Issue Report

License Number	Customer
CBD1400351	Nexteer Automotive Corporation Package: GM Global A / Flexray - ECU Product "Steering Systems"

Maintenance Expiry Date

2015-09-01

SIP Delivery Date	SIP Version
2016-09-16	16.00.00

SLP	Delivery Number
GM Global A / Flexray	D04

Report Creation Date

2016-09-16

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.
- **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.
- **Not Released Functionality:** Not released functionalities are modules and features that have not yet passed a complete development cycle (they are e.g. not or only partly tested). For serial production projects the integrator has to ensure that all BETA features are disabled as indicated. If a ESCAN affects a complete BSW module, the module must not be used for serial production.
- **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.

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’).

ESCAN00087476	Possible data loss due to aborted Flash write access
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	6.00.00
Fixed in versions:	8.00.82

Problem Description:

ESCAN00087476 Possible data loss due to aborted Flash write access

What happens (symptoms):

Data cannot be read; Fee_Read does not deliver any data.

In case of "critical data" (e.g. FBL information, ECU coding parameters) an ECU might be unable to start.

When does this happen:

It happens at power-up, either if FBL is using FEE to determine whether to start FBL or Application, or during Application start-up, typically during/after NvM_ReadAll.

The root cause occurred earlier: Generally, if a FLASH programming operation was aborted, e.g. due to an unintended reset (watchdog, power-loss, SW induced reset) or due to a cancellation request (Fls_Cancel), the cells of a FLASH page might not be fully programmed.
Subsequent reads on this page may deliver the written data correctly, however after some time passed (presumable environmental changes also have an effect) the same FLASH page may be read out as empty. This behaviour is caused by an insufficient charging level of the FLASH cells.

This may cause the FEE to detect most recent data as "completely written" initially, but later read attempts result in "incompletely written".
E.g. during next read FEE may detect a data block as being valid - it can be retrieved; a following read attempt may result in "not found" (reverting to previous instance, if available).

If the aborted write was part of a sector switch, i.e. copying data, FEE might detect a data block as already copied.
Finally, after all blocks have been processed, the older sector is deleted; one single block instance, which might be in an unstable state, remains in FLASH.
During next startup this instance might have become unreadable (as described above).

It is typically not an issue for frequently written data, as they are rewritten during run-time (at latest during shut-down). Therefore they have additional instances.

Finally, not every abort causes data loss: There are only few FLASH write accesses considered critical regarding this issue (In general, there are two critical FLASH pages per configured user block, resulting in a very limited time-frame that must be hit to force this issue).

Cancellation:
Cancellation happens in general if Fls_Cancel is called. Usually it is called by FEE (Fee_MainFunction), if Fee_Cancel was called, which, in turn is usually called by NvM.
Calling NvM_WriteBlock, NvM_EraseNvBlock or NvM_InvalidateNvBlock may cause NvM to issue a cancel request to lower layers.
Furthermore, if MICROSAR NvM is used, the shut-down process may be cancelled using a special service, NvM_KillWriteAll.

In which configuration does this happen:

It happens in any configuration, however its probability highly depends on write frequency and initial flash usage (defined by block configuration), thus on frequency of sector switch operations (per partition).
Additionally it depends on separation of data, using partitions

Cancellation via NvM write(-like) requests occurs if it was configured to use "Job prioritization" and there were blocks configured with priority 0.
NvM_KillWriteAll requires parameter "/MICROSAR/NvM/NvMCommonVendorParams/NvMKillWriteAllApi" to be TRUE.

ESCAN00087476	Possible data loss due to aborted Flash write access

This issue is known to occur on Renesas' RV40F flash devices (e.g. RH850F1x).

Resolution Description:
Workaround:

Since the issue generally affects infrequently written data; and because usually such data blocks are most crucial to an ECU, it is highly recommended to use partitions in order to minimize risks. Gathering (very) infrequently written data in a separate partition minimizes the need to copy them around. Therefore there are much less critical situations.

Resolution:

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

ESCAN00091216	Fee_ForceSectorSwitch may do more work than expected
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	8.01.01
Fixed in versions:	

Problem Description:
What happens (symptoms):

Fee_ForceSectorSwitch executes sector formatting three times (erases), resulting in increased run-time.
One Sector is being formatted twice. Additionally FEE checks all configured blocks whether they need to be copied. This results in pre-allocation of internal data structures in flash (link-tables for blocks configured as Datasets).

This is not directly observable outside FEE; as it does not affect its clients.

When does this happen:

It happens (per partition), if both sectors are INVALID (both headers erased or defective) or if "older" sector is empty (VALID sector header but no other content), which might be the result of another (aborted) sector switch.
Expected behavior in this case: Just erase both sectors.

This behavior has not other effects than increased processing time (which, in practice is still shorter than a sector switch with existing data to be copied) and the cost of one unnecessary sector format.
The described scenario - two INVALID sectors - is also very exceptional; it happens only initially with a brand-new (erased) Flash, or if the Flash is defective again.

