Issue Report

License Number	**Customer**
CBD1601056	Nexteer Automotive Corporation Package: MSR_Ford_SLP1 - ECU product, "Steering Systems"

Maintenance Expiry Date

2018-03-01

SIP Delivery Date	**SIP Version**
2017-02-10	17.00.04

SLP	**Delivery Number**
MSR_Ford_SLP1	D00

Report Creation Date

2017-02-16

Contact

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

Table of Contents

1. Introduction

1.1 Resolving Issues

Reported issues are not automatically fixed with the next update delivery.
If a reported issue shall be fixed, please contact Vector agree on the issues that can be fixed with upcoming deliveries.
Please note that Vector may fix issues without explicit request.

1.2 Issue Classification

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

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

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

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

2. New Issues

2.1 Safety Relevant Issues

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

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

Index

ESCAN00092747	Mode Switch event in the wrong context leads to unexpected behavior
Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The symptoms are highly dependent on the used ECU. It might happen that the OS error hook or the protection hook is called or the mode management behaves unexpectedly due to inconsistent internal data of the EcuM.

When does this happen:

This happens after calling the API EcuM_SetState() (Used Rte Mode: EcuM_CurrentMode) which can be called by application or by BswM.

In which configuration does this happen:

Only in configurations with EcuM/EcuMFlexGeneral/EcuMModeHandling is set to true

AND additional one of the following topics apply to your configuration.

- A non-MICROSAR RTE is used which provides memory protection
- The EcuM_MainFunction and the application which calls EcuM_SetState() belong to a different OS application

You are not affected if the EcuM_MainFunction and the BswM action which triggers the EcuM_SetState() are running in the same task context.

Resolution Description:

Workaround:

In case that the EcuM_SetState() API is called by application of by BswM (without using the ECU State Handling):

- The application or the BswM_MainFunction() has to run in the same context task context as the EcuM_MainFunction.

In case that the EcuM_SetState() API is called by BswM (with using the ECU State Handling):

- No workaround for this use case.

Resolution:

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

ESCAN00093275 ChainTask() returns to caller with disabled interrupts if the OS detects an error.

Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	1.00.00
Fixed in versions:	1.09.00

Problem Description:

What happens (symptoms):

Interrupts are disabled when function ChainTask() returns to calling task.

When does this happen:

- Task calls ChainTask and inter arrival time violation is detected and the ProtectionHook returns PRO_IGNORE.
- Task calls ChainTask with a TaskID which belongs on different core, and the X-Signal queue is already full.

In which configuration does this happen:

TimingProtection configured (SC2/SC4) and inter-arrival protection used.
or
Multi core task activation/chaining used.

Mind that this issue has been extended in:
ESCAN00093673: ChainTask() returns to caller with disabled interrupts if the OS detects an error.

Resolution Description:

Workaround:

Do not use ChainTask() for self chaining activation
Instead call
ActivateTask(<TaskID>);
TerminateTask(<TaskID>);

Resolution:

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

ESCAN00093505	Automatic mode for calculating slave trigger pattern may expand time duration for reaching safe state or omit reaching safe state at all
Component@Subcomponent:	If_Asr4IfWd@Implementation
First affected version:	5.00.00
Fixed in versions:	5.02.00

Problem Description:

What happens (symptoms):

In case of using the WdgIf with state combiner functionality (multi core) it is possible to configure a so called automatic mode for calculating the trigger pattern of the the slaves out of the given parameter from WdgM (configuration values WdgMTriggerWindowStart and WdgMTriggerConditionValue).

In that case the state combiner parameters WdgIfStateCombinerReferenceCycle, WdgIfStateCombinerSlaveIncrementsMin and WdgIfStateCombinerSlaveIncrementsMax can be left empty. The WdgM calls the WdgIf with API WdgIf_SetTriggerWindow with parameters WdgMDeviceIndex, WdgMTriggerWindowStart and WdgMTriggerTimeout and calculates these values at runtime. However, both given parameters WdgMTriggerWindowStart and WdgMTriggerTimeout are senseless and can be configured arbitrary, because no underlying watchdog device is used - only the state combiner master is connect to a wathdog device.

The result is that a necessary safe state is not reached at all or much later than expected.

Example:

The master is configured as follows:
WdgMTriggerWindowStart: 0ms / 1ms
WdgMTriggerTimeout: 50ms

The slave is configured as follows:
WdgMTriggerWindowStart: 0ms / 1ms
WdgMTriggerTimeout: 20ms

Usually when analyzing these values somebody assume that within 50ms the slave gets 2 or 3 trigger requests. However, with the current implementation of the WdgIf the master checks the slave trigger pattern after 20 master cycles which is 1000ms and expects between 1 and 1000 trigger requests. So if the slave gets triggered once within 1s, the master does not detect any error and continues triggering the underlying watchdog

When does this happen:

This happens if the WdgM detects errors for alive, deadline and / or logical supervision on a "slave core" and discontinues triggering temporarily.

In which configuration does this happen:

This issues does only affect multi-core configurations where:
 - /MICROSAR/WdgIf/WdgIfGeneral/WdgIfUseStateCombiner is enabled AND
 - /MICROSAR/WdgIf/WdgIfStateCombiner is configured AND
 - /MICROSAR/WdgIf/WdgIfStateCombiner/WdgIfStateCombinerGeneral/
WdgIfStateCombinerUseManualMode is disabled

Resolution Description:

ESCAN00093505	Automatic mode for calculating slave trigger pattern may expand time duration for reaching safe state or omit reaching safe state at all
Workaround: -- Do not use automatic mode. Enable the configuration switch /MICROSAR/WdgIf/WdgIfStateCombiner/WdgIfStateCombinerGeneral/WdgIfStateCombinerUseManualMode and configure all parameters within /MICROSAR/WdgIf/WdgIfStateCombiner/WdgIfStateCombinerSlave. Resolution: -- The described issue is corrected by modification of all affected work-products.	

ESCAN00093570 Data consistency problems due to usage of wrong interrupt lock APIs

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.07.00
Fixed in versions:	1.14.00, 1.13.01

Problem Description:

What happens (symptoms):

RTE opens level interrupts within Rte_COMCbk and Rte_LdComCbkRxIndication signal reception callbacks.

If the OS locks the level interrupts for category 2 interrupts, the locking is no longer active during the remaining runtime of the RTE callback and other BSW
and OS code that runs within the interrupt service routine that triggered the reception callback.

This means that other interrupts can interrupt the callback and other BSW and OS code, leading to misbehavior (e.g. data consistency problems)
when the other modules do not support this use case.

When does this happen:

During runtime when a COM or LDCOM callback for queued data reception is called.

In which configuration does this happen:

This only happens when the OS provides the optimized MICROSAR APIs osDisableLevelKM/UM/AM.
This happens when the configuration contains data elements with queued semantics that are mapped to
COM signals or LDCOM PDUs.
Moreover for COM signals, the handling of the COM signals needs to be configured to INTERRUPT (not DEFERRED).

Callbacks for which the issue can occur can be identified in the generated RTE code
by searching for the string Rte_QAddElement in Rte_COMCbk* or Rte_LdComCbkRxIndication* functions.

The issue can only occur when the called Rte_QAddElement function calls
Rte_EnableOSInterrupts() instead of ResumeOSInterrupts().

Resolution Description:

Workaround:

Set the preprocessor flag RTE_DISABLE_ENHANCED_INTERRUPT_LOCK_API

Resolution:

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

ESCAN00093734 NvM allows to generate NULL_PTR for explicit synchronization callbacks for blocks with enabled NvMBlockUseSyncMechanism and NvMUseServicePorts

Component@Subcomponent:	MemService_AsrNvM@GenTool_GeneratorMsr
First affected version:	4.02.00
Fixed in versions:	4.03.00

Problem Description:

What happens (symptoms):

NvM accesses a function pointer with value NULL_PTR. Bad pointer exception/ segmentation fault may occur.

When does this happen:

When a request accessing the block data is setup (multi or single block read or write request). Because of enabled explicit synchronization mechanism NvM want to invoke the callback from its function pointer which is actually NULL_PTR.

In which configuration does this happen:

NvM block with enabled NvMUseServicePorts and enabled NvMBlockUseSyncMechanism but the callbacks NvMReadRamBlockFromNvCallback and NvMWriteRamBlockToNvCallback are still NULL_PTR or empty. NvSwc is not configured to use explicit synchronization.

Resolution Description:

Workaround:

If the Explciit Synchronization mechanism is enabled via NvMBlockUseSyncMechanism, both NvMWriteRamBlockToNvCallback and NvMReadRamBlockFromNvCallback has to be configured (!= NULL_PTR). If NvMUseServicePorts is enabled, the block SWC shall be configured to configure the Explicit Synchronization Callbacks (via RTE). If this is not the case the block shall not use the Explicit Synchronization Mechanism!

Resolution:

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

ESCAN00093745 Data consistency problems because IOC API is called reentrantly in case of IOC queues with queue size 1

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.02.00
Fixed in versions:	1.14.00, 1.13.01

Problem Description:

What happens (symptoms):

The call to an IOC API in an Rte_Call API results in data consistency problems.

When does this happen:

During runtime when multiple clients for a server port operation call the server from concurrent tasks in the same partition.

In which configuration does this happen:

This happens when client and server are located in different partitions and when a server port is connected to multiple client ports.
It happens when client and server are located on different cores and EnforceIOC is set or when the clients are located in multiple partitions.
This will result in the generation of IocSend calls in the affected Rte_Call APIs.
The problem only occurs when the server queue size is 1.
Moreover, this only happens when an operating system is used that does not provide IocSend APIs that are reentrant.

Resolution Description:

Workaround:

Increase the queue size to 2.

Resolution:

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

ESCAN00093787	NvM GenTool_GeneratorMsr does not sync block length between NvM und underlying modules
Component@Subcomponent:	MemService_AsrNvM@GenTool_GeneratorMsr
First affected version:	4.02.01
Fixed in versions:	4.03.00

Problem Description:

What happens (symptoms):

NvM does not synchronize block length with underlying modules.

When does this happen:

When a NvM block is linked to a Fee/Ea block and its size changes.

In which configuration does this happen:
----
every

Resolution Description:

Workaround:

User has to check all Fee/Ea blocks referenced by NvM to be configured correctly - size, priority, datasets.

Resolution:

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

ESCAN00093999 API GetNumberOfActivatedCores() triggers kernel panic if caller is not trusted.

Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	1.06.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

The OS calls kernel panic, if the API GetNumberOfActivatedCores() is called.

When does this happen:

If the caller is not trusted.

In which configuration does this happen:

/MICROSAR/Os/OsOS/OsScalabilityClass = SC3 or SC4
/MICROSAR/Os/OsOS/OsDebug/OsORTIDebugSupport = ORTI_22_ADDITIONAL or ORTI_23_ADDITIONAL

Resolution Description:
Workaround:

Do not use ORTI_22_ADDITIONAL or ORTI_23_ADDITIONAL.
Use ORTI_22_STANDARD or ORTI_23_STANDARD instead.

Resolution:

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

2.2 Runtime Issues without Workaround

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

Index

ESCAN00087310 P2 timings are not recovered on ECU Reset/Power On (Multi Protocol)

Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	4.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

ECU operates after reset with the correct session but with P2 timings of the default session.

When does this happen:

When a recovery info with a non-default target session is provided by the application during reset.

In which configuration does this happen:

- State recovery is supported (i.e. /Dcm/DcmConfigSet/DcmGeneral/DcmStateRecoveryAfterResetEnabled = TRUE)

AND

- Session specific P2 timings are configured

AND

- Multi Protocol is supported

Resolution Description:

Workaround:

No workaround available.

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.

ESCAN00092861 Status of FeeInstance is determined incorrectly

Component@Subcomponent:	If_AsrIfFeeSmallSector@Implementation
First affected version:	1.00.00
Fixed in versions:	1.00.02

Problem Description:

What happens (symptoms):

In some rare cases the content in flash related to a FeeInstance is interpreted incorrectly by SmallSectorFee.

Consequently, further processing of this FeeInstance is incorrect.

An inconsistent instance may be interpreted erroneously as valid.

When does this happen:

If a reset occurs while a FeeBlock is being erased in order to free space for a upcoming write request.

This issue only occurs if erase direction of flash device is from back to front. Datasheet does not specify any details about erase direction.

In which configuration does this happen:

If FlsBlankCheck API is disabled for the corresponding FeePartitionConfiguration, which should not be the case in usual SmallSectorFee use case (RH850).

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00092914 Active protocol is not recovered on ECU Reset/ Power On

Component@Subcomponent: Diag_Asr4Dcm@Implementation
First affected version: 4.01.00
Fixed in versions:

Problem Description:
What happens (symptoms):

ECU operates after reset with a wrong protocol ID which leads to one of the following effects:
 - A new request of the tester, which has initiated the reset, is unexpectedly not responded
OR
 - A new request of the tester, which has initiated the reset, is responded with NRC 0x21 (BusyRepeatRequest)
OR
 - A request of a different tester with higher priority does not cancel an ongoing diagnostic service processing

When does this happen:

At run time (power on/reset) when the API Dcm_GetRecoveryStates returns E_OK (i.e. pre-reset/ power-down states of DCM shall be restored).

In which configuration does this happen:

- Feature "DCM state recovery on ECU power on/reset" is enabled (parameter "DcmStateRecoveryAfterResetEnabled" is TRUE).
AND
- More than one protocol is configured (more than one DcmDslProtocolRow is configured)

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

2.3 Runtime Issues with Workaround

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

Index

ESCAN00061207	DaVinci Configurator 5: Issue Reporting Procedure
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.

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

Component@Subcomponent: Ccl_Asr4SmCan@Implementation
First affected version: 2.00.00
Fixed in versions:

Problem Description:
What happens (symptoms):

Symptom 1)
OS API Invocation ends up with a call to ErrorHook.
Error Cause of this ErrorHook is that OS API is called whereas Interrupts are locked.
Errorcodes would then be:
osdErrATIntAPIDisabled 0x1104U,
osdErrHTMultipleActivation 0x1305U,
osdErrSEIntAPIDisabled 0x4105U,
osdErrSECallContext 0x4106U or
osdErrWEInterruptsDisabled 0x4403U

Symptom 2)
Rte Runnable that is triggered on TxError will not be called when Communication is shut down and Sending Request is outstanding

When does this happen:

----

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

In which configuration does this happen:

If any BswM rule (which belongs to a BswM_CanSM_Indication), is configured as immediate
AND
an OS function is called within the call context of the BswM rule
AND
this OS function must not be called with locked interrupts.

e.g.

Std Rules in BswM are configured
AND
A Rte Runnable is triggered on Tx Error
AND
Osek OS which implements the requirement "If ActivateTask / SetEvent is called with locked Interrupts, reject the action and call an ErrorHook" is used (for example Vector Os)

Resolution Description:
Workaround:

Configure the BswM Rule as deferred

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.

ESCAN00090885 CAN-FD only: Use different Rx-PDUs with same CAN-ID in FullCAN mailboxes

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

Problem Description:

What happens (symptoms):

Message will not be received (filtered by CanIf due to wrong CAN-FD type)

--> This is an invalid configuration !!! (but currently not secured by validator)

When does this happen:

Always and immediately during runtime.

In which configuration does this happen:

CAN-FD messages are used
 AND
two messages/PDUs with same CAN-ID are used (one Classic CAN and one CAN FD)
 AND
these two or at least one of the messages are configured as FullCAN

Resolution Description:

Workaround:

----

do not use PDUs with same CAN-IDs as FullCAN

Resolution:

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

- Add validator to avoid this invalid configuration

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

Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	3.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

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

When does this happen:

During runtime of the EcuM API EcuM_StartupTwo().

In which configuration does this happen:

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

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

Resolution Description:

Workaround:

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

Resolution:

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

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

Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

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

When does this happen:

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

In which configuration does this happen:

- Service 0x27 is supported

AND

- There is more than one security level configured

AND

- Any delay time is configured for any security level (in Dcm_Cfg.h:
DCM_STATE_SEC_RETRY_ENABLED == STD_ON or
DCM_STATE_SEC_DELAY_ON_BOOT_ENABLED == STD_ON)

AND

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

Resolution Description:

Workaround:

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

Therefore, the following workarounds are possible:
1) Increase the DcmTaskTime parameter value.
OR
2) Reduce the timeout value in the corresponding timing parameter.

Resolution:

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

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.

ESCAN00092745 Missing "if" statement

Component@Subcomponent:	DrvCan_Rh850McanAsr@Implementation
First affected version:	2.09.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

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

When does this happen:

At run time.

In which configuration does this happen:

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

Resolution Description:

Workaround:

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

Resolution:

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

ESCAN00092880 EcuM calls the Mcu_SetMode API with invalid mcu modes

Component@Subcomponent:	SysService_Asr4EcuM@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

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

When does this happen:

During sleep phases.

In which configuration does this happen:

In configurations with at least one sleep mode [EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMSleepMode]

AND

The Mcu_ModeType values start with 0 instead of 1.

Resolution Description:

ESCAN00092880 EcuM calls the Mcu_SetMode API with invalid mcu modes

Workaround:

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

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

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

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

****************************************************************************

 /* Add implementation of EcuM_McuSetMode() */

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


 return;
/
****************************************************************************

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

****************************************************************************

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

Resolution:

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

ESCAN00093001 CanNm calls Det_ReportError cyclically when not active in a predefined variant (PB-S)

Component@Subcomponent:	Nm_Asr4NmCan@Implementation
First affected version:	5.00.00
Fixed in versions:	

Problem Description:
What happens (symptoms):

CanNm calls Det_ReportError() cyclically with the following parameters:

ModuleId 0x1f, InstanceId 0x0, ApiId 0x13, ErrorId 0x1
(errorId = CANNM_E_NO_INIT)

The ECU is not blocked.
The CanNm behaves properly according to the selected variant.

When does this happen:

Always during runtime every time the CanNm_Mainfuncation is called by the OS.

In which configuration does this happen:

- /MICROSAR/EcuC/EcucGeneral/EcuCSafeBswChecks == ON
AND
- In Postbuild-Selectable configurations, with at least two predefined variants
AND
- CanNm is not active in one variant
AND
- The predefined variant where the CanNm is not active, is the currently running one.

