# Issue Report

| License Number | Customer |
|---|---|
| CBD1300660 | Nexteer Automotive Corporation<br>Package: CBD Psa SLP4<br>Micro: 0812BPGEQQ1<br>Compiler: TexasInstruments 4.9.5 |

**Maintenance Expiry Date**

2024-03-18

| SIP Delivery Date | SIP Version |
|---|---|
| 2014-03-18 | 05.00.17 |

| SLP | Delivery Number |
|---|---|
| CBD Psa SLP4 | D01 |

**Report Creation Date**

2014-04-02

**Contact**

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

## Table of Contents

# 1. Introduction

## 1.1 Resolving Issues

Reported issues are not necessarily fixed automatically by the next update delivery. If some of the reported issues shall be fixed, please contact Vector to establish an agreement about issues that shall be fixed in upcoming deliveries. Please note that Vector may fix additional issues without explicit request.

## 1.2 Issue Classification

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

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

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

- **Runtime Issues with Workaround:** It is not recommended to update a delivery due to a runtime issue with a documented workaround. The effect of an issue to the ECU functionality has to be analyzed by the customer as the basic software usage and its configuration is not known by Vector. The risk of change has also to be taken into account.
- **Compiler Warnings:** As a service we report the known compiler warnings. The occurrence of a compiler warning may depend on the used configuration and compiler settings.
- **Apparent Issues:** Apparent issues are detected immediately when using the basic software. If an issue does not show up while working with the basic software the ECU project is not affected by the issue. Apparent issues may or may not have workarounds.

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

# 2. New Issues

## 2.1 Runtime Issues without Workaround

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

## 2.2 Runtime Issues with Workaround

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

**Index**

| | |
|---|---|
| ESCAN00045854 | An incorrect timeout is issued for Flow Control and Consecutive Frame timing supervision.<br>Tp_Iso15765@GenTool_Geny |
| ESCAN00055528 | Missing call-context limitation in the description of all DescSetStateXXX API<br>Diag_CanDesc__coreBase@Doc_TechRef |
| ESCAN00056993 | Busoff event incorrectly also causes wakeup event<br>DrvCan_Tms470DcanHll@Implementation |
| ESCAN00065128 | CANbedded only: multiplex messages are not received correctly<br>GenTool_GenyDriverBase@GenTool_Geny |
| ESCAN00066659 | canbedded only: multiplex messages are not received correctly<br>Hw__baseCpuCan@GenTool_Geny |
| ESCAN00070923 | Overrun occurs with higher probability<br>DrvCan_Tms470DcanLl@Implementation |

## ESCAN00045854 An incorrect timeout is issued for Flow Control and Consecutive Frame timing supervision.

| | |
|---|---|
| ***Component@Subcomponent:*** | Tp_Iso15765@GenTool_Geny |
| ***First affected version:*** | 2.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
--------------------------------------------------------------------
An incorrect timeout is issued for Flow Control (TX) and Consecutive Frame (RX) timing supervision in case of large timeouts.

When does this happen:
--------------------------------------------------------------------
During runtime at transmission and/or reception of multi frames.

In which configuration does this happen:
--------------------------------------------------------------------
This can only appear if channel specific timing is activated (#if defined TP_CHANNEL_SPECIFIC_TIMING)
AND
the configured timeout values are greater than 255 "ticks".
Please note that the number of "ticks" is calculated by dividing the configured timeout value by the configured periodic cycle time of the TP.

***Resolution Description:***
Workaround:
--------------------------------------------------------------------
Use smaller timeouts or increase the call-cycle of the TP task functions.

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

| **ESCAN00055528** | **Missing call-context limitation in the description of all DescSetStateXXX API** |
|---|---|
| ***Component@Subcomponent:*** | Diag_CanDesc__coreBase@Doc_TechRef |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | 3.06.00 |

***Problem Description:***

What happens (symptoms):

-------------------------------------------------------------------

Since the technical reference CANdesc does not restrict the call-context of the "DescSetStateXXX" API, the application might call it from interrupt context or a task with higher priority than the DescTask.

This might result in undefined run time effects.

When does this happen:

-------------------------------------------------------------------

During diagnostic application integration, when using a "DescSetStateXXX" API.

In which configuration does this happen:

-------------------------------------------------------------------

Any configuration.

***Resolution Description:***

Workaround:

-------------------------------------------------------------------

Do not call any of the "DescSetStateXXX" APIs from interrupt context or a task with higher priority than the DescTask.

Resolution:

-------------------------------------------------------------------

Call context has been restricted to a task with priority lower or equal to the DescTask.

## ESCAN00056993 Busoff event incorrectly also causes wakeup event

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan_Tms470DcanHll@Implementation |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***

What happens (symptoms):

-------------------------------------------------------------------

When a busoff event is detected (via either CAN interrupt or polling CanTask), the driver executes the code to handle the busoff correctly, but it also incorrectly executes the code to handle a wakeup event, even though no wakeup event is pending.

When does this happen:

-------------------------------------------------------------------

On the next busoff event, if the last wakeup event was not immediately followed by at least 11 recessive bits.

In other words, if the bus is "noisy" when CAN wakes up, the next busoff event will cause the driver to execute the wakeup routine again.

In which configuration does this happen:

-------------------------------------------------------------------

Configurations where 'Sleep/Wakeup Functionality' is enabled on the DrvCan_Tms470DcanHll page in GENy.

***Resolution Description:***

Workaround:

-------------------------------------------------------------------

If global power down mode is configured: In ApplCanWakeUpFromSleepModeRequest, after clearing the appropriate PCR bits as documented in the CAN driver technical reference, the application must wait for the 'WakeUp Pnd' bit to clear in the DCAN Error and Status register. For example:

```
 while((*(vuint32 *)0xFFF7DC04 /* DCAN1 Error and Status register */) & (vuint32)0x00000200);
 {
 }
```

It is also recommended that the application have some sort of timeout for this loop in case the bus is permanently disturbed.

Resolution:

-------------------------------------------------------------------

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

## ESCAN00065128 CANbedded only: multiplex messages are not received correctly

| | |
|---|---|
| ***Component@Subcomponent:*** | GenTool_GenyDriverBase@GenTool_Geny |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | 2.09.00 |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
With IL: wrong signal values are received.

The RDS message structs of the multiplexed messages are generated incorrect to the can_par.h file.
The Il will access the wrong value as multiplexor signal within the message.