This issue documents unintended/unwanted internal behavior.

In which configuration does this happen:

It happens, if Fee_ForceSectorSwitch API (FeeSpecificFeatures/FeeForceSectorSwitchApi), or special Feature "Data Conversion" (FeeSpecificFeatures/FeeDataConversionApi) was enabled.

Resolution Description:
Workaround:

No workaround necessary.

Resolution:

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

ESCAN00091357	EcuM detects the wrong Wakeup Source during startup as the reset reason
Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

As a consequence of the wrong detected Wakeup Source as reset reason the wrong channel might be woken up. Depending on the configuration it is also possible that no channel is woken up.

When does this happen:

At startup of the Ecu.

In which configuration does this happen:

Only in configurations with at least one wakeup source with configured parameter ResetReasonRef [EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMWakeupSource/EcuMResetReasonRef]

AND

The Mcu data type Mcu_ResetType is bigger than uint8 (0xFF), e.g. uint16 or uint32.

Resolution Description:
Workaround:

No workaround available.

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.

ESCAN00088171	FEINT exception during ShutdownOS leads to memory violation
Component@Subcomponent:	Os_PlatformRh850Gen6@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

If ShutdownOS is interrupted by FEINT exception then a memory violation is reported by the OS.
ProtectionHook and ShutdownHook are called with error code = E_OS_PROTECTION_MEMORY = 0x0E
and global variable osCtrlVars0.ossLastError = osdErrEXMemoryViolation = 0xA501

When does this happen:

During runtime when application calls ShutdownOS

In which configuration does this happen:

SC3 and SC4

Resolution Description:

Workaround:

Disable FE level interrupts before calling ShutdownOS.
If ShutdownOS returns to caller then enable FE level interrupts.

Resolution:

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

ESCAN00089164	The EcuM stays in RUN state even if EcuM_KillAllRunRequests has been called
Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	3.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The ECU stays in RUN state, even if anyone has called the API EcuM_KillAllRunRequests.

When does this happen:

Always after EcuM_KillAllRunRequests() has been called and at least one channel is in a state unequal COMM_NO_COM_NO_PENDING_REQUEST.

In which configuration does this happen:

Only in configurations with ECUM_FIXED_BEHAVIOR is active (EcuM/EcuMGeneral/ EcuMEnableFixBehavior).

Resolution Description:

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

Workaround:

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

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

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

- Example Appl_ComM_EcuM.h:

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

# define ComM_GetState Appl_ComM_GetState
#endif
```

- Example Appl_ComM_EcuM.c:

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

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


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

 return retVal;
}
```

Resolution:

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

ESCAN00090356	Generator does not generate all Wakeup Sources defines to EcuM_Generated_Types.h
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

Some defines to access a wakeup source bitmask are not generated to EcuM_Generated_Types.h.

This might cause that during runtime a wakeup event is not handled as expected.

When does this happen:

In configurations with a wakeup source which has the prefix 'ECUM_WKSOURCE_' and a post fix which also exists as a wakeup source without this prefix.

Example:

Wakeup Source Name 1: 'ECUM_WKSOURCE_CHANNEL1'
Wakeup Source Name 2: 'CHANNEL1'

In which configuration does this happen:

If a Wakeup Source is configured with prefix ECUM_WKSOURCE_<NAME> and another Wakeup Source has only <NAME>

Resolution Description:
Workaround:

Prevent Wakeup Sources Names which only difference is the prefix 'ECUM_WKSOURCE_'.

Resolution:

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

ESCAN00091112	Variant Driver Init Lists are called for every Core, even for Slave Cores
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Initialization Code which shall only be called on the Master (BSW) Core is executed on every Code.

When does this happen:

Always during initialization.

In which configuration does this happen:

In all variant Multicore Configurations.

Resolution Description:

Workaround:

The generated driver init functions has to be surrounded by the following if condition:

```
if(GetCoreID() == ECUM_CORE_ID_BSW)
```

Resolution:

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

ESCAN00091305	EcuM with fixed state machine causes a Det error in Dem_Init because this module has been initialized two times
Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	3.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

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

When does this happen:

During runtime of the EcuM API EcuM_StartupTwo().