Resolution Description:
Workaround:

This Det_ReportError call can be ignored.

Permanently filter this Det_ReportError call in the Det component.
Refer to the Technical Reference of the Det for further information.

Resolution:

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

ESCAN00093055 Duplicate BlockNumbers are not detected in configuration

Component@Subcomponent:	If_AsrIfFeeSmallSector@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	1.01.00

Problem Description:

What happens (symptoms):

A BlockNumber uniquely identifies a block in FeeBlockConfiguration. If a BlockNumber is duplicate, FEE may perform a requested job for the wrong block. In case both blocks have different block lengths, the job request will be declined due to failed development checks.

When does this happen:

When BlockNumbers are modified manually via "Set User Defined" attribute in configuration and two (or more) blocks share the same BlockNumber.

In which configuration does this happen:

Every configuration using CFG5 5.14.00 and later.

Resolution Description:

Workaround:

BlockNumbers can be modified manually via "Set User Defined" attribute. Make sure that no BlockNumber occurs twice.

Resolution:

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

ESCAN00093185 Wrong behaviour of hierarchically ordered IPduGroups

Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	12.00.00

Problem Description:

What happens (symptoms):

Activating/ deactivating a parent ComIPduGroup with Com_IpduGroupStart, Com_IpduGroupStop has no influence the ComIPdu of the sub I-PduGroups..
Starting/ stopping the reception deadline monitoring with Com_EnableReceptionDM, Com_DisableReceptionDM will not start/stop the reception deadline monitoring of ComIPdu's which are referred by the Sub-ComIPduGroups of a parent ComIPduGroup.

When does this happen:

This issue will occur whenever Com_IpduGroupStart, Com_IpduGroupStop is called for a parent ComIPduGroup of hierarchically ordered IPduGroups at runtime.
OR
This issue will occur whenever Com_EnableReceptionDM, Com_DisableReceptionDM is called for a parent ComIPduGroup of hierarchically ordered IPduGroups at runtime.

In which configuration does this happen:

ComOptimizedIPduGroupHandling == ON
AND
ComIPduGroups are hierarchically ordered (/MICROSAR/Com/ComConfig/ComIPduGroup/ComIPduGroupGroupRef is configured).

Resolution Description:

Workaround:

Deactivate /MICROSAR/Com/ComGeneral/ComOptimizedIPduGroupHandling .

Resolution:

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

ESCAN00093263 Missing "if" statement

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

Problem Description:

What happens (symptoms):

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

When does this happen:

At run time.

In which configuration does this happen:

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

Resolution Description:

Workaround:

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

Resolution:

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

ESCAN00093335	**Service 0x10: Transition to/from FBL without final response**
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The ECU does not final positive response for diagnostic service 0x10 (DiagnosticSessionControl) after switching into or from the FBL software.

Note:
 This could happen only very rarely or on each diagnostic request of the above described kind. It mainly depends on the Dcm_MainFunction() scheduling.

When does this happen:

At runtime when:
- Service 0x10 with FBL related diagnostic session is requested with set SPRMIB = TRUE (SuppressPositiveResponseMessageIndicationBit).
AND
- The Dcm_MainFunction(Worker)() is not called within the P2server time (i.e. a RCR-RP could be sent prior execution of service 0x10 has started).

In which configuration does this happen:

- DCM is configured to not send final response prior switching to/from FBL (DCM ECUC parameter of all DcmDslProtocolRows: DcmSendRespPendOnTransToBoot == FALSE, resp. in Dcm_Cfg.h #define DCM_DIAG_RCRRP_ON_BOOT_ENABLED == STD_OFF)
- DCM is configured to not send any RCR-RP prior switching to/from FBL (DCM ECUC parameter DcmFinalResponseToFblEnabled == FALSE, resp. in Dcm_Cfg.h #define DCM_SVC_10_RST2BOOT_HIS_ENABLED == STD_ON)
AND
- DCM is configured to have two main-functions (DCM ECUC parameter DcmSplitTasksEnabled == TRUE, resp. in Dcm_Cfg.h #define DCM_SPLIT_TASKS_ENABLED == STD_ON)
AND
 - The scheduling time of the Dcm_MainFunctionWorker exceeds any P2server time.
 (DCM ECUC parameters are in relation: DcmMainFunctionWorkerTaskTime > (DcmDspSessionP2ServerMax - DcmTimStrP2ServerAdjust) for at least one session/diagnostic protocol combination)
 OR
 - Due to heavy CPU load, the execution of Dcm_MainFunctionWorker() is delayed for a time exceeding the P2server for the current request.

Resolution Description:

ESCAN00093335 **Service 0x10: Transition to/from FBL without final response**

Workaround:

Possible workarounds:
1) If there is a requirement to not send any RCR-RP prior FBL-application software switch, then assure (via task priorities, task cycle times) that DCM will never delay the service 0x10 processing prior resetting the ECU. In other words, the ECU shall be reset in any case within the P2server time.
2) If for some integration reasons this is not possible to guarantee the runtime needed till reset, then the configured option "DcmSendRespPendOnTransToBoot" obviously does not make any sense, since sometime there still will be a RCR-RP sent. To make the ECU behavior deterministic, do enable the RCR-RP prior reset option (DcmSendRespPendOnTransToBoot = TRUE).

Resolution:

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

ESCAN00093466	Unexpected DET in Dem_DcmDisableDTCRecordUpdate
Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

DET call DEM_DCMDISABLEDTCRECORDUPDATE_APIID, error code DEM_E_WRONG_CONDITION
Depending on DET configuration, the task calling Dem_DcmDisableDTCRecordUpdate will block due to an endless loop in the DET module.
This can cause unrepsonsive diagnostics or ECU reset by watchdog.

When does this happen:

When processing a diagnostic service 19 04, 19 06, 19 11 or 19 12,
such that Dem_DcmEnableDTCRecordUpdate is called after a call to
Dem_DcmDisableDTCRecordUpdate, but before the Dem task was executed at least once.

This can happen when the Dem task schedule is larger than the Dcm, or in multi-protocol use-cases where a high priority request cancels an ongoning diagnostic request before the Dem main function was called.

In which configuration does this happen:

The Dcm supports protocol cancellation
OR
The BSW task scheduling allows for a situation in which the Dcm will cancel a started request before the Dem task was executed.

Resolution Description:

Workaround:

Ignore the DET report

Resolution:

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

ESCAN00093886	**Service 0x28: Dcm performs Rte_Switch with a wrong mode from the corresponding CommunicationControl ModeDeclarationGroup**
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	7.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Depending on the application use of the CommunicationControl ModeDeclarationGroup, the result of the notification with an unexpected mode can vary. Here some examples:
- If the provided by DCM notification is not used by any SW-C, nothing happens (i.e. issue ticket can be ignored)
or
- If the application evaluates the new communication control mode, wrong action will be executed once the new communication control mode is received in the SW-C.

The following mapping applies here:
Value Expected (Correct) -> Received (Wrong)
 0 ENABLE_RX_TX_NORM DISABLE_RX_ENABLE_TX_NORM_NM
 1 ENABLE_RX_DISABLE_TX_NORM DISABLE_RX_TX_NM
 2 DISABLE_RX_ENABLE_TX_NORM ENABLE_RX_DISABLE_TX_NORM
 3 DISABLE_RX_TX_NORMAL ENABLE_RX_TX_NM
 4 ENABLE_RX_TX_NM ENABLE_RX_DISABLE_TX_NORM_NM
 5 ENABLE_RX_DISABLE_TX_NM DISABLE_RX_ENABLE_TX_NM
 6 DISABLE_RX_ENABLE_TX_NM ENABLE_RX_TX_NORM
 7 DISABLE_RX_TX_NM DISABLE_RX_TX_NORMAL
 8 ENABLE_RX_TX_NORM_NM DISABLE_RX_TX_NORM_NM
 9 ENABLE_RX_DISABLE_TX_NORM_NM ENABLE_RX_TX_NORM_NM
10 DISABLE_RX_ENABLE_TX_NORM_NM DISABLE_RX_ENABLE_TX_NORM
11 DISABLE_RX_TX_NORM_NM ENABLE_RX_DISABLE_TX_NM

When does this happen:

Each time one of the following events occurs:
- The ECU (re-)enters default diagnostic session: the application gets notified for entering mode DISABLE_RX_TX_NORM_NM instead of ENABLE_RX_TX_NORM_NM.
- Each time a valid request for diagnostic service 0x28 is received and processed without any detected errors, the new mode sent to the application will not match the requested new communication state. (see mapping table above)

In which configuration does this happen:

Service 0x28 is internally handled by DCM (in Dcm_Cfg.h #define DCM_SVC_28_SUPPORT_ENABLED is generated to be STD_ON)

Resolution Description:

ESCAN00093886	Service 0x28: Dcm performs Rte_Switch with a wrong mode from the corresponding CommunicationControl ModeDeclarationGroup

Workaround:

Using the provided information on the main page, implement a mapping table to convert the wrong mode passed by DCM. Apply this mapping in all affected SW-Cs.

Example implementation:

```
#include Dcm.h
const Rte_ModeType_DcmCommunicationControl Dcm_ComCtrlModeMap_ESCAN00093886[] =
{
 RTE_MODE_DcmCommunicationControl_DCM_DISABLE_RX_ENABLE_TX_NM
 ,RTE_MODE_DcmCommunicationControl_DCM_DISABLE_RX_ENABLE_TX_NORM
 ,RTE_MODE_DcmCommunicationControl_DCM_DISABLE_RX_ENABLE_TX_NORM_NM
 ,RTE_MODE_DcmCommunicationControl_DCM_DISABLE_RX_TX_NM
 ,RTE_MODE_DcmCommunicationControl_DCM_DISABLE_RX_TX_NORMAL
 ,RTE_MODE_DcmCommunicationControl_DCM_DISABLE_RX_TX_NORM_NM
 ,RTE_MODE_DcmCommunicationControl_DCM_ENABLE_RX_DISABLE_TX_NM
 ,RTE_MODE_DcmCommunicationControl_DCM_ENABLE_RX_DISABLE_TX_NORM
 ,RTE_MODE_DcmCommunicationControl_DCM_ENABLE_RX_DISABLE_TX_NORM_NM
 ,RTE_MODE_DcmCommunicationControl_DCM_ENABLE_RX_TX_NM
 ,RTE_MODE_DcmCommunicationControl_DCM_ENABLE_RX_TX_NORM
 ,RTE_MODE_DcmCommunicationControl_DCM_ENABLE_RX_TX_NORM_NM
};
```

Usage:

```
void ApplFunctionUsingComControlModes(void)
{
 Rte_ModeType_DcmCommunicationControl newMode;

 Rte_Read_<SwcA>_CommControl_Can0(&newMode);

 DoSomethingWithNewMode(Dcm_ComCtrlModeMap_ESCAN00093886[newMode]);
}
```

Note:
 Once this issue is resolved, this workaround mapping table can be removed for ROM/runtime efficiency, but even if left inside the code, will represent a 1:1 mapping, so it the adapted SW-C must not necessarily be adapted again.

Resolution:

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

ESCAN00093905	**Service 0x19: Sub-functions 0x04/0x06/0x10/0x18/0x19 report zero values for snapshot/extended data records**
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Positive responses for diagnostic service 0x19 sub-functions 0x04/0x06/0x10/0x18/0x19 report zero values for snapshot/extended data records.

The correct behavior of the ECU shall be to interrupt the response transmission for lack of further response data. The issue here is that the zero value might be interpreted as real EDR (ExtendedDataRecord) / SSR (SnapshotRecord) data and mislead the ECU's fault analysis.

When does this happen:

At runtime when
- Requesting one of the above listed sub-functions
AND
- The DEM reports initially to DCM a size of the EDR/SSR to be reported different from the later reported actual EDR/SSR sizes.
Note: This happens if the DEM implementation reports initially the record size without considering whether all data within the record is available, but takes the configured maximum possible EDR/ SSR size.

In which configuration does this happen:

- Dcm is configured to use paged-buffer responses for service 0x19 (in Dcm_Cfg.h #define DCM_DEM_API_PAGED_BUFFER_ENABLED is set to STD_ON && #define DCM_PAGED_BUFFER_ENABLED is set to STD_ON)
AND
- Diagnostic service 0x19 with any of the sub-functions 0x04/0x06/0x10/0x18/0x19 is supported by the ECU.
(in the Dcm_Cfg.h any of the #defines DCM_SVC_19_XX_SUPPORT_ENABLED (XX is the sub-function ID above) is generated to STD_ON)

Resolution Description:

Workaround:

Switch back to linear buffer response transmission.

Resolution:

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

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

Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

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

When does this happen:

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

In which configuration does this happen:

For any diagnostic service that has sub-functions supported in different security levels resp. different DcmModeRule pre-conditions.

Example:
- ECU is configured to support the following security levels: locked, level X.
- Service 0x10 has following sub-functions (SF): 0x01 and 0x02.

This issue will occur if:
- SF 0x01 is allowed in all securitly levels (locked one inclusively) i.e. has no security restrictions
- SF 0x02 is allowed only once security access level X is enabled
Any request for service 0x10 0x02 with any wrong length more than two bytes, while the ECU is locked, will result in NRC 0x13 (ICMLOF) instead of 0x33 (SAD).

Please note:
This issue will _not_ occur if:
- All SID 0x10 sub-functions have the same diagnostic security access execution precondition dependency e.g. allowed only in the security level X. In this case not only the sub-functions but the SID 0x10 itself will not be supported in the security access locked level. This will result in NRC 0x33 (SAD), which is no deviation of the ISO specification.

Resolution Description:

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

Workaround:

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

Resolution:

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

2.4 Not Released Functionality

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

Index

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:

ESCAN00088830 BETA Feature - Memory Protection in trusted applications is only provided as a BETA feature

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

Problem Description:

What is the impact of BETA software:

BETA feature:

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

Which functionality is BETA:

The following feature/function is in BETA state.

- Memory Protection in trusted applications.

To ensure that only productive code is used verify that:

- OsTrustedApplicationWithProtection is false for all applications.

Resolution Description:

Workaround:

Do not use memory protection for trusted.

Resolution:

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

ESCAN00089701	BETA Feature - Executing trusted applications in non privileged mode is only provided as a BETA feature.
Component@Subcomponent:	Os_CoreGen7@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What is the impact of BETA software:

BETA feature:

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

Which functionality is BETA:

The following feature/function is in BETA state.

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

To ensure that only productive code is used verify that:

- IsPrivileged is TRUE for all trusted applications

Resolution Description:

API Extensions:

No extension of the API.

API Changes:

No modification of the API.

Module handling changes:

No modification of the module handling.

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

ESCAN00090942	BETA version - the BSW module has a feature with BETA state (FEAT-1914)
Component@Subcomponent:	If_AsrIfCan@Implementation
First affected version:	6.11.00
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.
- Data checksum (Rx and Tx)

To ensure that only productive code is used verify that:
- Following parameters disabled in configuration tool:
/MICROSAR/CanIf/CanIfPrivateCfg/CanIfDataChecksumTxSupport
/MICROSAR/CanIf/CanIfPrivateCfg/CanIfDataChecksumRxSupport

or CanIf_Cfg.h:
- CANIF_DATA_CHECKSUM_RX_SUPPORT STD_OFF
- CANIF_DATA_CHECKSUM_TX_SUPPORT STD_OFF

Resolution Description:

ESCAN00091204	BETA version - the Nm module has a feature with BETA state (FEAT-1865)
Component@Subcomponent:	Nm_Asr4NmIf@Implementation
First affected version:	10.00.00
Fixed in versions:	

Problem Description:

What is the impact of BETA software:

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

Which functionality is BETA:

The following feature/function is in BETA state.
The NmOsek has to support the specific coordination use cases:
- Use different intervals between the Nm_SynchronizationPoint() function call and the expected NmOsek_NetworkRelease() call in dependency of the state of NmOsek
- Use different shutdown times for CanNm and NmOsek on the same channel

To ensure that only productive code is used verify that:
- If Nm Coordination is active in Nm and NmOsek is used in the configuration, then check that in NmOsek the configuration parameter "Wait Bus Sleep Extensions" is inactive

-Nm_Cfg.h contains the following line:
#define NM_WAIT_BUS_SLEEP_EXTENSIONS STD_OFF

Resolution Description:

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:

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

Component@Subcomponent: Diag_Asr4Dcm@Implementation

First affected version: 7.00.00

Fixed in versions:

Problem Description:

What is the impact of BETA software:

BETA software

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

Which functionality is BETA:

The following feature/function is in BETA state.

- The sender/receiver access to the application data.

To ensure that only productive code is used verify that:

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

Resolution Description:

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

Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What is the impact of BETA software:

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

Which functionality is BETA:

The following feature/function is in BETA state.
- Sender/Receiver Ports for NVM or complex types data.

To ensure that only productive code is used verify that:
- ECUC parameter /Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort == USE_DATA_ELEMENT_SPECIFIC_INTERFACES

Resolution Description:

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

Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What is the impact of BETA software:

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

Which functionality is BETA:

The following feature/function is in BETA state.
- Configuration of PNC references for WakeupSources

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

Resolution Description:

ESCAN00092764 BETA version - the BSW module has a feature with BETA state (FEAT-1454)

Component@Subcomponent:	Ccl_Asr4ComMCfg5@Implementation
First affected version:	8.00.00
Fixed in versions:	

Problem Description:

What is the impact of BETA software:

BETA software

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

Which functionality is BETA:

The following feature/function is in BETA state.

- FEAT-1454: Configuration of Switch Ports (switchable per PNC)

The above feature/function has following limitations:

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

To ensure that only productive code is used verify that:

- /MICROSAR/ComM/ComMConfigSet/ComMPnc/ComMPortGroupsPerPnc does not exist.

and

- ensure that COMM_WAKEUPENABLEDOFPNC is defined to STD_OFF in ComM_Cfg.h.

Note: otherwise MSSV fails with error message 'assertion 'COMM_WAKEUPENABLEDOFPNC: "STD_ON"' == '"STD_OFF"' does not hold'.