This Issue is fixed together with ESCAN00066659

When does this happen:
-------------------------------------------------------------------
At runtime on access to the multiplexed signals by the application if the described configuration is valid for the message of this signal

In which configuration does this happen:
-------------------------------------------------------------------
In configurations in which
- Multiplex messages
AND
- the multiplexor of a multiplexed message is not in the first byte
AND
- the signals before the multiplexor value are not assigned as receive message for this ECU.
ADN
- Interaction Layer used

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
access the multiplexor signal by using the buffer array access.
or if possible
Add your ECU as receiver of the signals before the multiplexor value.

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

| ESCAN00066659 | canbedded only: multiplex messages are not received correctly |
|---|---|
| ***Component@Subcomponent:*** | Hw__baseCpuCan@GenTool_Geny |
| ***First affected version:*** | 2.22.04 |
| ***Fixed in versions:*** | 2.27.00 |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
With IL: wrong signal values are received.

The message structs of the multiplexed messages are generated incorrect to the drv_par.h file.
The Il will access the wrong value as multiplexor signal within the message.

This Issue is fixed together with ESCAN00065128

When does this happen:
-------------------------------------------------------------------
At runtime on access to the multiplexed signals by the application if the described configuration is valid for the message of this signal

In which configuration does this happen:
-------------------------------------------------------------------
In configurations in which
- Multiplex messages
AND
- the multiplexor of a multiplexed message is not in the first byte
AND
- the signals before the multiplexor value are not assigned as receive message for this ECU.
AND
- Il_Vector is used

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
access the multiplexor signal by using the buffer array access.
or if possible
Add your ECU as receiver of the signals before the multiplexor value.

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

## ESCAN00070923 Overrun occurs with higher probability

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan_Tms470DcanLl@Implementation |
| ***First affected version:*** | 1.07.00 |
| ***Fixed in versions:*** | 1.20.00 |

***Problem Description:***

What happens (symptoms):
-------------------------------------------------------------------
Overrun occurs with higher probability, CAN communication still works.

When does this happen:
-------------------------------------------------------------------
When BasicCAN overrun occurs.

The following must happen to achieve that the overrun functionality behaves as expected again:
MSR:
1. Reinitialization of the CAN controller by calling the function Can_InitController().
2. Mode change to stop mode by calling the function Can_SetControllerMode(CAN_T_STOP).

CANbedded:
1. Reinitialization of the CAN controller by calling the function CanInit().
2. Mode change by calling the function CanStart()
(does only work if workaround for issue 22 is not disabled in user configuration file - enabled by default)

Further BasicCAN overrun leads to same issue again.

In which configuration does this happen:
-------------------------------------------------------------------
MSR:
BasicCAN objects are used
AND
feature "overrun notification" is either set to APPL or DET.
(see can_cfg.h if CAN_OVERRUN_NOTIFICATION is set either to CAN_APPL or CAN_DET)

CANbedded:
BasicCAN objects are used
AND
feature "overrun notification" is enabled.
(see can_cfg.h if C_ENABLE_OVERRUN is set)

***Resolution Description:***

Workaround:
-------------------------------------------------------------------
Disable overrun notification.

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

## 2.3 Apparent Issues

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

**Index**

| | |
|---|---|
| ESCAN00024492 | API prototypes for InmNmRxGetCondition are not correct<br>Nm_IndOsek@Doc_TechRef |
| ESCAN00039653 | The interrupt lock functions do not work correctly for the user mode<br>VStdLib_Arm7@Implementation |
| ESCAN00047907 | limitation "InmNmTask" (no interrupt context usage)<br>Nm_IndOsek@Doc_TechRef |
| ESCAN00049589 | Compile error: direct signal access feature in CANdesc does not consider far memory pointers<br>Diag_CanDesc__coreBase@Implementation |
| ESCAN00053779 | Linker error: CanBaseAddressRequest() and CanBaseAddressActivate() are not available<br>DrvCan__coreHll@Implementation |
| ESCAN00055957 | appdesc.c missing line feed (LF) after carraige return (CR) on some lines<br>Diag_CanDesc__coreBase@Implementation |
| ESCAN00056617 | Compile error when compiling CanInterruptDisable(): missing ;<br>DrvCan__coreHll@Implementation |
| ESCAN00059562 | Compile error: Size of array CanRxMsgIndirection is zero if index search and no Rx FullCANs are used<br>DrvCan__coreHll@Implementation |
| ESCAN00062165 | Compiler error: Interrupt control macros prevent can_drv.c from being compiled in THUMB mode<br>VStdLib_Arm7@Implementation |
| ESCAN00062316 | [canbedded only] Wrong Rx Data Length of message displayed<br>GenTool_GenyDriverBase@GenTool_Geny |
| ESCAN00062872 | the function CanLL_HandleIllIrptNumber didn't clear a illegal interrupt<br>DrvCan_Tms470DcanLl@Implementation |
| ESCAN00063756 | certain extended IDs may not be received after Full CAN overrun ( if extended ID masking is enabled )<br>DrvCan_Tms470DcanLl@Implementation |
| ESCAN00070517 | Compiler error: missing constant kDescStateSessionDefault<br>Diag_CanDesc__coreBase@Implementation |
| ESCAN00071804 | Functions and flags can be added to a Update Bit signal after an dbc update was performed<br>Il_Vector@GenTool_Geny |
| ESCAN00073608 | "Unknown Service Support" feature in GENy is referenced as "Support Generic User Service" feature<br>Diag_CanDesc__coreBase@Doc_TechRef |

| ESCAN00024492 | API prototypes for InmNmRxGetCondition are not correct |
|---|---|
| ***Component@Subcomponent:*** | Nm_IndOsek@Doc_TechRef |
| ***First affected version:*** | 1.12.00 |
| ***Fixed in versions:*** | 1.13.00 |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
The API prototypes for InmNmRxGetCondition() are not correct. They contain the function name InmNmRxTimeOut() instead of InmNmRxGetCondition() .

When does this happen:
-------------------------------------------------------------------
When reading the document

In which configuration does this happen:
-------------------------------------------------------------------
This issue occurs always.

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
When using the API in the code, use the correct API name instead of the one given in the API prototype.

Resolution:
-------------------------------------------------------------------
The described issue is corrected by modification of all affected workproducts.

**ESCAN00039653** **The interrupt lock functions do not work correctly for the user mode**

| | |
|---|---|
| ***Component@Subcomponent:*** | VStdLib_Arm7@Implementation |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
--------------------------------------------------------------------
The interrupt lock functions have no effect and the interrupt will never be locked. This could lead to data inconsistency.
This issue is permanent and happens more often with higher system load.

When does this happen:
--------------------------------------------------------------------
This happens at runtime.

In which configuration does this happen:
--------------------------------------------------------------------
It happens if the CPU core is running in the user mode.