In which configuration does this happen:

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

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

Resolution Description:

Workaround:

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

Resolution:

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

ESCAN00091611	Wrong negative response code for ControlDTCSetting
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	4.00.00
Fixed in versions:	8.00.08, 11.01.00

Problem Description:

What happens (symptoms):

Dcm sends NRC 22 (Conditions not Correct) instead of NRC 31 (RequestOutOfRange)

When does this happen:

When requesting $85 for a different DTC group than 0xFFFFFF (ALL DTCs)

In which configuration does this happen:

/Dcm/DcmConfigSet/DcmDsp/DcmDspControlDTCSetting/
DcmSupportDTCSettingControlOptionRecord == TRUE

Resolution Description:

Workaround:

There is no real use-case for return code DEM_CONTROL_DTC_SETTING_N_OK, so it can be replaced by DEM_CONTROL_DTC_WRONG_DTCGROUP to get the correct NRC response from Dcm:

Since version 5.00.00: Create a user config file for Dem with following content:

```
#if defined DEM_SOURCE
# undef DEM_CONTROL_DTC_SETTING_N_OK
# define DEM_CONTROL_DTC_SETTING_N_OK DEM_CONTROL_DTC_WRONG_DTCGROUP
#endif
```

Set the user configuration file for LATE inclusion (configuration parameter DemUserConfigFilePost)

Before version 5.00.00, introduce the code above into the include path of Dem.c, e.g. by adding it a user config file to the NvM or Dcm

Resolution:

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

ESCAN00091755	PID $01: OBD relevant montioring functions are always completed
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	12.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

No matter whether an emission related montioring function is completed or not, it will be always reported as completed.

When does this happen:

Always when PID $01 is reuqested.

In which configuration does this happen:

Dem is licensed for OBD 2
AND
Major Monitors is supported (in Dem_Cfg.h: DEM_CFG_SUPPORT_MAJOR_MONITORS == STD_ON)
AND
Event Availability is not supported (in Dem_Cfg.h DEM_CFG_SUPPORT_EVENTAVAILABLE == STD_OFF)

Resolution Description:

Workaround:

Enable Event Availability support in DaVinci Configurator Pro. (/MICROSAR/Dem/DemGeneral/DemAvailabilitySupport)

ESCAN00091756	Random modification of memory
Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	6.00.00
Fixed in versions:	10.00.00

Problem Description:

What happens (symptoms):

Memory corruption due to function argument mismatch.

The RTE port interface operation is defined incorrectly:
wrong: Std_ReturnType <>(uint8* data, Dem_EventIdType EventId)
expected: Std_ReturnType <>(Dem_EventIdType EventId, uint8* data)

Due to the switched argument position, the event ID is passed in place of the destination pointer, leading to random memory modification.

When does this happen:

Depending on configuration of Dem/DemGeneral/DemDataElement/
DemExternalCSDataElementClass/DemDataElementStoreNonVolatile
the issue will trigger when an event is stored in the event memory, or when a diagnostic service requests event data using service 19 04 or 19 06

In which configuration does this happen:

Dem/DemGeneral/DemUseRte == TRUE
AND
Dem/DemGeneral/DemDataElement/DemExternalCSDataElementClass/DemDataElementUsePort == TRUE
AND
Dem/DemGeneral/DemDataElement/DemExternalCSDataElementClass/
DemUseEventIdArgumentForCallback == TRUE

Resolution Description:

Workaround:

Set Dem/DemGeneral/DemDataElement/DemExternalCSDataElementClass/
DemDataElementUsePort == FALSE and provide a wrapper function in Dem/DemGeneral/
DemDataElement/DemExternalCSDataElementClass/DemDataElementReadFnc
The wrapper function can call the application port interface with the correct order of arguments.

e.g.

```
Std_ReturnType ApplWrapper(Dem_EventIdType EventId, uint8* Data)
{
 return Rte_Call_<ApplicationPortInterfaceName>(Data, EventId);
}
```

Please note, the issue correction will change the port interface description as defined in the technical reference. The application SWC interface needs to be adapted accordingly.

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

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.

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.

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.

ESCAN00080731	Fee_ForceSectorSwitch might do nothing
Component@Subcomponent:	If_AsrIfFee@Implementation
First affected version:	8.00.00
Fixed in versions:	8.00.82

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.

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.

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.

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 comply to 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.

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:

ESCAN00084863	API names of 3rd-party Wdg drivers may not be recognized correctly

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.

Alternative workaround using a wrapper (trigger function in the driver is Wdg_Trigger() in the example):

Insert in the Makefile the following line:

CFLAGS_WDGIF_LCFG = --include _Wdg_Wrapper.h

and include and compile the following files:

```
/* _Wdg_Wrapper.h */
#include "Std_Types.h"

extern void Wdg_SetTriggerCondition(uint16);
```

```
/* _Wdg_Wrapper.c */
#include "_Wdg_Wrapper.h"
#include "Wdg.h"

void Wdg_SetTriggerCondition(uint16 x)
{
 Wdg_Trigger();
}
```

Resolution:

ESCAN00085447	generator exits unsuccessfully with no error message (on empty peripheral region identifier)
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.

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.

ESCAN00085897	Generator warning: Calculation of time macro is inaccurate.
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	9.01.06
Fixed in versions:	9.01.08

Problem Description:
What happens (symptoms):

The generator states that one or more timing macros have been computed inaccurately. The stated deviation is slightly higher than 1ppm.
Such a stated deviation is typically no issue of the generator. However, there are some cases where the approximation of timing macros stops with a deviation slightly above 1ppm and a better approximation is possible. the timing macro approximation typically stops when the deviation is below 1ppm. In this case the generator does not generate a warning.