Resolution Description:

ESCAN00092868 BETA version - the BSW module is in BETA state

Component@Subcomponent:	DrvCry_Rh850Icum@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What is the impact of BETA software:

BETA software

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

Which functionality is BETA:

The complete BSW module is in BETA state

Resolution Description:

2.5 Apparent Issues

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

Index

Index

Index

ESCAN00092720	DataRenamer not working for MICROSAR Define block CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr
ESCAN00092756	Name of OBD calibrateable configuration symbols are out of date Diag_Asr4Dcm@Doc_TechRef
ESCAN00092812	Using trusted function stubs with user defined types may lead to compile error: left operand must be an l-value Os_CoreGen7@GenTool_GeneratorMsr
ESCAN00092839	Linker include files are not generated for non-AUTOSAR cores. Os_CoreGen7@GenTool_GeneratorMsr
ESCAN00092892	Compiler error: function "EcuM_BswErrorHook" has no prototype SysService_Asr4EcuM@Implementation
ESCAN00092955	Compiler error: incompatible types - from 'const <MSN>_PCConfigType *' to 'const <MSN>ConfigType *const SysService_Asr4EcuM@GenTool_GeneratorMsr
ESCAN00092986	Compiler error: Error directive due to inconsistent configuration of Communication Mode Info Cp_AsrXcp@GenTool_GeneratorMsr
ESCAN00092995	CAN-FD message without BRS will not be received DrvCan_Mpc5700McanLl@Implementation
ESCAN00093023	RTE Analyzer sporadically triggers windows application crash dialog Rte_Analyzer@Application
ESCAN00093048	RTE generator generates unused Rte_TrustedCom_SendSignal Rte_Core@Implementation
ESCAN00093079	*_GLOBALSHARED_VAR_NOCACHE_* defines are used by Rte_MemMap.h but not generated by Os_MemMap.h. Os_CoreGen7@GenTool_GeneratorMsr
ESCAN00093109	Missing Limitation for aggregated measurement data size of < 64KB Cp_AsrXcp@Doc_TechRef
ESCAN00093110	Service 0x2F: ECU sends NRC 0x10 to a valid IO control DID request Diag_Asr4Dcm@GenTool_GeneratorMsr
ESCAN00093127	RTE generator does not create IOCs in the OS configuration for single core systems Rte_Core@Implementation
ESCAN00093144	Class cast exception when the configuration contains a subelement mapping for application datatypes Rte_Asr4@GenTool_GeneratorMsr
ESCAN00093236	Compiler error: OS does not compile if only application specific hooks are configured. Os_CoreGen7@GenTool_GeneratorMsr
ESCAN00093294	Invalid key accepted due to inconsistent Csm and CryFord job processing configuration Diag_AsrSwcSecAccess_Ford@Implementation
ESCAN00093309	RTE Analyzer fails due to duplicated mainfunction Rte_Core@Implementation
ESCAN00093317	Value Calculations does not act as expected CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr
ESCAN00093405	Auto Configuration - Invalid multiplicity after manual adaptations of container BswMAvailableActions SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00093413	Auto Configuration Module Initialization - Changed User Include Files always restores SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
ESCAN00093449	A2L compu method RTE_CM_BOOLEAN cannot be used to calibrate boolean values Rte_Core@Implementation

Index

Index

ESCAN00073545	Final FBL response not cancelled on protocol preemption
Component@Subcomponent:	Diag_Asr4Dcm@Implementation
First affected version:	1.05.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The ECU will process the FBL final response even if there is higher protocol request sent.

When does this happen:

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

In which configuration does this happen:

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

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00079399 Linker error: '<Cdd>_Transmit' : undeclared identifier

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.

ESCAN00082111 Data Type Truncation when using more than 255 Mailboxes

Component@Subcomponent:	DrvCan__baseAsr@GenTool_GeneratorMsr
First affected version:	1.05.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

One or more CAN-busses quit transmitting frames or never start transmission.
Another symptom which can occur are Det-Errors in CAN-Write

A reliable indicator for this issue are compiler warnings while compiling CanIf reporting data type truncation.
e.g. "integer conversion resulted in truncation" in Table "CanControllerConfig[]"

When does this happen:

----

runtime effect:
- Always respectively after a small period of time.

compiler warning:
- always while compile time.

In which configuration does this happen:

Whenever there are more than 255 mailboxes (LPDUs) configured in Can driver.
Can_Mailbox[] table is bigger than 255.

Resolution Description:

Workaround:

Set Can/CanGeneral/CanHardwareHandleType to user defined 16 bit.

Resolution:

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

ESCAN00082619 Out of bounds exception

Component@Subcomponent:	DrvCan_Mpc5700McanAsr@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	3.00.02

Problem Description:

What happens (symptoms):

During generation time an "illegal out of bounds" exception appears.
This will result in inconsistent generation data.

When does this happen:

At generation time.

In which configuration does this happen:

This happens if a configuration with Variant PostBuild
and
Standard IDs or Extended IDs only is configured to support Mixed IDs as well.

Resolution Description:

Workaround:

Configure at least one Extended ID Filter entry when using Standard IDs.
Configure at least one Standard ID Filter entry when using Extended IDs.

Resolution:

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

ESCAN00082683	Compiler Error: Datatype PduR__PBConfigIdType not defined by PduR
Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	1.00.00
Fixed in versions:	7.02.00

Problem Description:

What happens (symptoms):

Datatype PduR_PBConfigIdType isn't defined by the PduR as defined within AUTOSR_SWS_PDURouter within ASR4.2.1 Release and therefore not used as return value type of the PduR_GetConfigurationId API. Instead within the API a uint16 is used as return value type. Due to this there could be compilation issues if some other module or integration code is using this data type.

When does this happen:

When the API PduR_GetConfigurationId(void) is used

In which configuration does this happen:

All

Resolution Description:

Workaround:

The problem can be solved by defining the datatype within a User Config File of the PduR.

Resolution:

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

ESCAN00082963	IllegalStateException: The size of the keys: (..) of the ConstStruct: SIDFE/XIDFE does not match the size of the ConstStruct
Component@Subcomponent:	DrvCan_Mpc5700McanAsr@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	3.00.03

Problem Description:

What happens (symptoms):

Generation fails with an Exception similar to this:

IllegalStateException: The size of the keys: (..) of the ConstStruct: SIDFE/XIDFE does not match the size of the ConstStruct

When does this happen:

During Generation / Validation.

In which configuration does this happen:

Mixed ID Configuration used
AND
At least one channel has only Standard / only Extended filters.

Resolution Description:

Workaround:

Ensure that all channels have standard and extended filters configured.

Create a (closed) dummy filter with the missing id type if necessary.

Resolution:

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

ESCAN00084949 Validation error COM02411: solving action does not solve in post-build selectable projects.

Component@Subcomponent:	Il_AsrComCfg5@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The solving action for COM02411, offering to delete the parameter
/MICROSAR/EcuC/EcucPduCollection/Pdu/MetaDataLength
does not delete it. The parameter can be deleted manually though.

When does this happen:

During execution of the solving action.

In which configuration does this happen:

Post-build selectable configurations which match all of the following criteria:
- Contains at least one instance of /MICROSAR/EcuC/EcucPduCollection/Pdu that is not active in all variants.
- Validator message COM02411 is issued in this PDU with the solving action to delete the parameter MetaDataLength.

Resolution Description:

Workaround:

Manually delete the parameter.

Resolution:

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

ESCAN00086253	Validation message PDUR13006 is wrongly shown for N:1 transport protocol routing paths
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	8.01.00

Problem Description:

What happens (symptoms):

The validation message PDUR13006 is shown in the Configurator 5 validation window, if a N:1 transport protocol routing path is configured. The message should only appear for N:1 communication interface routing paths.

When does this happen:

The message is shown if N:1 transport protocol routing paths are configured.

In which configuration does this happen:

N:1 transport protocol routing paths may only be configured for single frame transport protocol messages (only single buffer call is supported). This is validated by another validator.

Resolution Description:

Workaround:

Ignore the validator message for N:1 transport protocol routing paths.

Resolution:

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

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

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

Problem Description:

What happens (symptoms):

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

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

When does this happen:

----

while compile time for DummyFunction and DummyStatement
while runtime for SafeBswChecks

In which configuration does this happen:

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

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

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.

ESCAN00087948 Update Bits are not cleared if Com_IpduGroupControl is called with initialize = false

Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

After a IpduGroup is stared with initialize = false a Signal is transmitted with set Update Bit although signal was not updated since the IpduGroup was stopped.

When does this happen:
----
during the call of Com_IpduGroupControl.

In which configuration does this happen:

If Tx UpdateBits are used
AND
if Com_IpduGroupControl is used with initialize = false

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00087958 Wrong return value of GetTaskState when called from PostTaskHook

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

Problem Description:

What happens (symptoms):

GetTaskState returns SUSPENDED for current task when called from PostTaskHook.
Return 'RUNNING' instead.

When does this happen:

In PostTaskHook the task is still running.

In which configuration does this happen:

Configuration invariant.

Resolution Description:

Workaround:

Do not use the API GetTaskState for the current task in the PostTaskHook.

Resolution:

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

ESCAN00087977 Compiler error: PduR_Lcfg.c: 'PDUR_FCT_IPDUMTX' : undeclared identifier

Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	2.03.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler error: PduR_Lcfg.c: 'PDUR_FCT_IPDUMTX' : undeclared identifier

Hint: If a module without routing paths is configured the validation can not determine the communication type.
 Ensure that the /MICROSAR/PduR/PduRBswModules Parameter are configured suitable to the post- build scenario.

When does this happen:

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

In which configuration does this happen:

/ActiveEcuC/PduR/PduRGeneral[PduRDevErrorDetect] == true AND
Impl. Config Variant == VARIANT-POST-BUILD-LOADABLE AND
/ActiveEcuC/PduR/IpduM exists (/MICROSAR/PduR/PduRBswModules) AND
No routing path for IpduM exists AND the /ActiveEcuC/PduR/IpduM[PduRCommunicationInterface] == false

Resolution Description:

Workaround:

The communication type must be configured for BSW modules without routing paths.

Configure /ActiveEcuC/PduR/IpduM[PduRCommunicationInterface] to "true"

Resolution:

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

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.

ESCAN00089287 Dem APIs are incompatible to the application

Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	3.00.00
Fixed in versions:	9.00.00

Problem Description:

What happens (symptoms):

Application interfaces cannot be connected to the Dem ports because of incompatible type definitions.

The Dem APIs use Dem_ExtendedStatusByteType which uses an incompatible compu-method compared to Dem_UdsStatusType.

When does this happen:

Always when trying to connect an application software to the Dem

In which configuration does this happen:

Applications using port definitions according to Autosar 4.1.2

Resolution Description:

Workaround:

Change the datatype used in the application SWC to enumeration instead of bitfield.

Resolution:

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

ESCAN00089309	Generation error PDUR10530 Handle Id parameter is missing
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	9.00.00
Fixed in versions:	9.03.00

Problem Description:

What happens (symptoms):

During code generation, the PduR generator throws error PDUR10530 and stops generation.
Additionally, errors PDUR90001 and PDUR90005 are thrown for the same problem.

When does this happen:

During code generation.

In which configuration does this happen:

All configurations including the module:
Il_AsrIpduMCfg5@root version 7.00.00 and higher
where the optional parameter
/MICROSAR/IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxConfirmationPduId
does not exist
AND
/MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/
PduRTransmissionConfirmation == FALSE
AND
/MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/
PduRDestPduDataProvision == PDUR_DIRECT

Resolution Description:

Workaround:

Set /MICROSAR/IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/
IpduMTxConfirmationPduId to any value (e.g. 0), a validator will then calculate the correct the value.

Resolution:

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

ESCAN00089580	Solving action of validation PDUR10500 has not effect
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	9.03.00

Problem Description:

What happens (symptoms):

The provided solving action of PduR validation result PDUR10500 has not effect. The validation error gets not resolved.

PDUR10500 PduRDestTxBufferRef parameter missing. (58 messages)
 PDUR10500 The parameter PduRDestTxBufferRef must be configured when PduRDestPduDataProvision(value=PDUR_TRIGGERTRANSMIT) is set to PDUR_TRIGGERTRANSMIT.
 Create the container DefinitionRef: /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRDestTxBufferRef in

When does this happen:

During execution of the provided solving action:
 Create the container DefinitionRef: /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRDestTxBufferRef in

In which configuration does this happen:

In case a validation result for PDUR10500 was found.

This happens for PDUS with the following criteria:
Communication Interface Gateway Routing Paths are configured
AND
PduRDestPduDataProvision == PDUR_TRIGGERTRANSMIT
AND
The parameter /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRDestTxBufferRef does not reference any PduRTxBuffer

Resolution Description:

Workaround:

The incorrect /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRDestTxBufferRef parameter(s) needs to be configured manually.

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.

ESCAN00090253	Communication interface PDU are misleading to a wrong destination
Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	6.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

If Communication Interface and Transport Protocol routings are configured and the Communication Interface option in the PduR BSW module is disabled the SecOC communication interface PDU is misleading to a wrong destination.

Hint: Missing validation for communication interface pathways.

When does this happen:

Always and immediately during runtime

In which configuration does this happen:

- Configuration with SecOC and
- If Communication Interface and Transport Protocol routings are configured for the SecOC and
- Communication Interface option in the PduR BSW Modules is disabled

Resolution Description:

Workaround:

Enable the communication interface option in the PduR- BSW Modules for SecOC module

Resolution:

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

ESCAN00090430 Module initialization fails

Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	5.00.00
Fixed in versions:	8.00.08, 12.00.00

Problem Description:

What happens (symptoms):

In PostBuild Loadable configurations
Initializing the Dem will result in EcuM_BswErrorHook

In PostBuild Selectable configurations (only PB-S, no PB-L)
The Dem will behave in a random way

When does this happen:

Always and immediately

In which configuration does this happen:

Postbuild Selectable and/or Postbuild Loadable configurations with module individual post-build enabled
AND
Segmented memory model, where memory mapping of DEM_PBCFG differs in pointer distance from DEM_PBCFG_ROOT and/or DEM_CONST

PB-Loadable with Module individual PB: Affected if pointers to DEM_PBCFG_ROOT cannot be assigned to a pointer to DEM_PBCFG
PB-Selectable without post-build loadable: Affected if pointers to DEM_CONST cannot be assigned to a pointer to DEM_PBCFG

Resolution Description:

Workaround:

Change the memory mapping configuration such that pointers to DEM_PBCFG_ROOT / DEM_CONST can be assigned to a pointer variable defined for DEM_PBCFG
e.g. use far pointers for both DEM_PBCFG_ROOT / DEM_CONST and DEM_PBCFG

Resolution:

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

ESCAN00090666 Linker error caused by wrong memory section name

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

Problem Description:
What happens (symptoms):

A linker error occurs due to a missing memory section. E.g. OsAppBswNonTrusted

When does this happen:

If a special memory mapping configuration in MemMap.h was used to build the library.
E.g. if memory protection is used and the BSW is mapped in the memory section of a special OS application.

In which configuration does this happen:

If the customer uses a different name for the related memory sections or OS applications.

Resolution Description:
Workaround:

Renaming the section due to the Vector Configuration

Resolution:

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

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

Problem Description:

What happens (symptoms):

The configuration tool reports Rte90005 - Generator (MICROSAR RTE Generator) failure, because of an exception

- Exception in Rte generator during Generation encountered: java.lang.IllegalArgumentException

When does this happen:

This happen during generation phase.

In which configuration does this happen:

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

Resolution Description:

Workaround:

Solving the reported RTE errors messages.

Resolution:

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

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

Problem Description:

What happens (symptoms):

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

When does this happen:

Always during the NvM_WriteAll() is performed.

In which configuration does this happen:

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

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

Resolution Description:

Workaround:

The only workaround is to filter this DET message.

Resolution:

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

ESCAN00091322	Validiation error message cannot be solved: Error at validator runtime Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModel
Component@Subcomponent:	Nm_Asr4NmIf@GenTool_GeneratorMsr
First affected version:	9.00.00
Fixed in versions:	
Problem Description:	

ESCAN00091322 Validiation error message cannot be solved: Error at validator runtime Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModel

What happens (symptoms):

The following validation error message appears in the Validation view in DaVinci Configurator:

ConsistencyRT00002 Error at validator runtime (1 message)
 ConsistencyRT00002 Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModelView
This is not a configuration problem but an internal implementation error. Please contact Vector for support.
Validator-Class:
com.vector.cfg.gen.Nm_Asr4NmIf.validation.NmGlobalCoordinatorTimeAllNmOsekInNormalValidator
Validator-Description:NmGlobalCoordinatorTimeAllNmOsekInNormalValidator validates that the setting NmGlobalCoordinatorTimeAllNmOsekInNormal is defined if it required.
Further runtime errors of this validator won't be reported in the UI.
Ex: com.vector.cfg.gen.core.moduleinterface.exceptions.ValidationFailedException: [Error]
NM01003 - A Specific Channel configuration is missing for the NmChannelConfig
- The corresponding CanNmChannelConfig is missing for this NmChannelConfig
We are sorry, but due to this internal error, code generation of /[ANY]/CanNm, /MICROSAR/NmOsek, /[ANY]/FrIf, /[ANY]/FrNm, /[ANY]/UdpNm, /[ANY]/ComM, /MICROSAR/Nm has to be blocked. As a workaround, you may try to restart DaVinci Configurator. Otherwise, please call Vector for support
 /ActiveEcuC/ComM
 FrIf
 UdpNm
 CanNm
 /ActiveEcuC/Nm
 FrNm
 /ActiveEcuC/NmOsek

Apparently, the message cannot be resolved.

When does this happen:

During configuration with DaVinci Configurator.

In which configuration does this happen:

Any configuration with Nm where a NmChannelConfig container exists that does not have a correspondent BusNmChannelConfig container (e.g. CanNmChannelConfig, FrNmChannel, LinNmChannelConfig, UdpNmChannelConfig, J1939NmChannelConfig, NmOsekChannelConfig, ...)