For this mode, the change from enabled to disabled interrupts and vice versa is not possible.
To do this, the privileged operation mode is necessary. The current implementation of the CAN driver
does not support switching from user mode to privileged mode and so the interrupt lock mechanism of
the CAN driver does not work with the user mode.

***Resolution Description:***
Workaround:
--------------------------------------------------------------------
Run the CAN driver in the privileged operating mode.

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

**ESCAN00047907** **limitation "InmNmTask" (no interrupt context usage)**

| | |
|---|---|
| ***Component@Subcomponent:*** | Nm_IndOsek@Doc_TechRef |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
The application must not call the service function "InmTask" in interrupt context (according to the general CBD design rules). But such a limitation is not described in the technical reference.

When does this happen:
-------------------------------------------------------------------
When reading the technical reference.

In which configuration does this happen:
-------------------------------------------------------------------
Does not depend on any configuration.

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
No workaround available.

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

## ESCAN00049589 Compile error: direct signal access feature in CANdesc does not consider far memory pointers

| | |
|---|---|
| ***Component@Subcomponent:*** | Diag_CanDesc__coreBase@Implementation |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***

What happens (symptoms):

---------------------------------------------------------------------

Compile error for mismatching pointer type assignment.

When does this happen:

---------------------------------------------------------------------

At compile time.

In which configuration does this happen:

---------------------------------------------------------------------

- CANdesc

AND

- Direct signal access to RAM/ROM objects is used.

AND

- FAR memory

Some services such as the UDS ones 0x22/0x2A and 0x2E, can be processed on signal level. If they are processed on signal level it is possible to choose "Direct Access" as Signal Handler Type. In this case, CANdesc reads or writes the value of signal direct of/to a variable. (The name of the variable is configured in the cdd file or GENy.) If this variable is located in FAR memory a Compiler/Linker warning or error will occur.

***Resolution Description:***

Workaround:

---------------------------------------------------------------------

Avoid direct signal access to such objects and implement the main-handler within the application code. (Choose "Signal Handler" for the Signal Handler Type and copy the data that is located in the FAR memory in the application callback for this signal.)

Resolution:

---------------------------------------------------------------------

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

| ESCAN00053779 | Linker error: CanBaseAddressRequest() and CanBaseAddressActivate() are not available |
|---|---|
| ***Component@Subcomponent:*** | DrvCan__coreHll@Implementation |
| ***First affected version:*** | 2.10.00 |
| ***Fixed in versions:*** | 2.14.00 |

***Problem Description:***
What happens (symptoms):
--------------------------------------------------------------------
A Linker error occurs: CanBaseAddressRequest() and CanBaseAddressActivate() are not available

When does this happen:
--------------------------------------------------------------------
This happens at link time

In which configuration does this happen:
--------------------------------------------------------------------
This happens if virtual addressing is activated for the CAN driver without QNX support:
- C_ENABLE_UPDATE_BASE_ADDRESS is activated in can_cfg.h
AND
- VGEN_ENABLE_MDWRAP is not defined in the system
AND
- VGEN_ENABLE_QWRAP is not defined in the system

This is a special feature which is not implement in general.

***Resolution Description:***
Workaround:
--------------------------------------------------------------------
add the following code to the user configuration file of the CAN driver:

```
#if defined C_ENABLE_UPDATE_BASE_ADDRESS
#define C_ENABLE_BASE_ADDRESS_UPDATE
```

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

**ESCAN00055957** **appdesc.c missing line feed (LF) after carraige return (CR) on some lines**

| | |
|---|---|
| ***Component@Subcomponent:*** | Diag_CanDesc__coreBase@Implementation |
| ***First affected version:*** | 5.07.26 |
| ***Fixed in versions:*** | |

***Problem Description:***

What happens (symptoms):

-------------------------------------------------------------------

The appdesc.c file is missing the line feed (LF) character at the end of certain lines. It should follow the carraige return (CR) character. This will cause compilers and debuggers to display the incorrect line of source code. Additionally, some IDEs will complain that the line feed character is missing.

When does this happen:

-------------------------------------------------------------------

At generation time.

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.

## ESCAN00056617 Compile error when compiling CanInterruptDisable(): missing ;

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan__coreHll@Implementation |
| ***First affected version:*** | 2.01.00 |
| ***Fixed in versions:*** | 2.14.00 |

***Problem Description:***

What happens (symptoms):

-------------------------------------------------------------------

The compiler gives an error because of a missing ; when compiling
CanInterruptDisable();

In some configurations (see below), this is a macro:
```
# define CanInterruptDisable() (VStdSuspendAllInterrupts())
```

VStdSuspendAllInterrupts() is defined in osek.h. If it is a function,
there is no problem.
If it is itself a macro and if the macro starts with {, the code doesn't compile.

The same problem happens with the CanInterruptRestore() macro.

When does this happen:

-------------------------------------------------------------------

At compile time.

In which configuration does this happen:

-------------------------------------------------------------------

Any configuration with "Locking Mechanism" of VStdLib set to "OSEK".
There is only a problem if SuspendAllInterrupts() is a macro beginning with {

***Resolution Description:***

Workaround:

-------------------------------------------------------------------

configure "Locking Mechanism" of VStdLib to "User defined"
call SuspendAllInterrupts() and ResumeAllInterrupts() within the application function.

Resolution:

-------------------------------------------------------------------

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

## ESCAN00059562 Compile error: Size of array CanRxMsgIndirection is zero if index search and no Rx FullCANs are used

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan__coreHll@Implementation |
| ***First affected version:*** | 2.00.00 |
| ***Fixed in versions:*** | 2.15.01 |

***Problem Description:***

What happens (symptoms):
-------------------------------------------------------------------
the compiler complains about "the size of an array must be greater than zero" about the following code in can_def.h:

```
 V_MEMROM0 extern V_MEMROM1 CanReceiveHandle V_MEMROM2
CanRxMsgIndirection[kCanNumberOfRxFullCANObjects];
```

The tabel CanRxMsgIndirection table itself is not available and no access to this table is performed if not generated/necessary is no Rx FullCANs messages are used.

When does this happen:
-------------------------------------------------------------------
This happens during compile process.

In which configuration does this happen:
-------------------------------------------------------------------
HighEnd-license is used
AND
Index search is used
AND
No Rx FullCAN objects are used.

It depends on the Compiler whether no information is given or a compiler warning or compiler error is generated. E.g. the GNU compiler don't complain about this.

The following compiler is currently known to be affected:
MPC Greenhills v5.2.4

***Resolution Description:***

Workaround:
-------------------------------------------------------------------
a) A compiler warning can be ignored.
b) A compiler error can be reduced to warning or ignored if possible. Otherwise there is no workaround available.