When does this happen:

This happens always and immediately at generation time. However, this happens only in rare configurations.

In which configuration does this happen:

This happens only if the parameter SECONDSPERTICK is set to a rarely ever used value.

Resolution Description:
Workaround:

No workaround necessary, the small deviation should be acceptable in any case.

Resolution:

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

ESCAN00086270	Generator does not allow usage of multiple hardware timer types
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	9.01.06
Fixed in versions:	

Problem Description:
What happens (symptoms):

Generator reports multiple definitions of the internally added COUNTER attributes.

When does this happen:

When more than one hardware timer type is used in the platform generator.

In which configuration does this happen:

Independent from the configuration.

Resolution Description:
Workaround:

No workaround available.

ESCAN00086809	Generator Output differs on different PCs
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	

Problem Description:
What happens (symptoms):

Even with same input data (OIL) generated data differs in its content if it is generated on different PC (maybe also on the same?)
In the specific case event masks and arrays oskResCeilingPrio oskResCeilingPrioMask are affected.

The full system needs to be compiled again, although the configuration has not changed.

When does this happen:

During generation

In which configuration does this happen:

Potentially all where multiplicity of a OS configuration element is greater 1.

Resolution Description:
Workaround:

No workaround necessary as the described behaviour causes no malfunction of the embedded code (only just potential confusion of customer).

Resolution:

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

ESCAN00087912	Compiler error: Double definition of CC
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler states an error message as CC has already been defined.

When does this happen:

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

In which configuration does this happen:

This happens in configurations where the application code has also defined a macro named CC.

Resolution Description:

Workaround:

Remove definition of CC from the application.

Resolution:

Rename the conformance class define from 'CC' to 'osdCC'.

ESCAN00087932	ConsistencyRT00002 reports IllegalStateException: Optional.get() cannot be called on an absent value
Component@Subcomponent:	Cdd_AsrCddCfg5@GenTool_GeneratorMsr
First affected version:	5.00.00
Fixed in versions:	5.00.01

Problem Description:

What happens (symptoms):

The validation result with ID ConsistencyRT00002 is reported in the DaVinci Configurator.
In the validation details contains the following info:
Exception: java.lang.IllegalStateException: Optional.get() cannot be called on an absent value

When does this happen:

During live-validation of the EcuC model in the DaVinci Configurator

In which configuration does this happen:

This issue happens if no related /MICROSAR/PduR/PduRBswModules container(s), referencing the Cdd module(s) is existing
AND
In the Cdd module Rx/Tx Pdus routed via the PduR are configured

Resolution Description:

Workaround:

Add the missing /MICROSAR/PduR/PduRBswModules referencing the exsting Cdd module(s) and restart DaVinci Configurator.

Resolution:

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

ESCAN00088524	Compiler error: Undeclared identifier in the initialization structure
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	8.01.00

Problem Description:
What happens (symptoms):

Compiler throws an error for an undeclared identifier used in the root initialization structure.

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 contains multiple predefined variants (selectable)
AND
array or struct symbols are generated to the configuration class precompile
AND
isReduceConstantData2Define() returns true
AND
isInterfacesForDeactivatedData() returns true

Resolution Description:
Workaround:
---
if isInterfacesForDeactivatedData() is user configurable a workaround is available else not.

Resolution:

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

ESCAN00088979	Activation of an auto started task is not performed
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	6.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The OS does not perform the activation of a task which has a configured auto start.

When does this happen:

This happens always and immediately at OS start.

In which configuration does this happen:

This happens only on configurations where an auto start is configured for a task and the application mode is not configured, this task shall be activated in. This is an erroneous configuration which is not detected by the generator.

Resolution Description:

Workaround:

No workaround needed.
The configuration has to be corrected.

Resolution:

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

ESCAN00089176	mssv does not resolve all symbols for checking
Component@Subcomponent:	Elisa__core@Application
First affected version:	1.05.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The function FindConstSymbol provided to checker plugins does not resolve all symbols

When does this happen:

Always, if an mssv plugin reads in several *.c files

In which configuration does this happen:

Always

Resolution Description:

Workaround:

Do not pass several *.c files to mssv plugins if the checks rely on the functions FindConstSymbol() und FindVariablesSymbol()

Resolution:

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

ESCAN00089766	CDD90025: Error at validator runtime
Component@Subcomponent:	Cdd_AsrCddCfg5@GenTool_GeneratorMsr
First affected version:	3.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

An exception is thrown, if multiple CDDs with J1939RmContribution are configured.

When does this happen:

During validation.

In which configuration does this happen:

In configurations with multiple J1939RmContribution CDDs.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00089834	Generator expects a configuration which is not AUTOSAR conform
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	6.20.07

Problem Description:
What happens (symptoms):

Generator refuses to generate with messages:
Missing value assignment for attribute CC with NO_DEFAULT.
Missing value assignment for attribute SCHEDULE with NO_DEFAULT.