AND

(
('Coordinator Support Enabled' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmCoordinatorSupportEnabled) is turned OFF in the NmGlobalFeatures container in Nm in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator-> Nm_Cfg.h contains #define NM_COORDINATOR_SUPPORT_ENABLED STD_OFF)
AND/OR
('Wait Bus Sleep Extensions' (/MICROSAR/NmOsek/NmOsekGlobalConfig/NmOsekWaitBusSleepExtensions) is turned OFF or not defined or cannot be found in the

ESCAN00091322 **Validiation error message cannot be solved: Error at validator runtime Consistency: an exception was caught while executing onModelEvent() of a validator. Configuration inconsistencies couldn't be reported by this validator.ModelView:UnfilteredInvariantProjectModel**

NmOsekGlobalConfig container in NmOsek in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator -> NmOsek_Cfg.h does not contain #define NMOSEK_WAIT_BUS_SLEEP_EXTENSIONS)
AND/OR
('Synchronizing Network' (/MICROSAR/Nm/NmChannelConfig/NmSynchronizingNetwork) is turned OFF for at least one NmChannelConfig container in Nm in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator)
AND/OR
('Coord Cluster Index' (/MICROSAR/Nm/NmChannelConfig/NmCoordClusterIndex) is undefined or set to 255 for at least one NmChannelConfig container in Nm in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator)
)

Please note that this is an invalid configuration because either the NmChannelConfig container without a BusNmChannelConfig must be deleted or the corresponding BusNmChannelConfig container must be created.

Resolution Description:
Workaround:

In DaVinci Configurator:
1) Create the corresponding BusNmChannelConfig container and configure its parameters and sub-containers.
OR
2) Delete the NmChannelConfig container that lacks a corresponding BusNmChannelConfig container.

Afterwards (no matter whether 1) or 2) has been applied), save the configuration, close it and re-open it.

Resolution:

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

ESCAN00091373 RTE01069 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:	4.14.00

Problem Description:

What happens (symptoms):

RTE generation is incorrectly aborted with an RTE01069 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.

ESCAN00091493	**An unnecessary DET error is reported when using extended or mixed addressing**
Component@Subcomponent:	Tp_Asr4TpCan@Implementation
First affected version:	3.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

----

1. A DET error is reported when receiving an Rx PDU containing a "Target Address" that doesn't match any of the Rx SDUs configured with "Extended Addressing" in the CanTp.

AND/OR

2. A DET error is reported when receiving an Rx PDU containing an "Address Extension" that doesn't match any of the Rx SDUs configured with "Mixed Addressing" in the CanTp.

When does this happen:

This happens always when receiving an Rx PDU with an "unknown" address extension or target address (i.e., the ECU is not the intended addressee).

In which configuration does this happen:

This happens only in configurations where at least one Rx SDU is configured with extended addressing or mixed addressing:

(CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxAddressingFormat == CANTP_EXTENDED)

AND/OR

(CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxAddressingFormat == CANTP_MIXED)

DET error reporting must also be enabled (CANTP_DEV_ERROR_REPORT == STD_ON)

Resolution Description:

Workaround:

No error is reported if DET error reporting is disabled.

In case DET error reporting is required, the error CANTP_E_INVALID_RX_ID could be filtered out when reported from CANTP_SID_RXINDICATION.

Resolution:

This issue hasn't been solved yet.

ESCAN00091624 Using array types for unqueued IOC communication leads to compile error: array initialization needs curly braces

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

Problem Description:

What happens (symptoms):

The compiler states: array initialization needs curly braces

When does this happen:

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

In which configuration does this happen:

This happens only when array types are used for unqueued IOC communication.

Resolution Description:

Workaround:

Do not use array types for unqueued communication. If necessary put the array into a structure first.

Resolution:

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

ESCAN00091822	Error message "The number of channels and/or BusNms on one channel inside one coordination cluster must be greater than '1'." is displayed at the wrong place
Component@Subcomponent:	Nm_Asr4NmIf@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	9.00.01

Problem Description:

What happens (symptoms):

The error message
NM02041 The Coordinator settings are configured incorrectly. (1 message)
NM02041 The number of channels and/or BusNms on one channel inside one coordination cluster must be greater than '1'.
/ActiveEcuC/Nm/NmChannelConfig_001[NmCoordClusterIndex]
/ActiveEcuC/Nm/NmGlobalConfig/NmGlobalFeatures[NmCoordinatorSupportEnabled]
is displayed at a wrong place in DaVinci Configurator in the 'Network Management General' / 'Basic Editor' at some 'Coord Cluster Index' parameter that is actually configured correctly.

Example:
In a setup with three channels having each one of them a NmChannelConfig container, the Coord Cluster Index setting is set to 0 for the first and second NmChannelConfig container and set to 1 for the third NmChannelConfig container.
Then the error message might be displayed at the 'Coord Cluster Index' parameter of the first or second NmChannelConfig container.

When does this happen:

During configuration with DaVinci Configurator.

In which configuration does this happen:

In invalid configurations where
'Coordinator Support Enabled' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmCoordinatorSupportEnabled) is turned ON in the NmGlobalFeature container in the Network Management General' / 'Basic Editor' in DaVinci Configurator

AND

a NmChannelConfig container exists that has the NmCoordClusterIndex (/MICROSAR/Nm/NmChannelConfig/NmCoordClusterIndex) defined to a value that no other NmChannelConfig container contains

AND

the aforementioned NmChannelConfig container has only one NmBusType (/MICROSAR/Nm/NmChannelConfig/NmBusType) subcontainer

Please note that this is an invalid configuration and only the error message is displayed at the wrong configuration item.

Resolution Description:

ESCAN00091822 Error message "The number of channels and/or BusNms on one channel inside one coordination cluster must be greater than '1'." is displayed at the wrong place

Workaround:

Locate the NmChannelConfig container that has the only unique value for 'Coord Cluster Index'.

If the channel that the NmChannelConfig container belongs to shall be coordinated:
Either
A) make sure that another NmBusType container exists below the NmChannelConfig container
or
B) make sure that another NmChannelConfig container exists that has the same 'Coord Cluster Index' value

If the channel that the NmChannelConfig container belongs to shall not be coordinated: just delete the 'Coord Cluster Index' value

Please note that the workaround actually fixes the configuration to make it a valid one.

Resolution:

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

ESCAN00092001 Compiler error: Undefined identifier *IterType with size relations

Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	8.00.00
Fixed in versions:	8.03.01

Problem Description:
What happens (symptoms):

Compile error occurs in the doxygen group *IterableTypesWithSizeRelations. The type definition of the size relevant array *IterType is undefined.

When does this happen:

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

In which configuration does this happen:

Any configuration with a ConstStruct containing only indirection, which is deactivated by a reason.

Resolution Description:
Workaround:

No Workaround available.

Resolution:

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

ESCAN00092035 BswM generator throws an exception in case of incorrect referenced SWC Modes

Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

An exception is thrown during SWC Template Generation similar to the following one:

```
48599 ERROR - Consistency: an exception was caught while executing onModelEvent() on rule
"com.vector.cfg.gen.SysService_Asr4BswMCfg5.validation.swc.BswMSwcModeNotificationValidation"

java.lang.IllegalStateException: An PreferredSolvingAction was already assign. Two
PreferredSolvingActions are not allowed.
```

As a consequence, SWC Template Generation fails.

When does this happen:

During tool usage.

In which configuration does this happen:

Incorrect SWC Modes are referenced by /MICROSAR/BswM/BswMConfig/BswMArbitration/BswMModeRequestPort/BswMModeInitValue/BswMSwcModeInitValue/BswMSwcModeInitValueRef and other Swc Mode parameters.

Resolution Description:

Workaround:

Invalid references to the SWC modes has to be resolved manually.

Resolution:

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

ESCAN00092058 Inconsistent data types in interface DcmIf

Component@Subcomponent:	Diag_Asr4Dem@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	10.00.00

Problem Description:

What happens (symptoms):

SWC-Validation of Dcm and Dem generates inconsistent data types in interface DcmIf.
Limits of the corresponding enumeration data types do not match.

When does this happen:

At Dem/Dcm SWC template import time in DaVinci Developer.

In which configuration does this happen:

/Dem/DemGeneral/DemDcmSupport = True
within an AR 3.x RTE.

Resolution Description:

Workaround:

Import first the Dem_Swc.arxml file, then the Dcm_Swc.arxml file to override the Dem data types (imported through Dcm).

Resolution:

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

ESCAN00092064	SwcTemplate: Missing back-references to Ecuc-Parameter in Port-Interfaces
Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	7.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Tools that rely on back-references to Ecuc-Parameter such as BaseEcucGen will not be able to link the port interface to corresponding entry in DIAG-EX.

When does this happen:

Each time Dcm Swc is updated.

In which configuration does this happen:

For already existing Routine Service port interfaces that have been updated with DcmDspRoutineSystemTemplateDiagnosticRoutineRef.

Resolution Description:

Workaround:

Variant A) Delete Dcm Swc file and reopen and save Cfg5-project.
Variant B) Temporarily deactivate affected Routine (toggle DcmDspRoutineUsed to False and True).
Variant C) Use Cfg5.13 SP 3, Cfg5.14 SP1 or a newer version.

Resolution:

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

ESCAN00092245 TechRef: Integration of secret key is not correct

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

Problem Description:

What happens (symptoms):

The Technical Reference is slightly outdated.
In Chapter 4.1.1 the TechRef states out:
"SwcSecAccessFord_Cfg.c In this file the secret keys can be stored"

This is no longer correct since the implementation version 1.00.00. The key data are now fetched by using the DCM service "GetSecurityLevelFixedBytes".

When does this happen:

Implementation version > 1.00.00

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00092505 The start address for CAN Message RAM only works for 64KByte alignment.

Component@Subcomponent:	DrvCan_Mpc5700McanAsr@GenTool_GeneratorMsr
First affected version:	3.00.01
Fixed in versions:	3.03.01

Problem Description:

What happens (symptoms):

No CAN communication on CAN Bus.
Just CAN ID "0" with dlc "0" is sent by the ECU.

When does this happen:

Always and immediately

In which configuration does this happen:

It happens when the start address is not aligned to a 64KByte block.

Resolution Description:

Workaround:

Do only configure Start Addresses for the CAN Message RAM which are aligned to 64KByte boundaries.

Resolution:

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

ESCAN00092569	**Compiler error: identifier "pduInfo_var_id" is undefined**
Component@Subcomponent:	DrvCan_Mpc5700McanLl@Implementation
First affected version:	2.08.00
Fixed in versions:	2.10.00

Problem Description:

What happens (symptoms):

Compiler error: identifier "pduInfo_var_id" is undefined

When does this happen:

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

In which configuration does this happen:

Only if the MCAN Revision is less than 3.1.0 (CAN_MCAN_REVISION < 0x0310)
AND
CAN FD is activated (CAN_FD_SUPPORT != CAN_NONE).

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00092571	Compiler error: Undefined symbol is used
Component@Subcomponent:	If_AsrIfFeeSmallSector@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	1.00.01

Problem Description:

What happens (symptoms):

A compile error will occur because of using an undefined symbol: EA_PRIVATE_CONST

When does this happen:

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

In which configuration does this happen:

In every configuration.

Resolution Description:

Workaround:

Add #define EA_PRIVATE_CONST to compiler abstraction.

Resolution:

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

ESCAN00092622	A change of the main function period does not lead to a rebuild of the SWC description
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The SWC description file is not updated after a change of the EcuM main function period.

When does this happen:

After change of the parameter /MICROSAR/EcuM/EcuMGeneral/EcuMMainFunctionPeriod.

In which configuration does this happen:

In all configurations.

Resolution Description:

Workaround:

Adapt another parameter which leads to a rebuild of the SWC description, e.g. rename of a sleepmode [/EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMSleepMode].

After rebuild the name of this sleepmode can be switched back to the old name, the rename is only necessary to trigger a rebuild.

Resolution:

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

ESCAN00092623 Compiler error: fatal error: Nvm_Cfg.h: No such file or directory

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

Problem Description:

What happens (symptoms):

The compiler throws the following error:

fatal error: Nvm_Cfg.h: No such file or directory

When does this happen:

During compilation.

In which configuration does this happen:

A compiler / operating system (Linux) is used which acts case sensitive.

AND

A NvM module is part of the configuration.

Resolution Description:

Workaround:

Provide a header file Nvm_Cfg.h which includes the original file NvM_Cfg.h.

Resolution:

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

ESCAN00092644 ConsistencyRT00002 after adding multiple ComStackContributions of the same type

Component@Subcomponent:	Cdd_AsrCddCfg5@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

ConsistencyRT00002 is thrown

When does this happen:

----

1) you add an unnecessary ComStackContribution that voids the allowed multiplicity
2) you remove the unnecessary ComStackContribution by delete or undo to comply with the allowed multiplicity again
3) the above exception is thrown

In which configuration does this happen:

Any configuration containing the module Cdd_AsrCddCfg5

Resolution Description:

Workaround:

Restart CFG5 and the message is gone again.

Resolution:

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

ESCAN00092718 <MSN>90005 - Generator (<Generator Name>) failure, because of an exception "exception in <Msn> generator during Validation encountered: java.lang.NullPointerException"

Component@Subcomponent:	CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr
First affected version:	5.01.00
Fixed in versions:	5.02.00

Problem Description:

What happens (symptoms):

During code generation, an error message similar to the following one:

```
[Error] <MSN>90005 - Generator (<Generator Name>) failure, because of an exception
- Exception in <Msn> generator during Validation encountered:
java.lang.NullPointerException
 --------------------
 Erroneous CEs:
 [DefinitionRef: /MICROSAR/<Msn>]
 --------------------
 CEs:
 [DefinitionRef: /MICROSAR/<Msn>]
 --------------------
 AutoSolvingAction: <none>
 PreferredSolvingAction: <none>
 SolvingActions: <none>
 --------------------
 GeneratorPackage: <Generator Name>(<Generator version> - com.vector.cfg.gen.<module
name>)
```

When does this happen:

During code generation with DaVinci Configurator.

In which configuration does this happen:

Any

Resolution Description:

Workaround:

Use
com.vector.cfg.gen.CommonAsr_ComGenericGenLib.data.access.GenericGenAccess.getStruct(String
instead

Resolution:

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

ESCAN00092720 DataRenamer not working for MICROSAR Define block

Component@Subcomponent:	CommonAsr_ComGenericGenLib@GenTool_GeneratorMsr
First affected version:	5.01.00
Fixed in versions:	5.02.00

Problem Description:

What happens (symptoms):

The data renamer does not work for the MICROSAR define block.

When does this happen:

During code generation with DaVinci Configurator.

In which configuration does this happen:

Any

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00092756 Name of OBD calibrateable configuration symbols are out of date

Component@Subcomponent:	Diag_Asr4Dcm@Doc_TechRef
First affected version:	4.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Within chapter 9.11.1.2 "Calibration of Supported OBD Parameter Identifier" some calibratable symbols are listed with an obsolete naming convention:
For example "dcmCfg_Svc01SupportedIdMask" should be "Dcm_CfgSvc01SupportedIdMask".

When does this happen:

At ECU project setup time.

In which configuration does this happen:

- Calibration of supported OBD parameter identifier is supported

Resolution Description:

Workaround:

Read the cited global const table names as if prefixed with "DcmCfg_" instead of the printed "dcmCfg_".

Resolution:

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

ESCAN00092812 Using trusted function stubs with user defined types may lead to compile error: left operand must be an l-value

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

Problem Description:

What happens (symptoms):

The compiler states the error message: "left operand must be an l-value" (or a similar message).

When does this happen:

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

In which configuration does this happen:

This happens only in scalability classes SC3 and SC4 when trusted functions are configured with trusted function stub generation and one of the parameter types of a trusted function is an array type.

Resolution Description:

Workaround:

Use a structure instead (which may contain an array).

Resolution:

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

ESCAN00092839 Linker include files are not generated for non-AUTOSAR cores.

Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	1.08.00

Problem Description:

What happens (symptoms):

Linker reports an error.

The following sections are not defined:
- OS_NONAUTOSAR_CORE<x>_CONST
- OS_CORESTATUS_CORE<x>_VAR_NOCACHE_NOINIT

Where <x> is the core ID of the non-AUTOSAR core.

For AUTOSAR cores there are linker include files (Os_Link_Core<x>.ld) generated, which shall be used in the main linker file.
For non-AUTOSAR cores there are no such files generated.

When does this happen:

Always and immediately.

In which configuration does this happen:

There are cores which are configured as non-AUTOSAR.
This is also the case, if a multi-core hardware is used as a single core system.

Resolution Description:

ESCAN00092839 Linker include files are not generated for non-AUTOSAR cores.

Workaround:

For each non-AUTOSAR core create a linker include file (e.g. Os_Link_Core<x>.ld) with the following content:

```
--[ Os_Link_Core<x>.ld ]-----------
#if defined ( OS_LINK_CONST ) || defined ( OS_LINK_CONST_KERNEL ) || defined
( OS_LINK_CONST_KERNEL_NEAR )
.OS_NONAUTOSAR_CORE<x>_CONST ALIGN(32) :> .
_OS_NONAUTOSAR_CORE<x>_CONST_START = ADDR(.OS_NONAUTOSAR_CORE<x>_CONST);
_OS_NONAUTOSAR_CORE<x>_CONST_END = ENDADDR(.OS_NONAUTOSAR_CORE<x>_CONST);
#endif

#if defined ( OS_LINK_VAR ) || defined ( OS_LINK_VAR_KERNEL ) || defined
( OS_LINK_VAR_KERNEL_FAR ) || defined ( OS_LINK_VAR_KERNEL_FAR_NOCACHE ) || defined
( OS_LINK_VAR_KERNEL_FAR_NOCACHE_NOINIT )
.OS_CORESTATUS_CORE<x>_VAR_NOCACHE_NOINIT ALIGN(32) :> .
_OS_CORESTATUS_CORE<x>_VAR_NOCACHE_NOINIT_START =
ADDR(.OS_CORESTATUS_CORE<x>_VAR_NOCACHE_NOINIT);
_OS_CORESTATUS_CORE<x>_VAR_NOCACHE_NOINIT_END =
ENDADDR(.OS_CORESTATUS_CORE<x>_VAR_NOCACHE_NOINIT);
#endif

#ifdef OS_LINK_CONST
# undef OS_LINK_CONST
#endif

#ifdef OS_LINK_VAR
# undef OS_LINK_VAR
#endif
-------------
```

Use the appropriate syntax for your compiler. For correct syntax you can look at the generated files of AUTOSAR cores.
Use the manually created linker include file(s) in the main linker file.