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

## ESCAN00062165 Compiler error: Interrupt control macros prevent can_drv.c from being compiled in THUMB mode

| | |
|---|---|
| ***Component@Subcomponent:*** | VStdLib_Arm7@Implementation |
| ***First affected version:*** | 0.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
VStdLL_GlobalInterruptDisable and VStdLL_GlobalInterruptRestore are implemented as macros instead of functions. Since these macros use the compiler intrinsics __get_CPSR() and __set_CPSR(), any file which relies on these functions (such as can_drv.c) cannot be compiled in THUMB mode (compiler option -mt).

The following compiler warning occur in any file which uses VStdLL_GlobalInterruptDisable or VStdLL_GlobalInterruptRestore:

warning #1445-D: intrinsic "_get_CPSR" not supported in thumb mode, treated as function call
warning #1445-D: intrinsic "_set_CPSR" not supported in thumb mode, treated as function call

In the linking phase, the symbols _get_CPSR and _set_CPSR will be undefined, preventing the project from building.

When does this happen:
-------------------------------------------------------------------
At compile time.

In which configuration does this happen:
-------------------------------------------------------------------
TI compiler
AND
can_drv.c or other invoking file compiled in THUMB mode (compiler option -mt)
AND
Configurations which use default interrupt control in GENy on the Hw page
(all conditions must be true for the issue to occur)

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
No workaround available.

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

| **ESCAN00062316** | **[canbedded only] Wrong Rx Data Length of message displayed** |
|---|---|
| ***Component@Subcomponent:*** | GenTool_GenyDriverBase@GenTool_Geny |
| ***First affected version:*** | 2.08.00 |
| ***Fixed in versions:*** | 2.09.00 |

***Problem Description:***

What happens (symptoms):

------------------------------------------------------------------

In the GENy GUI the Rx Data Length of a message is erroneously displayed as 0 for messages that fulfill the conditions described below.

When does this happen:

------------------------------------------------------------------

Configuration time

In which configuration does this happen:

------------------------------------------------------------------

Only in dbc-based configurations if the dbc file contains RxMessages which contain exactly one signal and this signal meets the following conditions:
- Endianess is MOTOROLA
- Signal length is 8 bit
- Signal Start bit is 7

***Resolution Description:***

Workaround:

------------------------------------------------------------------

No workaround available.

Resolution:

------------------------------------------------------------------

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

**ESCAN00062872** **the function CanLL_HandleIllIrptNumber didn't clear a illegal interrupt**

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan_Tms470DcanLl@Implementation |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | 1.17.00 |

***Problem Description:***

What happens (symptoms):

-------------------------------------------------------------------

If an undefined Interrupt occur, this interrupt will not be cleared.

When does this happen:

-------------------------------------------------------------------

If an interrupt with an undefined mailbox number occur. Normally this can not happen.

In which configuration does this happen:

-------------------------------------------------------------------

interrupt configurations

***Resolution Description:***

Workaround:

-------------------------------------------------------------------

No workaround available.

Resolution:

-------------------------------------------------------------------

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

## ESCAN00063756 certain extended IDs may not be received after Full CAN overrun ( if extended ID masking is enabled )

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan_Tms470DcanLl@Implementation |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | 1.18.00 |

***Problem Description:***

What happens (symptoms):

-------------------------------------------------------------------

Some extended IDs may not be received.

When does this happen:

-------------------------------------------------------------------

This can happen after a Rx Full CAN overrun occurs on the assigned Rx Full CAN mailbox.

In which configuration does this happen:

-------------------------------------------------------------------

This occurs only in configurations with
Rx FullCAN messages (C_ENABLE_RX_FULLCAN_OBJECTS is defined in the file can_cfg.h)
AND
Mixed Id (standard and extended Identifier) is used ( C_ENABLE_MIXED_ID is defined in the file can_cfg.h)
AND
extended ID Masking is activated (C_ENABLE_RX_MASK_EXT_ID is defined in the file can_cfg.h)

This happens only with CANbedded CAN Driver.

***Resolution Description:***

Workaround:

-------------------------------------------------------------------

No workaround available.

Resolution:

-------------------------------------------------------------------

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

## ESCAN00070517 Compiler error: missing constant kDescStateSessionDefault

| | |
|---|---|
| ***Component@Subcomponent:*** | Diag_CanDesc__coreBase@Implementation |
| ***First affected version:*** | 1.00.05 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
--------------------------------------------------------------------
Compile error for missing constant kDescStateSessionDefault

When does this happen:
--------------------------------------------------------------------
At compile time.

In which configuration does this happen:
--------------------------------------------------------------------
In the used CDD the name of the default session state is different from "Default".

***Resolution Description:***
Workaround:
--------------------------------------------------------------------
Rename the default session state in the CDD to "Default"

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

## ESCAN00071804 Functions and flags can be added to a Update Bit signal after an dbc update was performed

| | |
|---|---|
| ***Component@Subcomponent:*** | Il_Vector@GenTool_Geny |
| ***First affected version:*** | 1.15.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
--------------------------------------------------------------------
Functions and flags can be added to an Update Bit signal, in the GENy GUI, after an dbc update was performed. Functions and flags should not be available for Update Bit signals.

When does this happen:
--------------------------------------------------------------------
After a dbc update.
As the functions/flags will not be saved and generated this is only relevant for the configuration time.

In which configuration does this happen:
--------------------------------------------------------------------
In a configuration with Update Bits.

***Resolution Description:***
Workaround:
--------------------------------------------------------------------
Save the configuration and reopen the configuration.

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

## ESCAN00073608 "Unknown Service Support" feature in GENy is referenced as "Support Generic User Service" feature

| | |
|---|---|
| ***Component@Subcomponent:*** | Diag_CanDesc__coreBase@Doc_TechRef |
| ***First affected version:*** | 2.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***

What happens (symptoms):

--------------------------------------------------------------------

In the technical reference a "Support Generic User Service" feature is referenced. In the GENy configuration only a "Unknown Service Processing" feature exists.

When does this happen:

--------------------------------------------------------------------

-

In which configuration does this happen:

--------------------------------------------------------------------

-

***Resolution Description:***

Workaround:

--------------------------------------------------------------------

These are two names for the same feature.