When does this happen:

The configuration tool other than Vector Configurator is used to create an AUTOSAR ECUC file.

In which configuration does this happen:

The configuration contains AUTO value for CC and/or SCHEDULE.

Resolution Description:
Workaround:

As a workaround, the correct settings of the parameters may be used (instead of empty setting which is allowed by the standard)

Resolution:

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

ESCAN00089910	Windows states that the OS generator does not work anymore
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	5.28.80
Fixed in versions:	

Problem Description:

What happens (symptoms):

The OS generator erroneously stops working with an error but only the error message of Windows that the generator does not work anymore.

When does this happen:

This happens always and immediately if the configuration is as described below. It happens even at configuration time as part of a cyclic validation.

In which configuration does this happen:

This happens only if an expiry point of a schedule table is configured to be cyclic and the cycle time is configured to be zero:

/.../Os/OsScheduleTable/OsScheduleTableExpiryPoint/OsScheduleTableEventSetting/ OsScheduleTableCyclic/TRUE/OsScheduleTableCycleTime = 0

(This is an undetected wrong configuration)

Resolution Description:

Workaround:

Simply correct the configuration to use a cycle time bigger than zero.

Resolution:

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

ESCAN00090715	Compiler error: BswM_Lcfg.c: '<Symbol>' undefined in case of individual PostBuild process
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

The compiler throws the following error:

BswM_Lcfg.c: '<Symbol>' undefined, assuming extern returning int

When does this happen:

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

In which configuration does this happen:

In configurations with modules which uses an individual PostBuild process (configured via EcuC/EcucGeneral/PostbuildLoadable/IndividualPostBuildLoadableModule)

AND

Module is initialized by the BswM via Auto Configuration: Module Initialization.

Resolution Description:
Workaround:

Adapt the corresponding init action (e.g. INIT_Action_Can_Init) and remove / adapt the access to the global configuration pointer in the parameter BswMUserCalloutFunction, so that a valid pointer is passed to the modules init function.

Resolution:

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

ESCAN00090724	Global Configuration Data is created as Array but should be a structure
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

The EcuM_GlobalConfigRoot is generated as an array EcuM_GlobalPcConfig[n] but should be a structure with n elements which can be accessed per variant.

This lead to an compiler error like the following:
EcuM_Init_Cfg.c:65:46: error: conflicting type qualifiers for 'EcuM_GlobalConfigRoot'

When does this happen:

Always during generation

In which configuration does this happen:

Only in PB-S configurations with individual PB modules.

Resolution Description:
Workaround:

Adaption of the files EcuM_Init_Cfg.h and EcuM_Init_Cfg.c. The adaption of these files is okay because these are template files.

Adapt the EcuM_Init_Cfg.h as follows:
...
extern CONST(EcuM_GlobalConfigRootType, ECUM_CONST) EcuM_GlobalConfigRoot;
...

Adapt the EcuM_Init_Cfg.c as follows:
....
CONST(EcuM_GlobalConfigRootType, ECUM_CONST) EcuM_GlobalConfigRoot =
{
<CONTENT>
};
....

Resolution:

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

ESCAN00090803	Windows states that the OS generator does not work anymore
Component@Subcomponent:	Os_CoreGen6@Generator
First affected version:	6.16.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Windows states that the OS generator does not work anymore
- no output is generated
- no error message will be displayed

When does this happen:

When working with Autosar XML configurations
This issue cannot occur when working with OIL files

In which configuration does this happen:

If the short name of the choice container "OsIsrUseSpecialFunctionName" is changed (and not longer "OsIsrUseSpecialFunctionName").

Resolution Description:

Workaround:

The choice container must be named "OsIsrUseSpecialFunctionName"

Resolution:

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

ESCAN00090998	Configuration tool reports Rte90005 exception because of java.lang.IllegalArgumentException
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.08.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The configuration tool reports Rte90005 - Generator (MICROSAR RTE Generator) failure, because of an exception
- Exception in Rte generator during Generation encountered: java.lang.IllegalArgumentException

When does this happen:

This happen during generation phase.

In which configuration does this happen:

This can happen sometimes in configurations that contain RTE errors found in calculation or validation phase.

Resolution Description:

Workaround:

Solving the reported RTE errors messages.

Resolution:

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

ESCAN00091041	Compiler error: syntax error : identifier 'Dem_Cfg_StorageConditionStateIterType'
Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	9.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

If no DemStorageConditionGroup are used;
Compiler errors in Dem_Lcfg.h due to missing / undefined type:
syntax error : identifier 'Dem_Cfg_StorageConditionStateIterType'
syntax error : ';'

If Storage condition groups are configured, but none are used:
Compiler errors in Dem.c:
error C2065: 'DEM_CFG_NO_STORAGECONDITIONGROUPTABLEIDXOFEVENTTABLE' : undeclared identifier

When does this happen:

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

In which configuration does this happen:

DemStorageConditionSupport == True
AND
Some DemStorageCondition containers are configured
AND
No DemEventParameter/DemEventClass/DemStorageConditionGroupRef are used, so eventually the storage condition support is unused.

