-unsigned type conversion found

"../../../external/BSW/Fee/Fee.c", line 4424: warning (dcc:1643): narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned short const
"../../../external/BSW/Fee/Fee.c", line 4807: warning (dcc:1643): narrowing or signed-to-unsigned type conversion found: int to unsigned char
"../../../external/BSW/Fee/Fee.c", line 5722: warning (dcc:1643): narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned char const
"../../../external/BSW/Fee/Fee.c", line 6197: warning (dcc:1643): narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned short const
"../../../external/BSW/Fee/Fee.c", line 6611: warning (dcc:1643): narrowing or signed-to-unsigned type conversion found: int to unsigned char
"../../../external/BSW/Fee/Fee.c", line 7607: warning (dcc:1643): narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned char const
"../../../external/BSW/Fee/Fee.c", line 7626: warning (dcc:1643): narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned char const
"../../../external/BSW/Fee/Fee.c", line 7998: warning (dcc:1643): narrowing or signed-to-unsigned type conversion found: int to unsigned char

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:

OBD2 usecase with MSR3
MCAL used, NvM Blocks used with EEP

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00083804	Compiler warning: conditional expression or part of it is always true/false
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	8.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

Warnings during compilation of code:
Compiler warning: conditional expression or part of it is always true/false"
../../../external/BSW/Fee/Fee_Partition.c", line 189: 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:

OBD2 usecase with MSR3
MCAL used, NvM Blocks used with EEP

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed, since the warning does not appear correct.
Condition is neither always true, nor always false; code coverage report proves it.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00085294	osAmITrusted() used, even if SysCallIsFastAgainstDecisionAndCall=TRUE
Component@Subcomponent:	Os_CoreGen6@Implementation
First affected version:	9.01.00
Fixed in versions:	9.01.06

Problem Description:

What was done:

Embedded code change.

Why was it done:

- The if case is never reached on some platforms.

What is the impact on the application:

- Maybe compiler warning.

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.

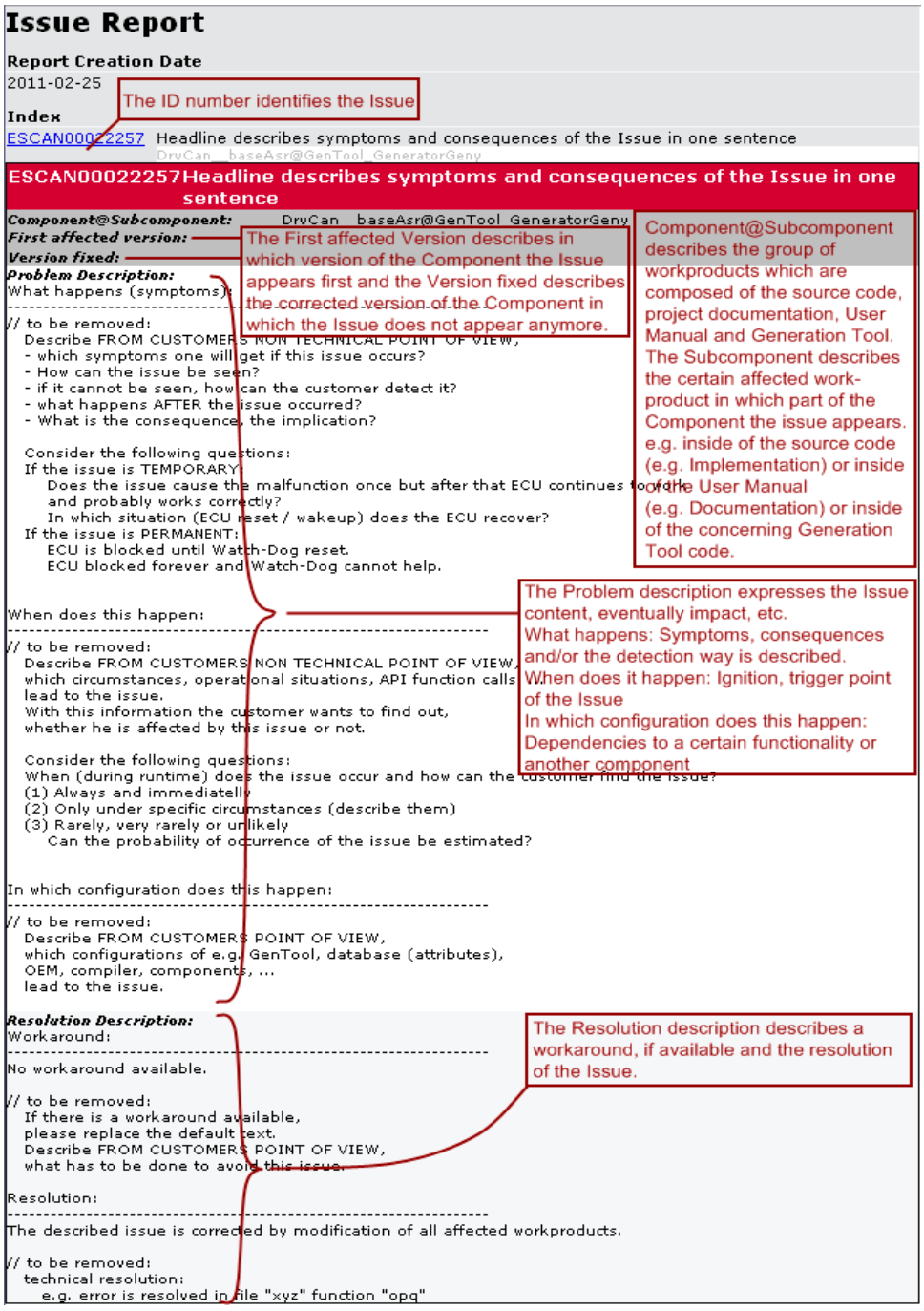
3. New Issues for Information

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



5. Quality Management Contact

Diemo Krüger
PES Quality Management Engineer
Productline Embedded Software (PES)

Vector Informatik GmbH
Ingersheimer Str. 24
D-70499 Stuttgart

Phone: +49 711 80670-3477
Fax: +49 711 80670-399
eMail: diemo.krueger@vector.com