Resolution:

--------------------------------------------------------------------

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

## 2.4 Compiler Warnings

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

**Index**

| | |
|---|---|
| ESCAN00022682 | Compiler warning statement not reached in DescUsdtNetIsoTpAssertUser<br>Diag_CanDesc__coreBase@Implementation |
| ESCAN00027751 | Compiler warning for cast to smaller type for "failedByteMask"<br>Diag_CanDesc__coreBase@Implementation |
| ESCAN00033658 | Compiler Warning: W549 condition is always true<br>Nm_IndOsek@Implementation |
| ESCAN00037685 | Compiler Warning: Possible loss of data<br>Nm_IndOsek@Implementation |
| ESCAN00038038 | Compiler warning: SP debug info incorrect because of optimization or inline assembler<br>Cp_Xcp@Implementation |
| ESCAN00044044 | Compiler Warning: condition is always false<br>Cp_Xcp@Implementation |
| ESCAN00044161 | Compiler Warning: Unused Static Function *ValueChanged<br>Il_Vector@GenTool_Geny |
| ESCAN00047283 | IL flags are declared without the "volatile" keyword.<br>Il_Vector@Implementation |
| ESCAN00048020 | Compiler warning: Deprecated use of PSR; flag bits not specified, "cf" assumed<br>VStdLib_Arm7@Implementation |
| ESCAN00057831 | Compiler warning: "canCanInterruptOldStatus" was declared but never referenced<br>DrvCan__coreHll@Implementation |
| ESCAN00057832 | Compiler warning: "canCanInterruptCounter" was set but never referenced<br>DrvCan__coreHll@Implementation |
| ESCAN00058586 | Compiler warning: comparison is always true due to limited range of data type<br>DrvCan__coreHll@Implementation |
| ESCAN00059701 | Compiler warning: condition is always true" in the IlTxTimerTask, IlTxStateTask and IlTxRepetitionsAreActive<br>Il_Vector@Implementation |
| ESCAN00059736 | Compiler warning: pointless comparison of unsigned integer with zero<br>DrvCan__coreHll@Implementation |
| ESCAN00061505 | Compiler warning:  ApplCanMsgReceived: prior identical declaration -- ignored<br>DrvCan__base@GenTool_Geny |
| ESCAN00061617 | Compiler warning: warning C4244: '=' : conversion from 'DescDynDidMemBlockSize' to 'vuint16', possible loss of data<br>Diag_CanDesc__coreBase@Implementation |
| ESCAN00066833 | Compiler warning: Redefined macro name when compiling a main GENy project with a sub-project<br>GenTool_GenyDriverBase@GenTool_Geny |

## ESCAN00022682 Compiler warning statement not reached in DescUsdtNetIsoTpAssertUser

| | |
|---|---|
| ***Component@Subcomponent:*** | Diag_CanDesc__coreBase@Implementation |
| ***First affected version:*** | 4.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens :
---------------------------------------------------------------
Compiler warning "statement not reached" in
DescUsdtNetIsoTpAssertUser((TP_CHANNEL_TX_PARAM_VALUE != kTpNoChannel),
kDescNetAssertWrongIsoTpTxChannel);

When does this happen:
------------------------------------------------------------------
At compile time

In which configuration does this happen:
-----------------------------------------------------------------
- CANdesc/CANdescBasic
AND
- Debug support has been activated
AND
- Static Multi TPMC has been configured

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
This warning can be ignored since there is no danger for the software normal functioning.

Resolution:
-------------------------------------------------------------------
The described issue is corrected by modification of all affected workproducts.

## ESCAN00027751 Compiler warning for cast to smaller type for "failedByteMask"

| | |
|---|---|
| ***Component@Subcomponent:*** | Diag_CanDesc__coreBase@Implementation |
| ***First affected version:*** | 3.01.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
Compiler warning message for the assignment:

*failedByteMask = (vuint8)(0x02 << *failedByteMask);

But there is no real danger of losing information by casting down to a smaller type since the code generator does not allow more than 7 (seven) sub-service bytes in the request message. So skipping the SID (bit 0) does not lead to losing the MSB and the value of the failedByteMask cannot be greater than six.

When does this happen:
-------------------------------------------------------------------
At compile time.

In which configuration does this happen:
-------------------------------------------------------------------
-CANdesc/CANdescBasic

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
Ignore the warning

Resolution:
-------------------------------------------------------------------
This ESCAN will not be resolved, since the fix might require more resources on the ECU. The code generator assures that there will be no overflow on the shift operation.

## ESCAN00033658 Compiler Warning: W549 condition is always true

|  |  |
|---|---|
| ***Component@Subcomponent:*** | Nm_IndOsek@Implementation |
| ***First affected version:*** | 2.13.00 |
| ***Fixed in versions:*** | 3.01.01 |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
There is a compiler warning about a condition that is always true.

When does this happen:
-------------------------------------------------------------------
This issue occurs at compile time.

In which configuration does this happen:
-------------------------------------------------------------------
This issue occurs-
- if the compiler and the configured warning level checks for conditions that are always true.
and
- if #define INM_ENABLE_CLEAR_COUNTER is set

Note: This issue was found with Tasking-Compiler v2.2r3

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
No workaround available.

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

The condition was replaces with the following code.

```
 #if defined ( INM_ENABLE_CLEAR_COUNTER )
 /* ESCAN00033658 */ /* check is not necessary, as counter is always 0 */
 #else
 if( pEvent->counter == 0 )
 #endif
```

| ESCAN00037685 | Compiler Warning: Possible loss of data |
|---|---|
| ***Component@Subcomponent:*** | Nm_IndOsek@Implementation |
| ***First affected version:*** | 2.14.00 |
| ***Fixed in versions:*** | 3.02.00 |

***Problem Description:***
What happens (symptoms):
--------------------------------------------------------------------
There is a compiler warning about possible loss of data.

When does this happen:
--------------------------------------------------------------------
This issue occurs at compile time.

In which configuration does this happen:
--------------------------------------------------------------------
- The compiler warning was found with a Metrowerks compiler
- The issue does not depend on the configuration.

***Resolution Description:***
Workaround:
--------------------------------------------------------------------
No workaround available.

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

## ESCAN00038038 Compiler warning: SP debug info incorrect because of optimization or inline assembler

| | |
|---|---|
| ***Component@Subcomponent:*** | Cp_Xcp@Implementation |
| ***First affected version:*** | 1.25.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
--------------------------------------------------------------------
Compiler warning by using the Metrowerks compiler V4.5

When does this happen:
--------------------------------------------------------------------
This occur always during compilation

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.