Resolution Description:

Workaround:

Set DemStorageConditionSupport == False

OR

Create at least one DemStorageConditionGroup
AND Reference at least one DemStorageCondition in that DemStorageConditionGroup
AND Reference at least one DemStorageConditionGroup in a DemEventParameter/DemEventClass/DemStorageConditionGroupRef

Resolution:

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

ESCAN00091118	EcuM causes a Rte Det error (RTE_E_DET_UNINIT) in the shutdown sequence while the Nvm write all is performed
Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	3.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The Rte throws a Det error with the ID RTE_E_DET_UNINIT during the shutdown sequence.

When does this happen:

Always during the NvM_WriteAll() is performed.

In which configuration does this happen:

Only in configurations with all the following parameters are set to true:

/ActiveEcuC/EcuM/EcuMGeneral/EcuMEnableFixBehavior
/ActiveEcuC/EcuM/EcuMFixedGeneral/EcuMModeSwitchRteAck
/ActiveEcuC/EcuM/EcuMFixedGeneral/EcuMIncludeNvramMgr
/ActiveEcuC/Rte/RteGeneration/RteDevErrorDetect

Resolution Description:

Workaround:

The only workaround is to filter this DET message.

Resolution:

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

ESCAN00091136	Compiler error: 'Fls_Cancel' undefined; assuming extern returning int
Component@Subcomponent:	If_AsrIfFee@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

Compiler issues the following error:

1> Fee_Processing.c
1>..\..\..\..\external\BSW\Fee\Fee_Processing.c(1818): error C4013: 'Fls_Cancel' undefined; assuming extern returning int

When does this happen:

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

In which configuration does this happen:

This happens in all configurations where Fee is used with FLS Driver and 'Fls_Cancel' API is disabled in FLS.

Resolution Description:
Workaround:

Enable 'Fls_Cancel' API in FLS Driver.

Resolution:

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

ESCAN00091277	Compiler error: multiple variable declarations for data conversion in task
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.12.00
Fixed in versions:	1.13.00

Problem Description:
What happens (symptoms):

There are multiple data declarations of type
<dataType> dataSig<dataType>;
at the beginning of a task.

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:

Implicit read or write accesses are configured for data elements mapped to the system signal with data conversion and the same base type. In addition, the data accesses have to be used by multiple runnables within the same task.

Resolution Description:
Workaround:

No workaround available. Try to modify one of the conditions leading to the error.

Resolution:

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

ESCAN00091373	RTE01064 error in case a BSW module provides core service SWCs with mapped server runnables
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.09.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

RTE generation is incorrectly aborted with an RTE01064 error message.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the configuration contains multiple service SWCs for the same BSW module.

Resolution Description:
Workaround:

Do not map the server runnable.

Resolution:

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

ESCAN00091455	A RuntimeException "unknown DataTapeRep enumeration" for sint64 is thrown
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	6.00.00
Fixed in versions:	8.03.01

Problem Description:
What happens (symptoms):

A RuntimeException "unknown DataTapeRep enumeration" for sint64 is thrown at generation time.

When does this happen:

Always and immediately under specific circumstances. See in which configuration does this happen.

In which configuration does this happen:

Any configuration using the EComStackDataTypeRep sint64.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00091629	Use of uninitialized value in case no core with id 0 exists
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

RTE generator aborts with use of uninitialized value.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the configuration contains no OsApplication that is assigned to core 0.

Resolution Description:

Workaround:

Create an OsApplication an assign it to core 0.
Add a task to the OsApplication.

Resolution:

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

ESCAN00091780	Compiler error: Pim variable name conflicts with parameter name of server runnable in RteAnalyzer stub
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

RteAnalyzer reports a compilation error.

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 per instance memory with the same name as an argument for a client-/server operation.

Resolution Description:

Workaround:

Rename the per instance memory.

Resolution:

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

ESCAN00091872	RTE950005 when the configuration contains AR3 modules
Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	4.13.00

Problem Description:
What happens (symptoms):

RTE generation aborts with a class cast exception.

When does this happen:

During generation.

In which configuration does this happen:

When the configuration contains BSWMD modules with AR3 schema.

Resolution Description:
Workaround:

Delete the MODULE-DESCRIPTION-REF for all AR3 modules in the BSWMD.
Ignore the Cfg00028 A module configuration does not reference a bsw implementation messages

Resolution:

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

2.4 Not Released Functionality

Not released functionalities are modules and features that have not yet passed a complete development cycle (they are e.g. not or only partly tested). For serial production projects the integrator has to ensure that all BETA features are disabled as indicated. If a ESCAN affects a complete BSW module, the module must not be used for serial production.