Resolution:

The described issue is corrected by modification of the generator.

ESCAN00092892 Compiler error: function "EcuM_BswErrorHook" has no prototype

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

Problem Description:

What happens (symptoms):

Compiler throws the following error:

function "EcuM_BswErrorHook" has no prototype

When does this happen:

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

In which configuration does this happen:

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

AND

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

Resolution Description:

Workaround:

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

Resolution:

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

ESCAN00092955	**Compiler error: incompatible types - from 'const <MSN>_PCConfigType *' to 'const <MSN>ConfigType *const**
Component@Subcomponent:	SysService_Asr4EcuM@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	
Problem Description:	

ESCAN00092955 Compiler error: incompatible types - from 'const <MSN>_PCConfigType *' to 'const <MSN>ConfigType *const

What happens (symptoms):

The compiler throws an error like the following:

```
1> EcuM_Init_Cfg.c
1>GenData/EcuM_Init_Cfg.c(86): error C4133: 'initializing' : incompatible types - from 'const CanNm_PCConfigType *' to 'const EcuM_PbConfigType *const '
1>GenData/EcuM_Init_Cfg.c(87): error C4133: 'initializing' : incompatible types - from 'const EcuM_PCConfigType *' to 'const SchM_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(88): error C4133: 'initializing' : incompatible types - from 'const SchM_ConfigType *' to 'const Can_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(89): error C4133: 'initializing' : incompatible types - from 'const Can_PCConfigType *' to 'const CanIf_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(90): error C4133: 'initializing' : incompatible types - from 'const CanIf_PCConfigType *' to 'const Com_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(91): error C4133: 'initializing' : incompatible types - from 'const Com_PCConfigType *' to 'const PduR_PBConfigType *const '
1>GenData/EcuM_Init_Cfg.c(92): error C4133: 'initializing' : incompatible types - from 'const PduR_PCConfigType *' to 'const CanSM_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(93): error C4133: 'initializing' : incompatible types - from 'const CanSM_PCConfigType *' to 'const CanNm_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(103): error C4133: 'initializing' : incompatible types - from 'const CanNm_PCConfigType *' to 'const EcuM_PbConfigType *const '
1>GenData/EcuM_Init_Cfg.c(104): error C4133: 'initializing' : incompatible types - from 'const EcuM_PCConfigType *' to 'const SchM_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(105): error C4133: 'initializing' : incompatible types - from 'const SchM_ConfigType *' to 'const Can_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(106): error C4133: 'initializing' : incompatible types - from 'const Can_PCConfigType *' to 'const CanIf_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(107): error C4133: 'initializing' : incompatible types - from 'const CanIf_PCConfigType *' to 'const Com_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(108): error C4133: 'initializing' : incompatible types - from 'const Com_PCConfigType *' to 'const PduR_PBConfigType *const '
1>GenData/EcuM_Init_Cfg.c(109): error C4133: 'initializing' : incompatible types - from 'const PduR_PCConfigType *' to 'const CanSM_ConfigType *const '
1>GenData/EcuM_Init_Cfg.c(110): error C4133: 'initializing' : incompatible types - from 'const CanSM_PCConfigType *' to 'const CanNm_ConfigType *const '
```

When does this happen:

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

In which configuration does this happen:

In variant configurations with modules which uses different EcuC init phases in different variants (/MICROSAR/EcuC/EcucGeneral/BswInitialization/InitFunction/InitPhase).

E.g.
VARIANT_1: InitPhase = NO_INIT
VARIANT_2: InitPhase = INIT_TWO_MCAL

Resolution Description:

ESCAN00092955 Compiler error: incompatible types - from 'const <MSN>_PCConfigType *' to 'const <MSN>ConfigType *const

Workaround:

To resolve this the content of theCONT EcuM_GlobalConfigRoot in EcuM_Init_Cfg.c has to be reordered to fit to the struct EcuM_GlobalConfigRootType.

e.g.

```
CONST(EcuM_GlobalConfigRootType, ECUM_CONST) EcuM_GlobalConfigRoot =
{
 {
 BswM_Config_CanNm_Ptr,
 EcuM_Config_CanNm_Ptr,
 CanNm_Config_CanNm_Ptr,
 },
 {
 BswM_Config_ClassB_Ptr,
 CanNm_Config_ClassB_Ptr, <====== Wrong position, must be the last one
 EcuM_Config_ClassB_Ptr,
 },
 {
 BswM_Config_ClassC_Ptr,
 CanNm_Config_ClassC_Ptr, <====== Wrong position, must be the last one
 EcuM_Config_ClassC_Ptr,
 }
};

typedef struct
{
 CONSTP2CONST(BswM_ConfigType, TYPEDEF, BSWM_INIT_DATA) CfgPtr_BswM_Init;
 CONSTP2CONST(EcuM_PbConfigType, TYPEDEF, ECUM_INIT_DATA) CfgPtr_EcuM_Init;
 CONSTP2CONST(CanNm_ConfigType, TYPEDEF, CANNM_INIT_DATA) CfgPtr_CanNm_Init;
} EcuM_GlobalPCConfigType;
```

Resolution:

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

ESCAN00092986 Compiler error: Error directive due to inconsistent configuration of Communication Mode Info

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

Problem Description:

What happens (symptoms):

The following error is shown by the compiler:
../../../external/bsw/xcp/Xcp.h:2442: #error "XCP consistency error: Communication mode info should be enabled when using block download."

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
- Xcp Block Download (/MICROSAR/Xcp/XcpGeneral/XcpBlockDownload) is enabled
AND
- Xcp Communication Mode Info (/MICROSAR/Xcp/XcpGeneral/XcpCommunicationModeInfo) is disabled.

Resolution Description:

Workaround:

Enable Xcp Communication Mode Info (/MICROSAR/Xcp/XcpGeneral/XcpCommunicationModeInfo)

Resolution:

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

ESCAN00092995	**CAN-FD message without BRS will not be received**
Component@Subcomponent:	DrvCan_Mpc5700McanLl@Implementation
First affected version:	2.09.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

A CAN-FD message without Bitrate Switching will:

- not be received by the upper layers.
- produce a DET Error (CAN_E_PARAM_DLC) for messages with a DLC greater than 8 bytes.

When does this happen:

During runtime,
always and immediately,
when a CAN-FD message is received without Bitrate Switching (CAN-FD BRS bit) set.

In which configuration does this happen:

Only in configurations using CAN-FD Rx messages without Bitrate switching.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00093023 RTE Analyzer sporadically triggers windows application crash dialog

Component@Subcomponent:	Rte_Analyzer@Application
First affected version:	0.07.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Windows issues a message box
MICROSAR RTE Analyzer (Beta) has stopped working.

If started on the console, the console might contain additional message "free from wrong pool during global destruction"

When does this happen:

This happens after RTE Analyzer terminated itself.

In which configuration does this happen:

This happens independent of the used configuration.

Resolution Description:

Workaround:

Set the environment variable KOMODO_VERSION to 1 before starting MicrosarRteAnalyzer
Example:

```
set KOMODO_VERSION=1
MicrosarRteAnalyzer -c config.json -o outputDirectory
```

Resolution:

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

ESCAN00093048	**RTE generator generates unused Rte_TrustedCom_SendSignal**
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.06.00
Fixed in versions:	1.14.00, 1.13.01

Problem Description:

What happens (symptoms):

RTE generator generates an API Rte_TrustedCom_SendSignal that is never called.

When does this happen:

During generation.

In which configuration does this happen:

This happens when memory protection is used and when the configuration contains sender-receiver communication
with receiver components that do not have port accesses for the received data elements.

Resolution Description:

Workaround:

Configure read accesses in the receiver component.

Resolution:

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

ESCAN00093079 *_GLOBALSHARED_VAR_NOCACHE_* defines are used by Rte_MemMap.h but not generated by Os_MemMap.h.

Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	1.01.05
Fixed in versions:	1.08.00

Problem Description:

What happens (symptoms):

Sources do not compile, the MemMap defines *_GLOBALSHARED_VAR_NOCACHE_* are used by Rte_MemMap.h, but they are not provided by the Os_MemMap.h.

When does this happen:

During compile time.
Always and immediately

In which configuration does this happen:

In every configuration where explicit linkage is used.

Resolution Description:

Workaround:

Use the OS generated define instead.

Resolution:

Provide compatibility define.

ESCAN00093109	**Missing Limitation for aggregated measurement data size of < 64KB**
Component@Subcomponent:	Cp_AsrXcp@Doc_TechRef
First affected version:	2.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The Xcp has a limitation that limits the aggregated data size during DAQ measurement to < 64KB.
This is currently not documented and mostly relevant for Linux systems.

When does this happen:

Always and immediately

In which configuration does this happen:

----

all configurations

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00093110 Service 0x2F: ECU sends NRC 0x10 to a valid IO control DID request

Component@Subcomponent:	Diag_Asr4Dcm@GenTool_GeneratorMsr
First affected version:	7.01.00
Fixed in versions:	7.02.00

Problem Description:

What happens (symptoms):

The ECU sends NRC 0x10 (GeneralReject) on a validly formatted service 0x2F (IoControlByIdentifier).

When does this happen:

- At runtime when a valid request to control an IO-DID is sent to the ECU
AND
- The requested IO-DID uses S/R (Sender/Receiver) communication
AND
- The application needs more time to prepare the final result of the requested operation (i.e. the IOControlResponse data of the corresponding S/R port is written at a later time)

In which configuration does this happen:

Any configuration that specifies all IO-DIDs to use S/R communication.

Resolution Description:

Workaround:

Define at least one IO-DID to use application function call interface with asynchronous data access i.e. parameter /Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort == USE_DATA_ASYNCH_CLIENT_SERVER or USE_DATA_ASYNCH_FNC

Resolution:

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

ESCAN00093127	RTE generator does not create IOCs in the OS configuration for single core systems
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.13.01, 1.14.00

Problem Description:

What happens (symptoms):

The RTE uses IOCs from the OS but the generator does not add the IOCs to the OS configuration.

When does this happen:

During generation.

In which configuration does this happen:

This happens when memory protection is used and when there are N:1 client-server or sender-receiver communication with multiple sending OS applications.

Resolution Description:

Workaround:

Manually configure the IOCs in the OS configuration.

Resolution:

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

ESCAN00093144 Class cast exception when the configuration contains a subelement mapping for application datatypes

Component@Subcomponent:	Rte_Asr4@GenTool_GeneratorMsr
First affected version:	4.09.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

```
RTE generator reports an error message Rte90005
Exception in Rte generator during Calculation encountered:
java.lang.ClassCastException:
com.vector.cfg.model.mdf.impl.ar4x.swcomponenttemplate.portinterface.VMApplicationComposition
cannot be cast to
com.vector.cfg.model.mdf.ar4x.swcmponenttemplate.portinterface.MIIMplementationDataTypeSubE
```

When does this happen:

During generation.

In which configuration does this happen:

This happens when the configuration contains a connection between a delegation port with application record type
and a component port with a primitive application datatype.
Moreover a subelement mapping needs to be configured for the connection.

Resolution Description:

Workaround:

Either use implementation datatypes for the subelement mapping
or
create a wrapper SWC that converts the data from the record type to a primitive type
or
receive the full record in the application SWC.

Resolution:

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

ESCAN00093236 Compiler error: OS does not compile if only application specific hooks are configured.

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

Problem Description:

What happens (symptoms):

There are compilation error if only application specific hooks (no global hooks) are configured.

```
ctc E272: ["GenData/Os_Hal_Context_Lcfg.c" 274/38] undeclared identifier
"OsCfg_Stack_OsCore0_Startup_Dyn"
1> ctc E245: ["GenData/Os_Hal_Context_Lcfg.c" 274/38] invalid type for left operand of subscript
1> ctc E244: ["GenData/Os_Hal_Context_Lcfg.c" 274/38] invalid operand types for + operator
1> ctc E245: ["GenData/Os_Hal_Context_Lcfg.c" 275/37] invalid type for left operand of subscript
1> ctc E244: ["GenData/Os_Hal_Context_Lcfg.c" 275/37] invalid operand types for + operator
1> ctc E306: ["GenData/Os_Hal_Context_Lcfg.c" 274/29] initializer must be constant
1> ctc E306: ["GenData/Os_Hal_Context_Lcfg.c" 275/29] initializer must be constant
```

When does this happen:

If only application specific hooks and no system hook is configured.

In which configuration does this happen:

SC3/SC4

Resolution Description:

Workaround:

Configure dummy system hooks.

Resolution:

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

ESCAN00093294	Invalid key accepted due to inconsistent Csm and CryFord job processing configuration
Component@Subcomponent:	Diag_AsrSwcSecAccess_Ford@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

SecurityAccess_FunctionFinish() declares key as valid even if the verification has failed.

When does this happen:

While processing the security access key.

In which configuration does this happen:

If configured Csm job processing type and CryFord job processing type are inconsistent.
E.g. Csm configured for sync job processing and CryFord for async processing.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00093309 RTE Analyzer fails due to duplicated mainfunction

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.13.00
Fixed in versions:	1.13.01, 1.14.00

Problem Description:

What happens (symptoms):

RTE Analyzer aborts with an error message.

```
Compiling...
Linking...
ERROR: Linking globals named 'WdgM_MainFunction': symbol multiply defined!
```

When does this happen:

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

In which configuration does this happen:

This happens when the configuration contains service components and BSW modules with different names and
the same exectuable entity names.

Resolution Description:

Workaround:

Duplicate the RteAnalyzerConfiguration.json and remove the source file that contains the duplicated
mainfunction from the SourceFileList (WdgM.c). Keep the source file of the service SWC (WdgM_OsApplicationTrustedC0).

Resolution:

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

ESCAN00093317 Value Calculations does not act as expected

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

Problem Description:

What happens (symptoms):

An Value Calculation has only influence to values which are added to the ComStackLib. The values of the Define Creator are directly taken from the corresponding EcuC parameter.
Furthermore, Value Calculations for references (i.e convert a reference to an integer of the target) do not work either.

When does this happen:

Always during the generation

In which configuration does this happen:

Generators which uses ValueCalculators for parameters which are created by the Microsar Define creator.
or
Value calculation is used for a reference

Resolution Description:

Workaround:

Use a struct extender / custom #define for the DefRef and blacklist the DefRef instead

Resolution:

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

ESCAN00093405 Auto Configuration - Invalid multiplicity after manual adaptations of container BswMAvailableActions

Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	10.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

User-modifications about a changed BswMAvailableActions subcontainer are recognized by the Auto Configuration assistant but even if they should be kept, the assistant will re-create the original action. This leads to an invalid model because the user modification is not removed by the assistant.

Example:
- Configure Communication Control is used and Reinitialize TX is turned ON, Finish is clicked.
- the /MICROSAR/BswM/BswMConfig/BswMModeControl/BswMAction CC_EnableDM_<I-PDU-Group> has a BswMDeadlineMonitoringControl container which is deleted within the Basic Editor
- Instead another BswMAvailableActions subcontainer is created of another type, e.g. BswMComMModeLimitation
- Configure Communication Control is used once again and Finish is clicked. An option if offered to either keep this modification or to restore it, but independent of the choice, the original BswMDeadlineMonitoringControl is restored without removing the user modification. Because the user modification is not removed the multiplicity of the container BswMAvailableActions[0...1] is violated.

When does this happen:

During the configuration with DaVinci Configurator in the BSW Management Editor in the following sequence:
- Configure <Auto Configuration> is clicked
- Finish is clicked
- Some objects like a /MICROSAR/BswM/BswMConfig/BswMModeControl/BswMAction/BswMAvailableActions/BswMDeadlineMonitoringControl container are deleted or changed
- Configure <Auto Configuration> is clicked once again
- Finish is clicked
- the dialog 'Manual Adaptions' does pop up
- Finish is clicked in the 'Manual Adaptions' dialog

In which configuration does this happen:

Any configuration using one of the Auto Configurations in BSW Management in DaVinci Configurator

Resolution Description:

Workaround:

Redo the previously manual changes that have been overwritten.

Resolution:

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

ESCAN00093413	Auto Configuration Module Initialization - Changed User Include Files always restores
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	2.00.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

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

When does this happen:

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

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

In which configuration does this happen:

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

AND

EcuM is configured as Postbuild Loadable or Postbuild Selectable

Resolution Description:

Workaround:

Redo the previously manual changes that have been overwritten.

Resolution:

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

ESCAN00093449 A2L compu method RTE_CM_BOOLEAN cannot be used to calibrate boolean values

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

Problem Description:

What happens (symptoms):

Only FALSE can be selected in calibration tools for boolean calibration values

When does this happen:

During calibration

In which configuration does this happen:

If calibration is configured for a boolean element

Resolution Description:

Workaround:

Configure an integer element with a dedicated compu method instead of a boolean element.

Resolution:

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

ESCAN00093455 Compile error in PduR: <CDD>_Transmit is not defined

Component@Subcomponent:	Cdd_AsrCddCfg5@GenTool_GeneratorMsr
First affected version:	2.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compile error in PduR: <CDD>_Transmit is not defined.

This mandatory API is not defined by CDD

When does this happen:

----

compile time

In which configuration does this happen:

A CddPduRLowerLayerRxPdu but no CddPduRLowerLayerTxPdu is configured.

Resolution Description:

Workaround:

Define the missing API in memmap.h, the section for the define <CDD>_START_SEC_CODE

Resolution:

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

ESCAN00093463 Wrong generated MemMap.h file for Gen7 Os

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.13.00
Fixed in versions:	1.13.01, 1.14.00

Problem Description:

What happens (symptoms):

The compiler resports #error "MemMap.h, wrong pragma command"

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 with Gen7 OS.