**ESCAN00044044 Compiler Warning: condition is always false**

| | |
|---|---|
| ***Component@Subcomponent:*** | Cp_Xcp@Implementation |
| ***First affected version:*** | 1.26.02 |
| ***Fixed in versions:*** | |

***Problem Description:***

What happens (symptoms):

```
--------------------------------------------------------------------
Compiling file: ../../BSW/Xcp/xcpProf.c
0 errors, 5 warnings
ctc W549: ["../../BSW/Xcp/xcpProf.c" 2518/22] condition is always false
ctc W549: ["../../BSW/Xcp/xcpProf.c" 2522/22] condition is always false
ctc W549: ["../../BSW/Xcp/xcpProf.c" 2526/22] condition is always false
ctc W549: ["../../BSW/Xcp/xcpProf.c" 2659/22] condition is always false
ctc W549: ["../../BSW/Xcp/xcpProf.c" 2663/22] condition is always false
0 errors, 5 warnings
```

When does this happen:

--------------------------------------------------------------------

This happens when XCP_DISABLE_WRITE_PROTECTION and XCP_DISABLE_WRITE_EEPROM are defined.

In which configuration does this happen:

--------------------------------------------------------------------

see above

***Resolution Description:***

Workaround:

--------------------------------------------------------------------

Enable
XCP_DISABLE_WRITE_PROTECTION or XCP_DISABLE_WRITE_EEPROM

Resolution:

--------------------------------------------------------------------

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

## ESCAN00044161 Compiler Warning: Unused Static Function *ValueChanged

| | |
|---|---|
| ***Component@Subcomponent:*** | Il_Vector@GenTool_Geny |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***

What happens (symptoms):
--------------------------------------------------------------------
The compiler warning unused static function occurs in il_par.c for *ValueChanged functions.

When does this happen:
--------------------------------------------------------------------
At compile time.

In which configuration does this happen:
--------------------------------------------------------------------
Any configuration, where Tx Signal > 32 Bit are used with the dbc GenSigSendType "OnChange" or "OnChangeWithRepetition" and the Put Signal Access is deactivated.

Using the Tasking Compiler V3_2r3.

***Resolution Description:***

Workaround:
--------------------------------------------------------------------
Activate the Put Signal Access.

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

## ESCAN00047283 IL flags are declared without the "volatile" keyword.

| | |
|---|---|
| ***Component@Subcomponent:*** | Il_Vector@Implementation |
| ***First affected version:*** | 3.10.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
IL flags (Indication, FirstValue, Confirmation, Timeout) are declared without the "volatile" keyword. Read and Write access to IL flags has no effect due to a Read-Modify-Write problematic.

e.g.
FlagA and FlagB are in the same byte and set on interrupt level

this sequence is executed on task level:

```
disable int;
clear FlagA; /*1*/
enable int;
... /*3*/
disable int;
clear FlagB; /*2*/
enable int;
```

The compiler might optimize this sequence and the flag read and write ALWAYS fails:

read the byte at 1), modify the local copy and write the byte at 2)
if the byte is written on interrupt level at 3), the data is lost.

When does this happen:
-------------------------------------------------------------------
At runtime (This Problem has been found by a review and has never been detected in a ECU)

In which configuration does this happen:
-------------------------------------------------------------------
- This issue highly depends on the used compiler and compiler options.
- Preemptive IL flag access is used (e.g. interrupt system)

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
Review the optimization configuration of your compiler.

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

|  |  |
|---|---|
| **ESCAN00048020** | **Compiler warning: Deprecated use of PSR; flag bits not specified, "cf" assumed** |
| ***Component@Subcomponent:*** | VStdLib_Arm7@Implementation |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** |  |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
Following warnings occur:

[W0000] Deprecated use of PSR; flag bits not specified, "cf" assumed: msr CPSR, r1

The warning can be ignored, because the implementation of the vstdlib handle the missing of "_cf".

When does this happen:
-------------------------------------------------------------------
The warning occurred during compile time

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

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
No workaround available.

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

## ESCAN00057831 Compiler warning: "canCanInterruptOldStatus" was declared but never referenced

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan__coreHll@Implementation |
| ***First affected version:*** | 2.07.00 |
| ***Fixed in versions:*** | 2.15.02 |

***Problem Description:***

What happens (symptoms):

--------------------------------------------------------------------

Compiler warns for a unused declaration of an variable which is not used in a special configuration: Can be accepted

When does this happen:

--------------------------------------------------------------------

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

In which configuration does this happen:

--------------------------------------------------------------------

Configurations which declare "#define C_DISABLE_CAN_CAN_INTERRUPT_CONTROL" in can_cfg.h, means that CAN-driver did not interrupt-control by himself -> FlashBootLoader.

***Resolution Description:***

Workaround:

--------------------------------------------------------------------

The warning can be ignored.

Resolution:

--------------------------------------------------------------------

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

| ESCAN00057832 | Compiler warning: "canCanInterruptCounter" was set but never referenced |
|---|---|
| ***Component@Subcomponent:*** | DrvCan__coreHll@Implementation |
| ***First affected version:*** | 2.07.00 |
| ***Fixed in versions:*** | 2.15.02 |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
Compiler warns for an unused variable which is not used in a special configuration: Can be accepted

When does this happen:
-------------------------------------------------------------------
The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:
-------------------------------------------------------------------
Configurations which declare "#define C_DISABLE_CAN_CAN_INTERRUPT_CONTROL", means that CAN-driver did not interrupt-control by himself -> FlashBootLoader.

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
The warning can be ignored.

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

## ESCAN00058586 Compiler warning: comparison is always true due to limited range of data type

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan__coreHll@Implementation |
| ***First affected version:*** | 2.00.00 |
| ***Fixed in versions:*** | 2.15.01 |

***Problem Description:***
What happens (symptoms):
------------------------------------------------------------------
Compiler warns that comparison is always true in following code:
assertUser((CanSignedRxHandle)rxObjHandle >= (CanSignedRxHandle)CAN_HL_HW_RX_BASIC_STARTINDEX(canHwChannel), channel, kErrorHwHdlTooSmall);

There is no impact during runtime. The warning can be ignored.

When does this happen:
------------------------------------------------------------------
The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:
------------------------------------------------------------------
This happens if
- user assertions are enabled (C_ENABLE_USER_CHECK is defined in can_cfg.h)
AND
- single receive channel is used (C_SINGLE_RECEIVE_CHANNEL)
AND
- individual polling is enabled (C_ENABLE_INDIVIDUAL_POLLING is defined in can_cfg.h; requires high end license)
AND
- at least one Rx BasicCAN object is present in the configuration (C_ENABLE_RX_BASICCAN_OBJECTS is defined in can_cfg.h)
AND
- at least one Rx BasicCAN object is configured to be processed by polling (C_ENABLE_RX_BASICCAN_POLLING is defined in can_cfg.h)
AND
- kCanHwRxBasicStartIndex is 0 (this is platform dependent)

Detected with DrvCan__Sh2RcanHll and "sh-elf-gcc.exe (GCC) 4.2-GNUSH_v0703"

***Resolution Description:***
Workaround:
------------------------------------------------------------------
The warning can be ignored

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

**ESCAN00059701** **Compiler warning: condition is always true" in the IlTxTimerTask, IlTxStateTask and IlTxRepetitionsAreActive**

| | |
|---|---|
| ***Component@Subcomponent:*** | Il_Vector@Implementation |
| ***First affected version:*** | 2.42.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
Compiler warns for "condition is always true" in the IlTxTimerTask, IlTxStateTask and IlTxRepetitionsAreActive API. This may happen depending on the configuration.