Index

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:

ESCAN00087538	BETA version - the module is in BETA state
Component@Subcomponent:	Rte_Analyzer@Application
First affected version:	0.05.00
Fixed in versions:	

Problem Description:

What is the impact of BETA software:

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

Which functionality is BETA:

The complete module is in BETA state

Resolution Description:

ESCAN00090630	BETA version - the BSW module has a feature with BETA state (FEAT-1810)
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	12.00.00
Fixed in versions:	

Problem Description:
What is the impact of BETA software:

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

Which functionality is BETA:

The following feature/function is in BETA state.
- Adapt DCM and DEM to FCA Requirements

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00091210	BETA version - the BSW module has a feature with BETA state (FEAT-1726)
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.12.00
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.
- This feature allows connection between Nv ports and S/R ports

To ensure that only productive code is used verify that:
- no Nv port is connected with a S/R port

Resolution Description:

2.5 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

ESCAN00067159	Compiler warning: cast truncates constant value MemService_AsrNvM@Implementation
ESCAN00074793	Compiler warning: Condition is always constant Diag_Asr4Dem@Implementation
ESCAN00081825	Compiler warning: non-portable OS code compiled with --asm_warnings Os_PlatformRh850Gen6@Implementation
ESCAN00087501	Compiler warning: "signed/unsigned mismatch" due to missing cast in 0:N or 1:N indirections CommonAsr_ComStackLib@GenTool_GeneratorMsr
ESCAN00088061	BswM_Lcfg.c: warning: 'function' : conversion from 'const BswM_ImmediateUserStartIdxOfModeReqeustMappingType' to 'BswM_SizeOfImmediateUserType', possible loss of data SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00088362	Compiler warning: "cast truncates constant value" with signed data CommonAsr_ComStackLib@GenTool_GeneratorMsr
ESCAN00089832	Compiler warning: GHS internal MISRA checker finds violations Os_CoreGen6@Implementation
ESCAN00091367	Compiler warning: unreferenced local variable for data conversion in task Rte_Core@Implementation

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.

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.

ESCAN00087501	Compiler warning: "signed/unsigned mismatch" due to missing cast in 0:N or 1:N indirections
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	8.01.00

Problem Description:

What happens (symptoms):

"Signed/unsigned mismatch" compiler warning due to missing cast for the subtracted indirection length.
The length macro of a 0:N or 1:N indirection calculates the length through endIndex - startIndex. This subtraction can be interpreted by the compiler as a signed value without a explicit unsigned cast.

When does this happen:

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

In which configuration does this happen:

---

any configuration using 0:N or 1:N Indirections with the length member
AND
the indirection configuration class is PRE-COMPILE

Resolution Description:

Workaround:

Perform a cast in your embedded code.

Resolution:

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

ESCAN00088061	BswM_Lcfg.c: warning: 'function' : conversion from 'const BswM_ImmediateUserStartIdxOfModeReqeustMappi to 'BswM_SizeOfImmediateUserType', possible loss of data
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

BswM_Lcfg.c: warning: 'function' : conversion from 'const BswM_ImmediateUserStartIdxOfModeReqeustMappingType' to 'BswM_SizeOfImmediateUserType', possible loss of data

When does this happen:

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

In which configuration does this happen:

All

Resolution Description:

ESCAN00088362	Compiler warning: "cast truncates constant value" with signed data
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	8.01.00

Problem Description:
What happens (symptoms):

Compiler warns for "cast truncates constant value" due to cast of subtracted signed data.

When does this happen:

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

In which configuration does this happen:
---
your component generator generates signed data in the configuration class precompile
AND
your component generator implementation returns in isReduceConstantData2Define() true
AND
your component generator implementation returns in getDataDeduplicationStrategy() != EDataDeduplicationStrategy.NONE

Resolution Description:
Workaround:

If the values for isReduceConstantData2Define() and getDataDeduplicationStrategy() are user configurable, you have a workaround else not.

Resolution:

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

ESCAN00089832	Compiler warning: GHS internal MISRA checker finds violations
Component@Subcomponent:	Os_CoreGen6@Implementation
First affected version:	5.28.80
Fixed in versions:	6.20.07

Problem Description:

What happens (symptoms):

MISRA 2004 Rule 10.5: bitwise ~ or << expression must be immediately cast back to underlying type, "unsigned char"

When does this happen:

Always

In which configuration does this happen:

Configuration independent

Resolution Description:

Workaround:

No workaround available. However, the code works properly and the compiler warnings can simply be ignored.

Resolution:

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

ESCAN00091367	Compiler warning: unreferenced local variable for data conversion in task
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.12.00
Fixed in versions:	1.13.00

Problem Description:

What happens (symptoms):

There are unnecessary data declarations of type
<dataType> dataSig<dataType>;
at the beginning of a task.