Resolution Description:

Workaround:

Change the template content of the generated <Swc>_MemMap.h files.
The changes will not be overwritten by a later generation step

Resolution:

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

ESCAN00093469 Compiler error: 'retVal' undeclared identifier

Component@Subcomponent:	Ccl_Asr4ComMCfg5@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	8.01.00

Problem Description:

What happens (symptoms):

Functions ComM_RequestBusSMMode and ComM_GetCurrentBusSMMode in file ComM_Lcfg.c use an undeclared variable retVal.
Typical compiler error explanations may be:
..\comm_lcfg.c: error C2065: 'retVal' : 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:

----

1) Only one ComM channel exists and the channel has ComM/ComMConfigSet/ComMChannel/ComMBusType == COMM_BUS_TYPE_INTERNAL
Note, this is an unusual use-case.
AND
2) ComM/ComMGeneral/ComMDevErrorDetect == true

Resolution Description:

Workaround:

Set ComM/ComMGeneral/ComMDevErrorDetect to false if possible, otherwise no workaround available.

Resolution:

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

ESCAN00093491 Compiler error: identifier PduR_GetBmTxBufferIndRomStartIdxOfRmGDestRon is undefined

Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	11.00.00
Fixed in versions:	11.01.00

Problem Description:

What happens (symptoms):

---

Missing validation of PduRDestPduQueueDepth in case of communication interface routing

If a communication interface gateway routing path with a dedicated buffer reference is configured and the "PduRDestPduQueueDepth" parameter does not exist, no validation error is reported.

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:

---

- Communication Interface gateway routings with TxBuffer References and the /MICROSAR/PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRDestPduQueueDepth is not configured.

Resolution Description:

Workaround:

---

Create/Set the corresponding PduRDestPduQueueDepth parameter manually.

Resolution:

---

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

ESCAN00093494	Compiler error: OS event not created for inactive/ unconnected server runnables
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.14.00

Problem Description:

What happens (symptoms):

Compilation fails because an RTE TASK contains code that checks for an OS event

```
 if ((ev & Rte_Ev_Run_XXXX) != (EventMaskType)0)
 {
 }
```

The OS event is not created by the RTE generator.

When does this happen:

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

In which configuration does this happen:

This happens when a mapped server runnable uses implicit APIs (e.g. has access to an IRV with buffered communication)
and the corresponding server port is not connected or the server runnable is not active.

Resolution Description:

Workaround:

Manually create the OS event in the OS configuration.

Resolution:

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

ESCAN00093502 Technical Reference: Wrong API description for Csm_SymKeyExtractStart

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

Problem Description:

What happens (symptoms):

On TechnicalReference_Csm, Csm_SymKeyExtractStart prototype is

```
 Csm_ReturnType Csm_SymKeyExtractStart (Csm_ConfigIdType cfgId, const Csm_SymKeyType *keyPtr)
```

But has to be

```
 FUNC( Csm_ReturnType, CSM_CODE ) Csm_SymKeyExtractStart( Csm_ConfigIdType cfgId )
```

There is no second parameter.

When does this happen:

-

In which configuration does this happen:

-

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00093539	**RTE49999 when SWC template generation tries to delete write protected files**
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.00.00
Fixed in versions:	1.14.00, 1.13.01

Problem Description:

What happens (symptoms):

SWC template generation aborts with an error message
RTE49999 An unexpected error occured.

When does this happen:

This happens when SWC templates are generated and the templates are write protected.

In which configuration does this happen:

Always

Resolution Description:

Workaround:

Remove the write protection from the SWC template files.

Resolution:

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

ESCAN00093569 The IOC const WriteOutParameters causes error in build

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.02.00
Fixed in versions:	1.13.01, 1.14.00

Problem Description:

What happens (symptoms):

The IOC const WriteOutParameters causes error in build.
Error: argument of type "<DataTypeName> *" is incompatible with parameter of type "const <DataTypeName> *"

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 Client/Server-Communication over processors cores with EnforceIOC enabled.
OR
In configurations with Client/Server-Communication with more than one Client on different OsApplications.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00093579 Incorrect usage of Det

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

Problem Description:

What happens (symptoms):

The component WdgM does not use the Det module as default. Instead of using Det a module Appl_Det module must be integrated.

Furthermore the second parameter of API (Appl_)Det_ReportError uses vendor id instead of instance id according to AUTOSAR.

When does this happen:

Always if Det (Appl_Det) is enabled and an error is reported

In which configuration does this happen:

If the following parameter is enabled and an runtime / Det error is detected:
- /MICROSAR/WdgM/WdgMGeneral/WdgMDevErrorDetect == true

Resolution Description:

Workaround:

Include the Det.h in customer / user file Appl_Det.h

Redefine the API Appl_Det_ReportError to Det_ReportError, do not use the second parameter and define the second parameter to 0.
Example:
#include "Det.h"

#define Det_Appl_Det_ReportError(ModuleId, VendorIf, ApiId, ErrorId)
DetStub_ReportError(ModuleId, InstanceID, ApiId, ErrorId) Det_ReportError(ModuleId, 0, ApiId, ErrorId)

Resolution:

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

ESCAN00093580 Incorrect usage of Det

Component@Subcomponent:	If_Asr4IfWd@Implementation
First affected version:	5.00.00
Fixed in versions:	5.02.00

Problem Description:

What happens (symptoms):

The component WdgIf does not use the Det module as default. Instead of using Det a module Appl_Det module must be integrated.

Furthermore the second parameter of API (Appl_)Det_ReportError uses vendor id instead of instance id according to AUTOSAR.

When does this happen:

Always if Det (Appl_Det) is enabled and an error is reported

In which configuration does this happen:

If the following parameter is enabled and an runtime / Det error is detected:
- /MICROSAR/WdgIf/WdgIfGeneral/WdgIfDevErrorDetect == true

Resolution Description:

Workaround:

Include the Det.h in customer / user file Appl_Det.h

Redefine the API Appl_Det_ReportError to Det_ReportError, do not use the second parameter and define the second parameter to 0.
Example:
#include "Det.h"

#define Det_Appl_Det_ReportError(ModuleId, VendorIf, ApiId, ErrorId)
DetStub_ReportError(ModuleId, InstanceID, ApiId, ErrorId) Det_ReportError(ModuleId, 0, ApiId, ErrorId)

Resolution:

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

ESCAN00093633	Compiler error: Mismatch between expected and generated trusted function signature
Component@Subcomponent:	Os_CoreGen7@GenTool_GeneratorMsr
First affected version:	1.00.00
Fixed in versions:	1.08.01

Problem Description:

What happens (symptoms):

A compiler error occurs if the user configures a trusted function 'IN' parameter with datatype like "const uint32*".

An error message like the following is shown:
"argument of type "uint32 *" is incompatible with parameter of type "uint32"".

When does this happen:

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

In which configuration does this happen:

If any /MICROSAR/Os/OsApplication/OsApplicationTrustedFunction/OsTrustedFunctionParam exists with
/MICROSAR/Os/OsApplication/OsApplicationTrustedFunction/OsTrustedFunctionParam/OsTrustedFunctionParamDirection set to 'IN' and
/MICROSAR/Os/OsApplication/OsApplicationTrustedFunction/OsTrustedFunctionParam/OsTrustedFunctionParamDataType with "const <type>*".

Resolution Description:

Workaround:

The issue can be circumvented by adding the "const" keyword for a second time.
E.g. instead of "const uint32 *", configure OsTrustedFunctionParamDataType with "const const uint32 *".

Resolution:

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

ESCAN00093634 CAN-FD format (Bosch V1.0, ISO-11898) inconsistent

Component@Subcomponent:	DrvCan_Mpc5700McanAsr@GenTool_GeneratorMsr
First affected version:	3.00.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

After changing the Mcan Revision to M_CAN_REV_315 the parameter CanFd NISO is grey'ed and cannot be edited.
This is not correct for M_CAN_REV_315 as for this revision both CAN-FD formats are available.

When does this happen:

During configuration time.

In which configuration does this happen:

In every configuration.

Resolution Description:

Workaround:

Mark CanFd NISO parameter as "Set user defined",
configure the parameter as required,
mark CanFd NISO parameter as "Set NOT user defined".

Resolution:

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

ESCAN00093638	Linking fails if the configuration contains many software components
Component@Subcomponent:	Rte_Analyzer@Application
First affected version:	0.05.00
Fixed in versions:	0.09.00

Problem Description:
What happens (symptoms):

RTE Analyzer aborts with a linking failed error message because
the commandline is too long.

When does this happen:

During analysis with RTE Analyzer.

In which configuration does this happen:

When the configuration contains many software components.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00093650 Generation of RTE Analyzer stubs fails due to invalid characters in object descriptions

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.09.00
Fixed in versions:	1.14.00, 1.13.01

Problem Description:

What happens (symptoms):

RTE generator fails to generate the RTE Analyzer stubs.

When does this happen:

During generation.

In which configuration does this happen:

This happens when the description for configuration objects contain special characters like ", $, %, @;

Resolution Description:

Workaround:

Do not use the special characters ", @, %, $ in descriptions.

Resolution:

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

ESCAN00093653 Compiler error: CheckDetErrorContinue function call uses optimized away trigger disabling flag

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.13.00
Fixed in versions:	1.13.01, 1.14.00

Problem Description:

What happens (symptoms):

The compiler reports undeclared identifier "Rte_TriggerDisableMaxCount_<Name>"

When does this happen:

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

In which configuration does this happen:

In configurations with ModeDisablings and active DET.

Resolution Description:

Workaround:

No workaround available.

ESCAN00093654 Compiler error: Redefinition of Sender/Receiver data element declaration

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.07.00
Fixed in versions:	1.13.01, 1.14.00

Problem Description:

What happens (symptoms):

The compiler reports redefinition of "Read_<PortName>_<DataElementName>" or "Receive_<PortName>_<DataElementName>" variable.

When does this happen:

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

In which configuration does this happen:

This happens in configurations with Sender/Receiver communication where the PortName and DataElementName is used multiple times.

Resolution Description:

Workaround:

Rename the PortName of the ports involved in the Sender/Receiver communication.

Resolution:

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

ESCAN00093656 Compiler error: Redefinition of enumeration data type

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.10.00
Fixed in versions:	1.13.01, 1.14.00

Problem Description:
What happens (symptoms):

The compiler reports redefinition of "<VariableName>_V_<Count>" variable.

When does this happen:

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

In which configuration does this happen:

This happens in configurations where the enumeration data type is used in different software components without LiteralPrefix.

Resolution Description:
Workaround:

No workaround available.

Resolution:

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

ESCAN00093669 Compiler error: Identifier OSError_GetScheduleTableStatus_ScheduleStatus not defined.

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

Problem Description:

What happens (symptoms):

The compiler states an error message like: Identifier OSError_GetScheduleTableStatus_ScheduleStatus not defined.

When does this happen:

The error is issued by the compiler during compilation of the code in case the API function OSError_GetScheduleTableStatus_ScheduleStatus shall be used.

In which configuration does this happen:

This happens only in configurations where schedule tables are used.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00093692	Auto Configuration dialog shows an empty message about "Manual Adaptations" after finishing the Auto Configuration.
Component@Subcomponent:	SysService_Asr4BswMCfg5@GenTool_GeneratorMsr
First affected version:	10.00.00
Fixed in versions:	11.00.00

Problem Description:

What happens (symptoms):

The Auto Configuration dialog shows an empty message about "Manual Adaptations" after finishing the Auto Configuration.

When does this happen:

After finishing the Auto Configuration dialog and one feature (e.g. one specific channel in Auto Configuration Communication Control) is unselected.

In which configuration does this happen:

In all configurations which use an Auto Configuration.

Resolution Description:

Workaround:

The manual change, e.g. switch from RefValueA to RefValueB has to be reverted manually.

Resolution:

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

ESCAN00093765	Compiler error: Rte_ComSendSignalProxyPeriodic accesses missing variable
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.03.00
Fixed in versions:	1.14.00, 1.13.01

Problem Description:

What happens (symptoms):

Compilation fails because Rte_ComSendSignalProxyPeriodic accesses a variable that is not part of Rte.c

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 senders that are not mapped to the partition that contains the BSW and when
the configuration contains no internal sender-receiver connections.

Resolution Description:

Workaround:

Create an internal sender-receiver connection.

Resolution:

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

ESCAN00093785	Mirror component Transmission pathways are not detected correctly
Component@Subcomponent:	Gw_AsrPduRCfg5@GenTool_GeneratorMsr
First affected version:	10.01.00
Fixed in versions:	11.01.00

Problem Description:

What happens (symptoms):

A Tx forwarding where the upper layer is a "/MICROSAR/Mirror" are not detected correctly and a routing is not created by a solving action. The validation reports an error that a routing path is incomplete. See workaround.

When does this happen:

Always and immediately during configuration

In which configuration does this happen:

Configuration where a "/MICROSAR/Mirror" upper layer is involved in a routing

Resolution Description:

Workaround:

Configure the routing path manually.

Refer the appropriate Pdu as source and destination Pdu to the routing.
Set Dest Pdu direction to Transmit and the Data Provision to PDUR_DIRECT and the DestPduRoutingType to API_FORWARDING. This parameter must be defined via "user define"

see also TechnicalReference_Mirror.pdf Version 1.0.0 chapter "Configure the destination channel"

Resolution:

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

ESCAN00093839 CFG5 Exception in <MSN> generator during Generation encountered and no files are generated

Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	4.00.00
Fixed in versions:	8.05.01, 8.06.00

Problem Description:

What happens (symptoms):

CFG5 shows the following error message
"Exception in <MSN> generator during Generation encountered"
and no files are generated.

The detailed error description is: java.lang.NullPointerException

When does this happen:
----
at generation time.

In which configuration does this happen:

Any configuration where the postbuild-selectable support is enabled for this module
AND
the generator uses the API setRequiresIndexUsedArray() with the parameter true.

Resolution Description:

Workaround:

Deactivate ComStackLib based optimizations if configurable and try to generate again.

Resolution:

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

ESCAN00093969	Missing Dem_[Dcm]ClearDTC in list of required DEM API
Component@Subcomponent:	Diag_Asr4Dcm@Doc_TechRef
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Missing information that Dem_[Dcm]ClearDTC is also a required API.

When does this happen:

Reading technical reference.

In which configuration does this happen:

N/A

Resolution Description:

Workaround:

Consider Dem_[Dcm]ClearDTC() to be a required API by DCM.

Resolution:

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

ESCAN00094010 Compiler error: Redefinition of data handle entry in component data structure for PR Ports using implicit communication

Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.05.00
Fixed in versions:	1.13.01

Problem Description:

What happens (symptoms):

A compiler error occurs due to a redefinition of a data handle entry in the component data structure for PR Ports using implicit communication

Typical compiler error explanations may be:
'struct' member redefinition

When does this happen:

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

In which configuration does this happen:

This happens for configurations using
- PR Ports
- implicit read AND implicit write access
- object code components OR multiple instantiated SWC leading to the usage of the component data structure

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00094026	Compiler error: Missing extern declaration for Rte_InitState
Component@Subcomponent:	Rte_Core@Implementation
First affected version:	1.07.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compilation fails because Rte_InitState has no extern declaration.

When does this happen:

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

In which configuration does this happen:

This happens when a task contains schedulable entities and background triggered runnables.

Resolution Description:

Workaround:

Set the parameter
/MICROSAR/Rte/RteGeneration/RteDevErrorDetectUninit
to true.

Resolution:

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

2.6 Compiler Warnings

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

Index

ESCAN00051574	[MSR4 only] Compiler warning: statement is unreachable SysService_AsrDet@Implementation
ESCAN00055307	Compiler warning: unused static function "XcpMemClr" Cp_AsrXcp@Implementation
ESCAN00065890	Compiler warning: cast discards '__attribute__((noreturn))' qualifier from pointer target type DrvCan_Mpc5700McanLl@Implementation
ESCAN00065891	Compiler warning: cast increases required alignment of target type DrvCan_Mpc5700McanLl@Implementation
ESCAN00067159	Compiler warning: cast truncates constant value MemService_AsrNvM@Implementation
ESCAN00067161	Compiler warning: conditional expression is constant Il_AsrComCfg5@Implementation
ESCAN00067237	Compiler warning: large type was implicitly cast to small type If_AsrIfCan@Implementation
ESCAN00068402	Compiler warning: conditional expression or part of it is always true/false DrvTrans_Tja1043CandioAsr@Implementation
ESCAN00068434	Compiler warning: conditional expression or part of it is always true/false DrvCan__coreAsr@Implementation
ESCAN00068435	Compiler warning: narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned char MemService_AsrNvM@Implementation
ESCAN00068872	Compiler warning: the order of volatile accesses is undefined in this statement DrvCan__coreAsr@Implementation
ESCAN00074793	Compiler warning: Condition is always constant Diag_Asr4Dem@Implementation
ESCAN00076827	Compiler warning: Prototypes for Sector Info not present Cp_AsrXcp@GenTool_GeneratorMsr
ESCAN00078543	Compiler warning: statement is unreachable when Calibration is deactivated Cp_AsrXcp@Implementation
ESCAN00081459	Compiler warning: function "ApplCanTimerLoop" was declared but never referenced DrvCan__coreAsr@Implementation
ESCAN00086596	Compiler warning: used wrong Memory class qualifier in several functions Gw_AsrPduRCfg5@Implementation
ESCAN00086650	Compiler warning: pointless comparison of unsigned integer with zero SysService_AsrCsm@Implementation
ESCAN00087235	Compiler warning: variable "lErrorId" was declared but never referenced Tp_Asr4TpCan@Implementation
ESCAN00087501	Compiler warning: "signed/unsigned mismatch" due to missing cast in 0:N or 1:N indirections CommonAsr_ComStackLib@GenTool_GeneratorMsr
ESCAN00087536	Compiler warning: 'function' : conversion from 'const <SomeType>' to '<AnotherType>', possible loss of data 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

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.