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:
-------------------------------------------------------------------
IlTxTimerTask, IlTxStateTask: Any configuration with exactly one tx message.
IlTxRepetitionsAreActive: Any configuration with exactly one tx message and the API is configured. (IL_ENABLE_SYS_TX_REPETITIONS_ARE_ACTIVE_FCT must be defined)

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 rare configuration. The code uses a while loop with a counter and can probably replaced by a for loop, but other compilers or codeanalysers may warn about a useless loop. The code exists for about 15 Years and will not be changed.

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
No workaround available.

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

## ESCAN00059736 Compiler warning: pointless comparison of unsigned integer with zero

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan__coreHll@Implementation |
| ***First affected version:*** | 2.00.00 |
| ***Fixed in versions:*** | 2.15.01 |

***Problem Description:***
What happens (symptoms):
--------------------------------------------------------------------
Compiler warns that comparison is always true in following code:
assertUser((CanSignedRxHandle)rxObjHandle >= (CanSignedRxHandle)CAN_HL_HW_RX_FULL_STARTINDEX(canHwChannel), channel, kErrorHwHdlTooSmall);

There is no impact during runtime. The warning can be ignored.

When does this happen:
--------------------------------------------------------------------
The warning is issued by the compiler during compilation of the code in case the configuration is as described below.

In which configuration does this happen:
--------------------------------------------------------------------
This happens if
- user assertions are enabled (C_ENABLE_USER_CHECK is defined in can_cfg.h)
AND
- single receive channel is used (C_SINGLE_RECEIVE_CHANNEL)
AND
- individual polling is enabled (C_ENABLE_INDIVIDUAL_POLLING is defined in can_cfg.h; requires high end license)
AND
- at least one Rx FullCAN object is present in the configuration (C_ENABLE_RX_FULLCAN_OBJECTS is defined in can_cfg.h)
AND
- at least one Rx FullCAN object is configured to be processed by polling (C_ENABLE_RX_FULLCAN_POLLING is defined in can_cfg.h)
AND
- kCanHwRxFullStartIndex is 0 (this is platform dependent)

Detected with DrvCan_Mpc5500Flexcan2Hll and GHS v6.1.0

***Resolution Description:***
Workaround:
--------------------------------------------------------------------
No workaround available.

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

## ESCAN00061505 Compiler warning: ApplCanMsgReceived: prior identical declaration -- ignored

| | |
|---|---|
| ***Component@Subcomponent:*** | DrvCan__base@GenTool_Geny |
| ***First affected version:*** | 3.18.01 |
| ***Fixed in versions:*** | |

***Problem Description:***

What happens (symptoms):

------------------------------------------------------------------

Compiler warns for duplicated function prototype: Can be accepted

When does this happen:

------------------------------------------------------------------

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

In which configuration does this happen:

------------------------------------------------------------------

One CAN channel is used (C_SINGLE_RECEIVE_CHANNEL is defined)
AND
Feature "Rx notification" is enabled (C_ENABLE_RECEIVE_FCT is defined)
AND
CANbedded CAN driver with Reference Implementation 1.4 or lower

Hint: This will not be resolved for RI 1.4 or lower. The warning can be ignored.

***Resolution Description:***

Workaround:

------------------------------------------------------------------

The warning can be ignored.

Resolution:

------------------------------------------------------------------

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

| ESCAN00061617 | Compiler warning: warning C4244: '=' : conversion from 'DescDynDidMemBlockSize' to 'vuint16', possible loss of data |
|---|---|
| ***Component@Subcomponent:*** | Diag_CanDesc__coreBase@Implementation |
| ***First affected version:*** | 1.00.00 |
| ***Fixed in versions:*** | |

***Problem Description:***
What happens (symptoms):
----------------------------------------------------------------------
desc.c(8368): warning C4244: '=' : conversion from 'DescDynDidMemBlockSize' to 'vuint16', possible loss of data
desc.c(8371): warning C4244: '=' : conversion from 'DescDynDidMemBlockSize' to 'DescMsgLen', possible loss of data
desc.c(8424): warning C4244: '+=' : conversion from 'DescDynDidMemBlockSize' to 'DescMsgLen', 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:
----------------------------------------------------------------------
constant FID (Format IDentifier) configured in cdd file
High Nibble of the FID (Length of the memory size parameter) is bigger than 2
=> no problem for the CAN use case (when CANdesc is used only with CAN)

***Resolution Description:***
Workaround:
----------------------------------------------------------------------
No workaround available.

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

## ESCAN00066833 Compiler warning: Redefined macro name when compiling a main GENy project with a sub-project

| | |
|---|---|
| ***Component@Subcomponent:*** | GenTool_GenyDriverBase@GenTool_Geny |
| ***First affected version:*** | 2.00.00 |
| ***Fixed in versions:*** | 2.10.00 |

***Problem Description:***
What happens (symptoms):
-------------------------------------------------------------------
Compiler generates warning on the following macro redefinitions:

```
v_cfg_1.h(180) : CC78K0R warning W0816: Redefined macro name
'V_ATOMIC_BIT_ACCESS_IN_BITFIELD'
v_cfg_1.h(181) : CC78K0R warning W0816: Redefined macro name
'V_ATOMIC_VARIABLE_ACCESS'
v_cfg_1.h(196) : CC78K0R warning W0816: Redefined macro name 'kComNumberOfNodes'
v_cfg_1.h(197) : CC78K0R warning W0816: Redefined macro name 'ComSetCurrentECU'
v_cfg_1.h(198) : CC78K0R warning W0816: Redefined macro name 'comMultipleECUCurrent'
```

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 two GENy projects are compiled together (e.g. CAN, LIN), one is setup as "main project" and the other is setup as "sub-project"

***Resolution Description:***
Workaround:
-------------------------------------------------------------------
No workaround available.