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:

When all of the following conditions are fulfilled:
- an P-Port is connected to an internal R-Port and an external P-Port
- the data element of the external P-Port is mapped to a system signal with data conversion
- an implicit read accesses is configured for the R-Port
- a write access is configured for the internal P-Port and it is either not implicit or not mapped to the same task
- there are no other implicit accesses using data conversion with the same basetype in the task on which the runnable with the implicit read access is mapped

Resolution Description:

Workaround:

No workaround available. Try to modify one of the conditions leading to this warning.

Resolution:

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

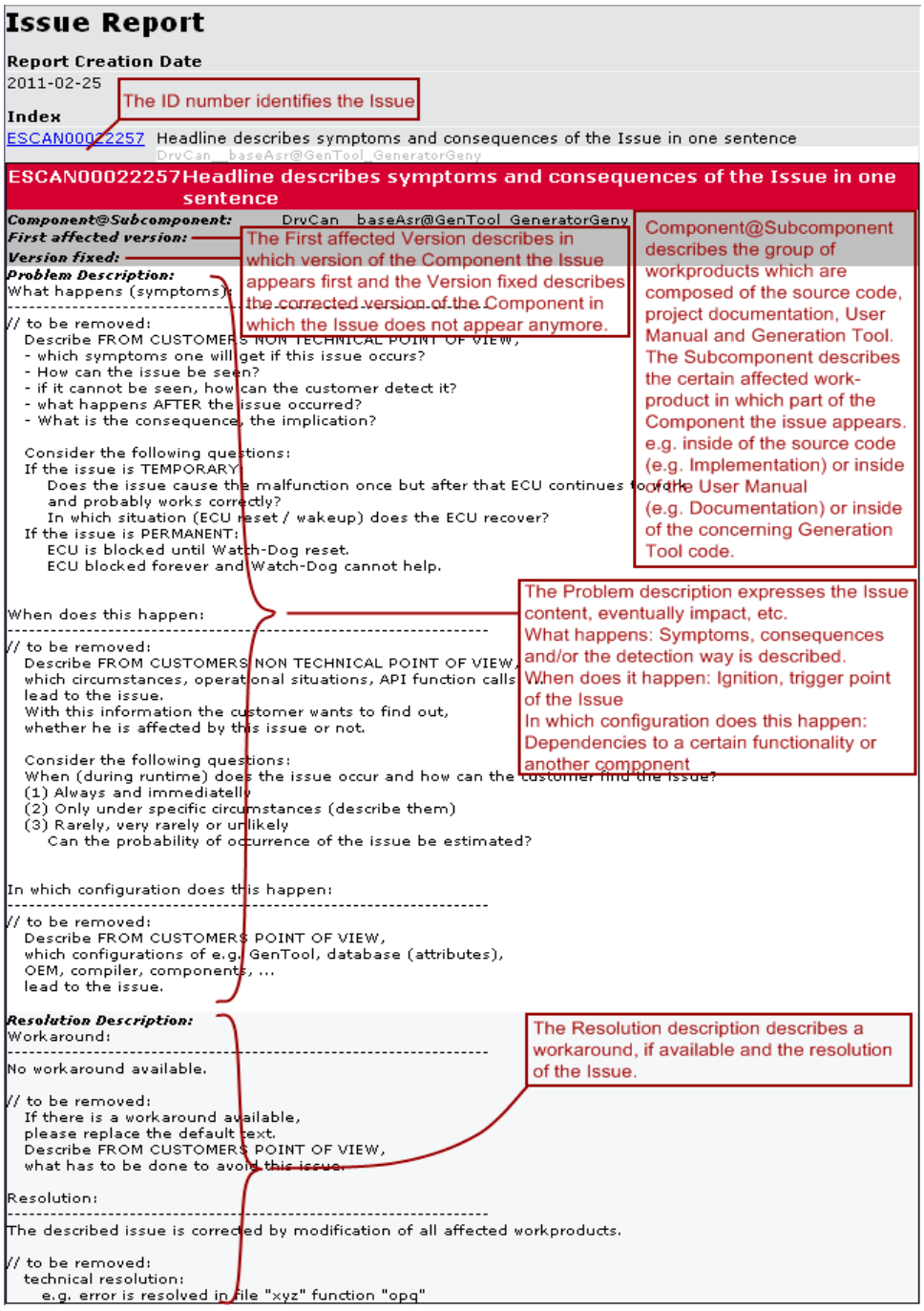
3. New Issues for Information

Issues which should not have an effect on the usage of the license as the issues are relevant for use cases other than those defined in the questionnaire. The list contains issues that have been detected since the last report.

Issues listed in this section are not relevant for the use case that has been documented in the questionnaire provided to Vector. However, the issues may be relevant for other use cases. Also issues that have been accepted or are tolerated by the OEM (as defined in the questionnaire) are reported here.

No issue to be reported.

4. Report Legend



5. Quality Management Contact

Quality Management
Productline Embedded Software (PES)

Vector Informatik GmbH
Ingersheimer Str. 24
D-70499 Stuttgart

Phone: +49 711 80670-3700
Fax: +49 711 80670-399
eMail: QualityPES@vector.com