ESCAN00055307 **Compiler warning: unused static function "XcpMemClr"**

Component@Subcomponent:	Cp_AsrXcp@Implementation
First affected version:	1.03.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The following (or similar) warning occurs:

ctc W536: ["../../BSW/Xcp/XcpProf.c" 642/29] unused static function "XcpMemClr"

Compiler warns for an unused function: can be accepted - is usually because of specific configuration and not too complex \#ifdef usage

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 DAQ measurement is deactivated.

Resolution Description:

Workaround:

The function XcpMemClr can be overwritten. Therefore a user config file can be used to set the define XcpMemClr.

This will deactivate the function

Resolution:

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

ESCAN00065890 Compiler warning: cast discards '__attribute__((noreturn))' qualifier from pointer target type

Component@Subcomponent:	DrvCan_Mpc5700McanLl@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The compiler generates the following warning:

```
Compiling file: ../../../external/BSW/Can/Can.c
../../../external/BSW/Can/Can.c: In function 'CanBasicCanMsgReceived':
../../../external/BSW/Can/Can.c:1745:16: warning: cast discards '__attribute__((noreturn))'
qualifier from pointer target type [-Wcast-qual]
../../../external/BSW/Can/Can.c:1750:10: warning: cast discards '__attribute__((noreturn))'
qualifier from pointer target type [-Wcast-qual]
../../../external/BSW/Can/Can.c:1780:55: warning: cast discards '__attribute__((noreturn))'
qualifier from pointer target type [-Wcast-qual]
```

When does this happen:

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

In which configuration does this happen:

GNU compiler and -Wcast-qual compiler option is used

Resolution Description:

Workaround:

Omit gcc command option -Wcast-qual.

Resolution:

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

ESCAN00065891 Compiler warning: cast increases required alignment of target type

Component@Subcomponent: DrvCan_Mpc5700McanLl@Implementation

First affected version: 1.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Compiler generates the following warning:

```
Compiling file: ../../../external/BSW/Can/Can.c
../../../external/BSW/Can/Can.c: In function 'CanBasicCanMsgReceived':
../../../external/BSW/Can/Can.c:1745:16: warning: cast increases required alignment of target type [-Wcast-align]
../../../external/BSW/Can/Can.c:1750:10: warning: cast increases required alignment of target type [-Wcast-align]
../../../external/BSW/Can/Can.c:1752:29: warning: cast increases required alignment of target type [-Wcast-align]
../../../external/BSW/Can/Can.c:1758:30: warning: cast increases required alignment of target type [-Wcast-align]
```

When does this happen:

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

In which configuration does this happen:

GNU compiler and -Wcast-align compiler option is used

Resolution Description:

Workaround:

Omit gcc command option -Wcast-align

Resolution:

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

ESCAN00067159 Compiler warning: cast truncates constant value

Component@Subcomponent:	MemService_AsrNvM@Implementation
First affected version:	3.08.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

>..\..\bsw\nvm\nvm_crc.c(229) : warning C4310: cast truncates constant value

When does this happen:

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

In which configuration does this happen:

CANoeEmu + VS2008

It depends on definition of uint16_least: Warning occures only if uint16_least is not of type int.

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed, because the cast confirms and enforces this behavior (i.e. the value SHALL be truncated, if necessary).
Additionally: Why uint16_least is not (unsigned) int? -> this data type fulfills all requirements on a 16 bit unsigned value...

Resolution Description:

Workaround:

No workaround necessary.

Resolution:

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

ESCAN00067161 Compiler warning: conditional expression is constant

Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warns for 'conditional expression is constant'.

When does this happen:

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

In which configuration does this happen:

In all configurations with activated optimization "Optimize Const Arrays 2 Define".

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed due to existence of a sufficient workaround.

Resolution Description:

Workaround:

De-activate /MICROSAR/Com/ComGeneral/ComOptimizeConstArrays2Define

Resolution:

The described issue is not planned to be corrected.

ESCAN00067237	**Compiler warning: large type was implicitly cast to small type**
Component@Subcomponent:	If_AsrIfCan@Implementation
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warns that a implicitly cast to a small type could occur.

When does this happen:

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

In which configuration does this happen:

All configurations. The issue occurs with a Renesas compiler for M16C but could happen with every other compiler.

Resolution Description:

Workaround:

No workaround available. No impact on runtime.

Resolution:

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

ESCAN00068402 Compiler warning: conditional expression or part of it is always true/false

Component@Subcomponent:	DrvTrans_Tja1043CandioAsr@Implementation
First affected version:	2.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

- Compiler warns for "condition is always true": This may happen depending on configuration, i.e. assert checks

in function CanTrcv_30_Tja1043_GetOpMode following code is available

```
 returnVal = E_OK;


 CanTrcv_LeaveCritical();

 if (returnVal == E_OK) /* PRQA S 3355,3358 */ /* MD_CanTrcv_30_Tja1043_13.7 */
 {
```

this issues following compiler warning:
warning (dcc:1606): conditional expression or part of it is always true/false

When does this happen:

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

In which configuration does this happen:

All configurations.

Resolution Description:

Workaround:

The code works as intended. The warning is uncritical and can be ignored.

Resolution:

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

ESCAN00068434 Compiler warning: conditional expression or part of it is always true/false

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

Problem Description:

What happens (symptoms):

- Compiler warns for "condition is always true": This may happen depending on configuration, i.e. assert checks

in function Can_SetControllerMode following code is available

```
...
 transitionRequest = kCanRequested;

 CanMicroModeRestore();
 }
 if ( transitionRequest == CAN_NOT_OK ) /* PRQA S 3355,3356,3358,3359 */ /* MD_Can_13.7 */
 { /* PRQA S 3201 */ /* MD_Can_3201 */
 retval = CAN_NOT_OK;
 transitionDone = CAN_NOT_OK; /* at least one HW channel is not in new state (CAN_MSR40: poll later) */
 }

..
```

this issues following compiler warning:
 if (transitionRequest == CAN_NOT_OK) - warning (dcc:1606): conditional expression or part of it is always true/false

When does this happen:

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

In which configuration does this happen:

All configurations.
but not for all Platform implementations (hw always return OK for state transition)

Resolution Description:

Workaround:

Ignore warning

ESCAN00068435 Compiler warning: narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned char

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

Problem Description:
What happens (symptoms):

- Compiler warns for narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned char

Warning occurs in following function:

```
FUNC(void, NVM_PRIVATE_CODE) NvM_QueueInit(void)

 ...
 NvM_JobQueue_at[index].PrevEntry = index - 1u;
```

When does this happen:

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

In which configuration does this happen:

It happens in all configurations

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed due to MISRA 2004 - implicit conversion is allowed in this case.
Additionally, it is obvious that actually no narrowing occurs (even a compiler could be capable of detection). Result of expression is always in range of [0,254].

Resolution Description:
Workaround:

Just ignore warning.

Resolution:

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

ESCAN00068872 Compiler warning: the order of volatile accesses is undefined in this statement

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

Problem Description:

What happens (symptoms):

Compiler issues warning messages like this:
undefined behavior: the order of volatile accesses is undefined in this statement

When does this happen:

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

In which configuration does this happen:

Rx Queue is enabled

Resolution Description:

Workaround:

Ignore Warning

Resolution:

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

ESCAN00074793 Compiler warning: Condition is always constant

Component@Subcomponent: Diag_Asr4Dem@Implementation

First affected version: 4.00.00

Fixed in versions:

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.

ESCAN00076827 Compiler warning: Prototypes for Sector Info not present

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

Problem Description:

What happens (symptoms):

When "Processor and Sector Info" is activated the compiler issues a warning because there are not prototypes for the generated Sector Info constants.

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 "Processor and Sector Info" is used.

Resolution Description:

Workaround:

Activate "General Info" configuration option

Resolution:

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

ESCAN00078543 Compiler warning: statement is unreachable when Calibration is deactivated

Component@Subcomponent:	Cp_AsrXcp@Implementation
First affected version:	1.30.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

The following compiler warnings happen, when the Option "Calibration enabled" is not active:

```
"../../../external/BSW/Xcp/XcpProf.c", line 3267: warning #111-D: statement is
 unreachable
 break;
 ^

"../../../external/BSW/Xcp/XcpProf.c", line 3300: warning #111-D: statement is
 unreachable
 break;
```

When does this happen:

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

--quit_after_warnings

In which configuration does this happen:

When Calibration is deactivated and compiler option above is active.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00081459 Compiler warning: function "ApplCanTimerLoop" was declared but never referenced

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

Problem Description:

What happens (symptoms):

----

compiler warning: function "ApplCanTimerLoop" was declared but never referenced

When does this happen:

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

In which configuration does this happen:

In all configurations where feature SLeep/Wakeup is not enabled. And no other transition needs this hardware transition loop.

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 compiler will remove this function so no effect in code size will occur.
And the code complexity will increase significant to fix this problem by pre-processor switches

Resolution Description:

Workaround:

Ignore Warning

Resolution:

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

ESCAN00086596	**Compiler warning: used wrong Memory class qualifier in several functions**
Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	6.00.00
Fixed in versions:	6.00.01

Problem Description:

What happens (symptoms):

- Compiler warns qualifier does not exist in the following functions:

PduR_GwIf_Transmit_Multi()
PduR_GwIf_TriggerTransmit_Multi()
PduR_GwIf_TxConfirmation_Multi()
PduR_GwIf_CancelTransmit_Multi()

If the qualifier is used to define the memory section the PduR use the Can Interface section for all PduR_GwIf_xxx_Multi() functions.

When does this happen:

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

In which configuration does this happen:

In any N:1 communication Interface routing is configuration.
AND
no Can Interface exits in this configuration.

The PduR_cfg.h contains the following defines:

PDUR_TRANSMITIDXOFTXIF2LOMULTI STD_ON
OR
PDUR_IFTRIGGERTRANSMITFCTPTRIDXOFTXIF2LOMULTI STD_ON
OR
PDUR_IFTXCONFIRMATIONIDXOFTXIF2LOMULTI STD_ON
OR
PDUR_CANCELRECEIVEFCTPTR AND PDUR_TXIF2LOMULTI STD_ON
OR
PDUR_TRANSMITIDXOFTXIF2LOMULTI STD_ON

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00086650 Compiler warning: pointless comparison of unsigned integer with zero

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

Problem Description:

What happens (symptoms):

The generic range check may produce a compiler warning if the CSM_OFFSET_* of a specific service is zero as the range check will compare "uint16 >= 0" which is always TRUE.

When does this happen:

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

In which configuration does this happen:

Every configuration with some compiler.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00087235 Compiler warning: variable "lErrorId" was declared but never referenced

Component@Subcomponent:	Tp_Asr4TpCan@Implementation
First affected version:	2.01.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

```
-------------------------------------------------------------------
"../../../external/BSW/CanTp/CanTp.c", line 3868: warning #550-D: variable
 "lErrorId" was set but never used
 uint8 lErrorId = CANTP_E_NO_ERROR;
 ^

"../../../external/BSW/CanTp/CanTp.c", line 3909: warning #177-D: variable
 "lErrorId" was declared but never referenced
 uint8 lErrorId = CANTP_E_NO_ERROR;
 ^

"../../../external/BSW/CanTp/CanTp.c", line 4258: warning #550-D: variable
 "lErrorId" was set but never used
 uint8 lErrorId = CANTP_E_NO_ERROR;
 ^

"../../../external/BSW/CanTp/CanTp.c", line 4361: warning #550-D: variable
 "lErrorId" was set but never used
 uint8 lErrorId = CANTP_E_NO_ERROR;
 ^

"../../../external/BSW/CanTp/CanTp.c", line 4493: warning #177-D: variable
 "lErrorId" was declared but never referenced
 uint8 lErrorId = CANTP_E_NO_ERROR;
 ^
```

When does this happen:

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

In which configuration does this happen:

If development error detection is disabled (CANTP_DEV_ERROR_DETECT == STD_OFF)
and no dummy statement is used (CANTP_USE_DUMMY_STATEMENT == STD_OFF)

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed because a dummy statement exists to avoid this warning (CW_003)

Resolution Description:

Workaround:

Enable dummy statements

Resolution:

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

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.

ESCAN00087536	Compiler warning: 'function' : conversion from 'const <SomeType>' to '<AnotherType>', possible loss of data
Component@Subcomponent:	CommonAsr_ComStackLib@GenTool_GeneratorMsr
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warns for possible loss of data in the module source code: 'function' : conversion from 'const <SomeType>' to '<AnotherType>', 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:

The module is in the configuration variant postbuild loadable or postbuild loadable selectable
AND
indirections are modelled in the code generator pointing from the configuration class is POSTBUILD to a destination in the configuration class PRE-COMPILE or LINK.

Resolution Description:

Workaround:

Add a type cast if in the embedded source code to avoid the warning.

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.

ESCAN00089241	Compiler warning: multiple warnings
Component@Subcomponent:	SysService_CryptoCv@Impl_actCLib
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

- Compiler warns for possible loss of data: Check if cast is missing and if there is really a data loss due to an implicit/explicit cast on the target platform
- Compiler warns for ambiguous code, parentheses recommended.

When does this happen:

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

In which configuration does this happen:

Always.

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00089425 Compiler warning: missing braces around initializer

Component@Subcomponent:	SysService_CryptoCv@Impl_ESLib
First affected version:	1.01.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

```
-----------------------------------------------------------------
Compiling file: ../../../BSW/SecMod/ESLib_version.c
ctc W542: ["../../../BSW/SecMod/ESLib_version.c" 73/4] missing braces around initializer
```

When does this happen:

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

In which configuration does this happen:

In all configurations.

Resolution Description:

Workaround:

Since ESLib_version.c is only used for component testing, it can be excluded from the build for integration.

Resolution:

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

ESCAN00089543 Compiler warning: dead assignment to "errorId" eliminated

Component@Subcomponent:	Nm_Asr4NmIf@Implementation
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

A compiler warning similar to the following one occurs for the compilation of Nm.c:
dead assignment to "errorId" eliminated

When does this happen:

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

In which configuration does this happen:

'Dev Error Detect' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalProperties/NmDevErrorDetect) in the NmGlobalProperties container is turned OFF in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_DEV_ERROR_REPORT STD_OFF).

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed due to the API pattern that Vector has decided to use: each API that may report development errors shall always have an errorId variable on the stack to which assignments are made - regardless of whether the variable is actually used or not.

Resolution Description:

Workaround:

Ignore the warning.

Resolution:

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

ESCAN00089544 Compiler warning: conversion to 'uint8' from 'int' may alter its value

Component@Subcomponent:	Nm_Asr4NmIf@Implementation
First affected version:	9.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warnings similar to the following one occur for the compilation of Nm.c:
conversion to 'uint8' from 'int' may alter its value

When does this happen:

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

In which configuration does this happen:

'Coordinator Support Enabled' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmCoordinatorSupportEnabled) is turned ON in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_COORDINATOR_SUPPORT_ENABLED STD_ON)

AND

(
'Remote Sleep Ind Enabled' (/MICROSAR/Nm/NmGlobalConfig/NmGlobalFeatures/NmRemoteSleepIndEnabled) is turned OFF in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_REMOTE_SLEEP_IND_ENABLED STD_OFF)

OR

all coordinated channels have 'Channel Sleep Master' (/MICROSAR/Nm/NmChannelConfig/NmChannelSleepMaster) turned ON in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_OPTIMIZE_ALL_SLEEP_MASTER STD_ON)

OR

all coordinated channels have 'Synchronizing Network' (/MICROSAR/Nm/NmChannelConfig/NmSynchronizingNetwork) turned ON in the 'Network Management General' / 'Basic Editor' in DaVinci Configurator (-> Nm_Cfg.h contains #define NM_OPTIMIZE_ALL_SYNC_CHANNEL STD_ON)
)

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed because there is no risk of an invalid conversion of value to uint8.

Resolution Description:

Workaround:

Ignore the warning.

Resolution:

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

ESCAN00089619 Compiler warning: last line of file ends without a newline

Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	11.00.00
Fixed in versions:	12.00.00

Problem Description:

What happens (symptoms):

Compiler warns about missing newline at the end of Dem_Cfg_Macros.h

```
"../../../../bsw/dem/Dem_Cfg_Macros.h", line 431: Warning: #1-D: last line of file ends without a newline
 #endif
 ^
"../../../../bsw/dem/Dem_Cfg_Types.h", line 363: Warning: #1-D: last line of file ends without a newline
 #endif
```

When does this happen:

The warning is issued by the compiler during compilation of the code

In which configuration does this happen:
----
all

Resolution Description:

Workaround:

Ignore the warning

Resolution:

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

ESCAN00090108	Compiler warning: function "PduR_Init_Rpg" was declared but never referenced
Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	5.00.00
Fixed in versions:	7.02.00

Problem Description:

What happens (symptoms):

Compiler warning: function "PduR_Init_Rpg" was declared but never referenced

When does this happen:

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

In which configuration does this happen:

Any configuration where routing path groups are not being used.

The PduR_Lcfg.h file contains the following define

#define PDUR_RPG STD_OFF

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. No impact on the application.

Resolution Description:

Workaround:

Define PduR_Init_Rpg to an empty statement via PduR user configuration file

Resolution:

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

ESCAN00090113 Compiler Warning: Result of function-call is ignored

Component@Subcomponent:	SysService_CryptoCv@Impl_ESLib
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

```
Compiling file: ../../../BSW/SecMod/ESLib_SHA256.c
../../../BSW/SecMod\ESLib_SHA256.c(71): WARNING C1420: Result of function-call is ignored
../../../BSW/SecMod\ESLib_SHA256.c(180): WARNING C1420: Result of function-call is ignored
```

When does this happen:

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

In which configuration does this happen:

----

in all configurations

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00090114 Compiler Warning: Assignment in condition

Component@Subcomponent:	SysService_CryptoCv@Impl_actCLib
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

```
Compiling file: ../../../BSW/SecMod/actBNReduce.c
../../../BSW/SecMod\actBNReduce.c(117): WARNING C5909: Assignment in condition
Compiling file: ../../../BSW/SecMod/actBigNum.c
../../../BSW/SecMod\actBigNum.c(234): WARNING C5909: Assignment in condition
```

When does this happen:

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

In which configuration does this happen:

----

in all configurations

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00090161	**Compiler warning: condition evaluates always to true/false**
Component@Subcomponent:	Ccl_Asr4ComMCfg5@Implementation
First affected version:	7.00.01
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warns for conditional expression being constant
a) in the function ComM_Init() when checking the generated data. Compiler warns about condition being always false in the following conditions:
if (ComM_GetWakeupStateOfChannel(ComM_ChannelIndex) >= COMM_MAX_NUMBER_OF_STATES)
if (ComM_GetSizeOfChannel() != ComM_GetSizeOfChannelPb())
if (ComM_GetSizeOfPnc() != ComM_GetSizeOfPncPb())
As secondary effect compiler might warn about unreachable code/statement.

b) in the function ComM_PncProcessRxSignalEra() compiler warns about condition being always true in
if(ComM_IsSynchronizedOfPnc(pncIndex))

c) in the functions ComM_PncSetBitInSignal() and ComM_PncClearBitInSignal() when checking the generated data. Compiler warns about condition being always true in
if(signalByteIndex < ComM_GetSizeOfPncSignalValues())

When does this happen:

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

In which configuration does this happen:

----

a) occurs when COMM_DEV_ERROR_DETECT == STD_ON

b) occurs when
- 'Pnc Support' is enabled in ComM (/MICROSAR/ComM/ComMGeneral/ComMPncSupport)
AND
- 'Pnc Gateway Enabled' is enabled in ComM (/MICROSAR/ComM/ComMGeneral/ComMPncGatewayEnabled)
AND
- Only one PNC exists (COMM_ACTIVE_PNC == 1U, can be found in ComM_Cfg.h).

c) occurs when 'Pnc Support' is enabled in ComM (/MICROSAR/ComM/ComMGeneral/ComMPncSupport)

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code.
Nevertheless it will not be fixed because no simple remedy exist.
The warning is caused by an if-statement applied on external configuration data. Configuration data is const for the given compilation context but might be changed at post-build time.

Resolution Description:

ESCAN00090806 Compiler warning: C4310: cast truncates constant value

Component@Subcomponent: Gw_AsrPduRCfg5@Implementation

First affected version: 7.00.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Compiler warns for cast truncates constant value

When does this happen:

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

In which configuration does this happen:

If the uint8_least is of type unsigned char

The Platform_Types.h contains the following define

typedef unsigned char uint8_least; /* At least 8 bit */

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code. Nevertheless it will not be fixed due to ensure that the init value is large enough. A cast to a smaller value is acceptable and has no impact on the application.

#define PDUR_INVALID_VARARRAYIDX ((uint16)0xFFFF) is cast for unsigned char to 0xFF which is correct.

Resolution Description:

Workaround:

----

ignore the warning

Resolution:

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

ESCAN00090831	**Compiler warning: integer conversion resulted in a change of sign**
Component@Subcomponent:	Il_AsrComCfg5@Implementation
First affected version:	1.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

---

Compiler warns that "integer conversion resulted in a change of sign".

When does this happen:

---

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

In which configuration does this happen:

---

If the compiler WindRiver Diab is used. (found with version 5.9.4.2.)

Hint:

---

The compiler warning is known and has been analyzed thoroughly for its impact on the code.

Resolution Description:

Workaround:

---

No workaround available.

Resolution:

---

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

ESCAN00090874 Compiler warning: DEM_CFG_SUPPORT_NVM_POLLING undefined

Component@Subcomponent:	Diag_Asr4Dem@Implementation
First affected version:	11.00.00
Fixed in versions:	12.00.00

Problem Description:

What happens (symptoms):

Compile warning occurs in Dem_Cbk.h:
'DEM_CFG_SUPPORT_NVM_POLLING' is not defined as a preprocessor macro, replacing with '0' for '#if/#elif'

When does this happen:

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

In which configuration does this happen:

In all configurations using .c-Files which include Dem_Cbk.h

MICROSAR4 DEM and MICROSAR NVM are not affected

Resolution Description:

Workaround:

Define DEM_CFG_SUPPORT_NVM_POLLING manually to STD_OFF
OR
Include Dem.h before including Dem_Cbk.h

Resolution:

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

ESCAN00091295 Compiler warning: dead assignment / variable set but not used

Component@Subcomponent:	Ccl_Asr4ComMCfg5@Implementation
First affected version:	5.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warns about an useless assignment to a local variable. Typically the warnings refer to local variables 'channel', 'errorId', 'Status' or 'User'.

Example compiler warning strings:

"Useless assignment to variable 'abc'. Assigned value not used."

"Removed dead assignment"

When does this happen:

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

In which configuration does this happen:

EcuC Parameter 'Dummy Statement Kind' is set to 'SelfAssignment'. This can be detected in ComM_Cfg.c: #define COMM_DUMMY_STATEMENT(v) (v)=(v)

Hint:

The compiler warning is known and has been analyzed thoroughly for its impact on the code.

Nevertheless it will not be fixed because no simple remedy exist.

If Dummy Statement is switched off, other compiler warnings might occur e.g. "Unused/ unreferenced variable".

Resolution Description:

ESCAN00091340 Compiler warning: cast truncates constant value

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

Problem Description:

What happens (symptoms):

Compile warning occurs.

When does this happen:

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

In which configuration does this happen:

If partial network wakeup PDU filtering is active.
(canifcfg.h: CANIF_PN_WU_TX_PDU_FILTER == STD_ON)

Resolution Description:

Workaround:

No workaround available. Issue is checked and not critical.

Resolution:

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

ESCAN00091343	Compiler warning: warning C4310: cast truncates constant value
Component@Subcomponent:	If_AsrIfCan@Implementation
First affected version:	6.09.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compile warning occurs.

When does this happen:

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

In which configuration does this happen:

If transmit buffer is configured as FIFO and cancel API is supported.
(canifcfg.h: CANIF_TRANSMIT_BUFFER_FIFO == STD_ON && CANIF_CANCEL_SUPPORT_API == STD_ON)

Resolution Description:

Workaround:

No workaround available. Warning was checked, not critical.

Resolution:

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

ESCAN00091547 Compiler warning: condition is always false

Component@Subcomponent: Diag_Asr4Dem@Implementation

First affected version: 11.01.00

Fixed in versions:

Problem Description:

What happens (symptoms):

Compiler warns for "condition is always true/false"
Some compiler will also warn because of dead code, resulting from the constant condition

When does this happen:

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

In which configuration does this happen:

Dem/DemGeneral/DemSafeBswModeEnabled == TRUE
AND
EcuC/EcucGeneral/EcuCSafeBswChecks == TRUE or undefined

Depending on the configuration, optimization can change a configuration table into a constant macro.
This can causes some run-time checks to check for equality of two constants.

Resolution Description:

Workaround:

The warning can be ignored.

Resolution:

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

ESCAN00092315 Compiler warning: function "CanLL_WakeUpHandling" was declared but never referenced

Component@Subcomponent:	DrvCan_Mpc5700McanLl@Implementation
First affected version:	2.00.00
Fixed in versions:	2.10.00

Problem Description:

What happens (symptoms):

Compiler warning occurs: "function "CanLL_WakeUpHandling" was declared but never referenced"

When does this happen:

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

In which configuration does this happen:

If "Sleep /Wake-up Functionality" is activated in the configuration (leading to the definition of C_ENABLE_SLEEP_WAKEUP).

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00092713 Preprocessor parse error

Component@Subcomponent:	DrvCan_Mpc5700McanLl@Implementation
First affected version:	2.07.00
Fixed in versions:	2.10.00

Problem Description:

What happens (symptoms):

---

Preprocessor stops with parsing error.
E.g. the MCAN can still be active with pending Tx requests although Stop Mode reached is notified.

When does this happen:

---

At compilation time.

In which configuration does this happen:

---

Only for CANbedded
AND
Range Filtering is used (C_ENABLE_RANGE_x is defined).

Resolution Description:

Workaround:

---

No workaround available.

Resolution:

---

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

ESCAN00093058 Compiler warning: conversion from 'PduIdType' to 'PduR_TxMulti2LoStateIterType', possible loss of data

Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	7.00.00
Fixed in versions:	

Problem Description:

What happens (symptoms):

Compiler warning: Conversion from 'PduIdType' to 'PduR_TxMulti2LoStateIterType', 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:

N:1 Tp routing configuration

The PduR_Lcfg.h file contains the following define:

#define PDUR_TXMULTI2LOSTATE STD_ON

Resolution Description:

Workaround:

No workaround available. Missing explicit cast. The warning can be ignored

Resolution:

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

ESCAN00093096 Compiler warning: truncating assignment

Component@Subcomponent:	Ccl_Asr4ComMCfg5@Implementation
First affected version:	5.00.00
Fixed in versions:	8.01.00

Problem Description:

What happens (symptoms):

Compiler warns about possible loss of data in the function ComM_LimitPncToChannelRouting() for example

ComM.c 1591(190+5) truncating assignment

There is no risk of data loss and the warning can be ignored.

Background:

The expression is like (type) = (type) | (uint8)

The result of logical OR and AND operator has type int according to C standard. The warning is issued because the affected expression implicitly casts the result to (type) if (type) is not int.

There is no risk of data loss because (type) is uint8 or uint16.

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:

ComM/ComMGeneral/ComMPncToChannelRoutingLimitationEnabled == true

Resolution Description:

Workaround:

No workaround available.

Resolution:

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

ESCAN00093790	Compiler warning: conversion from 'PduIdType' to 'PduR_TxMulti2LoStateIterType', possible loss of data
Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	7.00.00
Fixed in versions:	10.00.00

Problem Description:

What happens (symptoms):

The external type is larger than the internal type. The internal type is adapted automatically during generation to a suite able type due to optimization reason. If the internal type is smaller than the external type the value of the external type could never be greater than the internal type could hold, because there is a direct dependency between the value range.

The warning can be ignored the cast is always valid.

When does this happen:

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

In which configuration does this happen:

If the /MICROSAR/EcuC/EcucPduCollection/PduIdTypeEnum is set to UINT16, but the maximum configured PDU handle is smaller than 255

Resolution Description:

Workaround:

Solve: ECUC01006 PduIdTypeEnum validation and reduce the type to the suite able type

Resolution:

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

ESCAN00093792	Compiler warning: conversion from 'PduIdType' to 'PduR_RmSrcRomIterType', possible loss of data
Component@Subcomponent:	Gw_AsrPduRCfg5@Implementation
First affected version:	9.00.00
Fixed in versions:	10.00.00

Problem Description:

What happens (symptoms):

The external type is larger than the internal type. The internal type is adapted automatically during generation to a suite able type due to optimization reason. If the internal type is smaller than the external type the value of the external type could never be greater than the internal type could hold, because there is a direct dependency between the value range.

The warning can be ignored the cast is always valid.

When does this happen:

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

In which configuration does this happen:

If the /MICROSAR/EcuC/EcucPduCollection/PduIdTypeEnum is set to UINT16, but the maximum configured PDU handle is smaller than 255

Resolution Description:

Workaround:

Solve: ECUC01006 PduIdTypeEnum validation and reduce the type to the suite able type

Resolution:

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

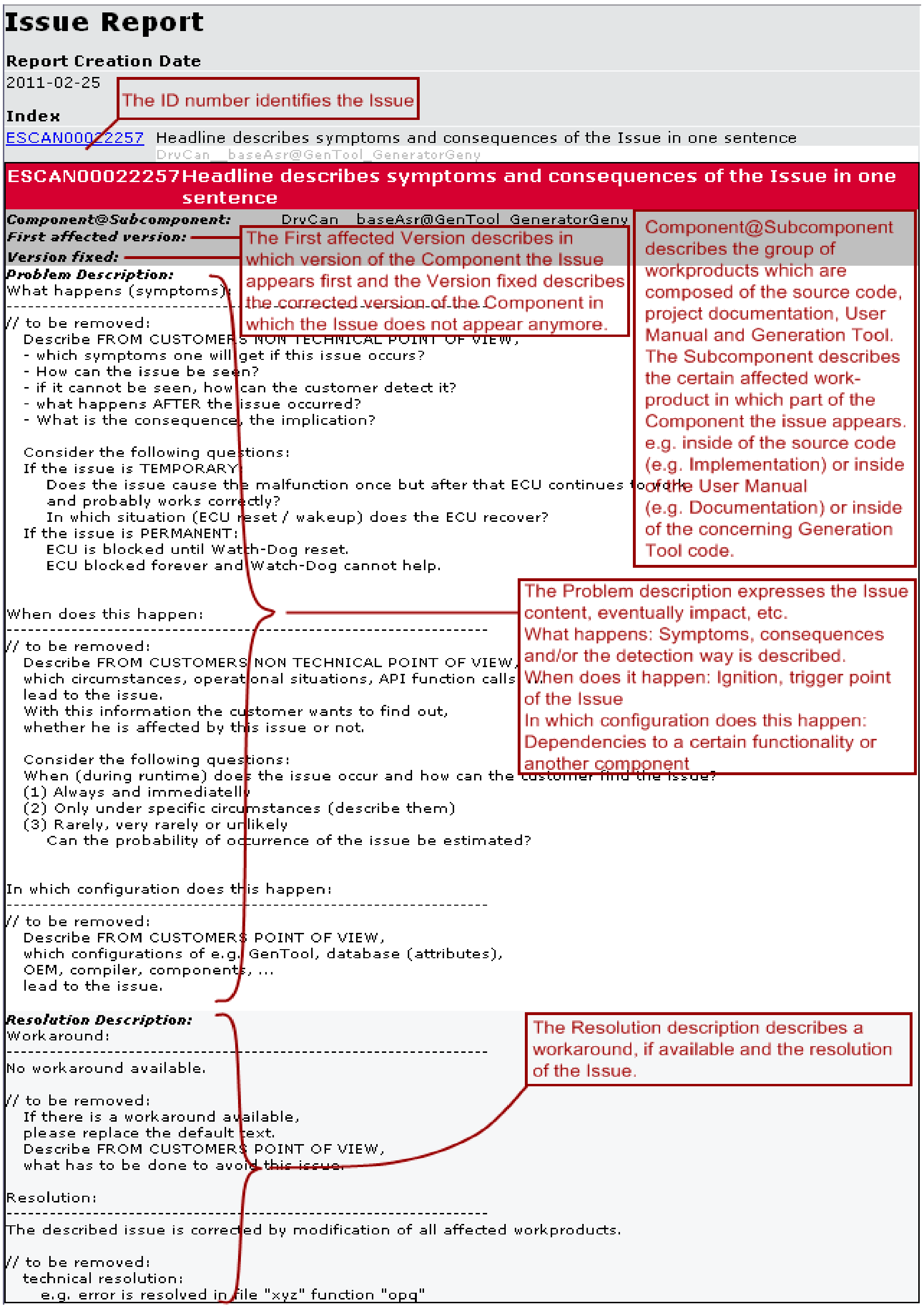
3. New Issues for Information

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

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

No issue to be reported.

4. Report Legend



Issue Report

Report Creation Date

2011-02-25

Index

ESCAN00022257 Headline describes symptoms and consequences of the Issue in one sentence

DrvCan__baseAsr@GenTool_GeneratorGeny

ESCAN00022257Headline describes symptoms and consequences of the Issue in one sentence

Component@Subcomponent: DrvCan__baseAsr@GenTool_GeneratorGeny

First affected version:

Version fixed:

Problem Description:

```
What happens (symptoms):
-------------------------------------------------------------
// to be removed:
   Describe FROM CUSTOMERS NON TECHNICAL POINT OF VIEW,
   - which symptoms one will get if this issue occurs?
   - How can the issue be seen?
   - if it cannot be seen, how can the customer detect it?
   - what happens AFTER the issue occurred?
   - What is the consequence, the implication?

   Consider the following questions:
   If the issue is TEMPORARY:
       Does the issue cause the malfunction once but after that ECU continues to
       and probably works correctly?
       In which situation (ECU reset / wakeup) does the ECU recover?
   If the issue is PERMANENT:
       ECU is blocked until Watch-Dog reset.
       ECU blocked forever and Watch-Dog cannot help.


When does this happen:
-------------------------------------------------------------
// to be removed:
   Describe FROM CUSTOMERS NON TECHNICAL POINT OF VIEW,
   which circumstances, operational situations, API function calls
   lead to the issue.
   With this information the customer wants to find out,
   whether he is affected by this issue or not.

   Consider the following questions:
   When (during runtime) does the issue occur and how can the customer find the issue?
   (1) Always and immediately
   (2) Only under specific circumstances (describe them)
   (3) Rarely, very rarely or unlikely
       Can the probability of occurrence of the issue be estimated?


In which configuration does this happen:
-------------------------------------------------------------
// to be removed:
   Describe FROM CUSTOMERS POINT OF VIEW,
   which configurations of e.g. GenTool, database (attributes),
   OEM, compiler, components, ...
   lead to the issue.
```

Resolution Description:

```
Workaround:
-------------------------------------------------------------
No workaround available.

// to be removed:
   If there is a workaround available,
   please replace the default text.
   Describe FROM CUSTOMERS POINT OF VIEW,
   what has to be done to avoid this issue.

Resolution:
-------------------------------------------------------------
The described issue is corrected by modification of all affected workproducts.

// to be removed:
   technical resolution:
      e.g. error is resolved in file "xyz" function "opq"
```

The ID number identifies the Issue

The First affected Version describes in which version of the Component the Issue appears first and the Version fixed describes the corrected version of the Component in which the Issue does not appear anymore.

Component@Subcomponent describes the group of workproducts which are composed of the source code, project documentation, User Manual and Generation Tool. The Subcomponent describes the certain affected work-product in which part of the Component the issue appears. e.g. inside of the source code (e.g. Implementation) or inside of the User Manual (e.g. Documentation) or inside of the concerning Generation Tool code.

The Problem description expresses the Issue content, eventually impact, etc. What happens: Symptoms, consequences and/or the detection way is described. When does it happen: Ignition, trigger point of the Issue In which configuration does this happen: Dependencies to a certain functionality or another component

The Resolution description describes a workaround, if available and the resolution of the Issue.

5. 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