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

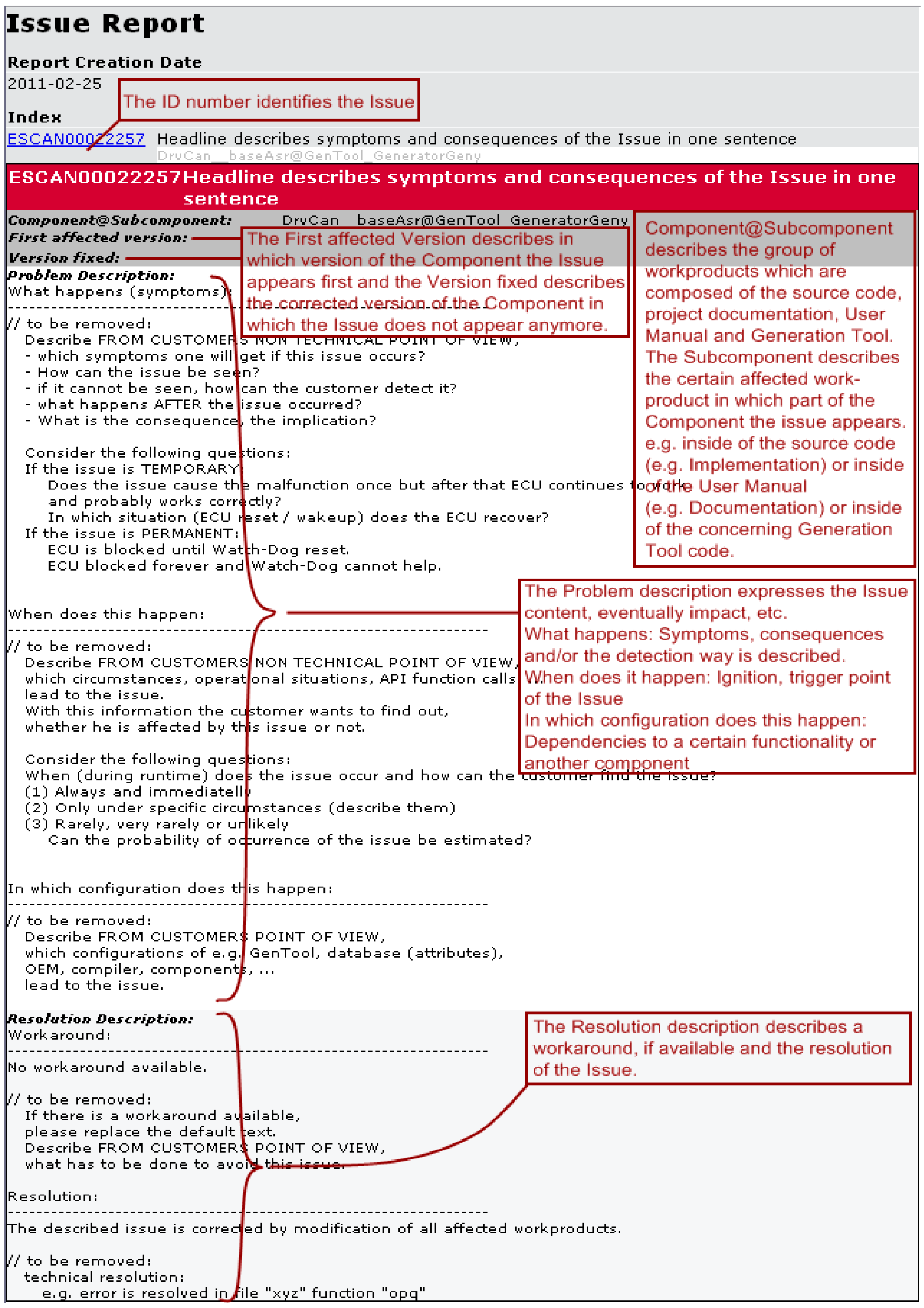
## 3. New Issues for Information

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

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

No issue to be reported.

## 4. Report Legend



**Issue Report**

**Report Creation Date**
2011-02-25

**Index**
ESCAN00022257 Headline describes symptoms and consequences of the Issue in one sentence
DrvCan__baseAsr@GenTool_GeneratorGeny

**ESCAN00022257Headline describes symptoms and consequences of the Issue in one sentence**

***Component@Subcomponent:*** DrvCan__baseAsr@GenTool_GeneratorGeny
***First affected version:***
***Version fixed:***
***Problem Description:***

```
What happens (symptoms):
-----------------------------------------------------------------
// to be removed:
   Describe FROM CUSTOMERS NON TECHNICAL POINT OF VIEW,
   - which symptoms one will get if this issue occurs?
   - How can the issue be seen?
   - if it cannot be seen, how can the customer detect it?
   - what happens AFTER the issue occurred?
   - What is the consequence, the implication?

   Consider the following questions:
   If the issue is TEMPORARY:
       Does the issue cause the malfunction once but after that ECU continues to work
       and probably works correctly?
       In which situation (ECU reset / wakeup) does the ECU recover?
   If the issue is PERMANENT:
       ECU is blocked until Watch-Dog reset.
       ECU blocked forever and Watch-Dog cannot help.


When does this happen:
-----------------------------------------------------------------
// to be removed:
   Describe FROM CUSTOMERS NON TECHNICAL POINT OF VIEW,
   which circumstances, operational situations, API function calls
   lead to the issue.
   With this information the customer wants to find out,
   whether he is affected by this issue or not.

   Consider the following questions:
   When (during runtime) does the issue occur and how can the customer find the issue?
   (1) Always and immediatelly
   (2) Only under specific circumstances (describe them)
   (3) Rarely, very rarely or unlikely
       Can the probability of occurrence of the issue be estimated?


In which configuration does this happen:
-----------------------------------------------------------------
// to be removed:
   Describe FROM CUSTOMERS POINT OF VIEW,
   which configurations of e.g. GenTool, database (attributes),
   OEM, compiler, components, ...
   lead to the issue.
```

***Resolution Description:***

```
Workaround:
-----------------------------------------------------------------
No workaround available.

// to be removed:
   If there is a workaround available,
   please replace the default text.
   Describe FROM CUSTOMERS POINT OF VIEW,
   what has to be done to avoid this issue.

Resolution:
-----------------------------------------------------------------
The described issue is corrected by modification of all affected workproducts.

// to be removed:
   technical resolution:
      e.g. error is resolved in file "xyz" function "opq"
```

The ID number identifies the Issue

The First affected Version describes in which version of the Component the Issue appears first and the Version fixed describes the corrected version of the Component in which the Issue does not appear anymore.

Component@Subcomponent describes the group of workproducts which are composed of the source code, project documentation, User Manual and Generation Tool. The Subcomponent describes the certain affected work-product in which part of the Component the issue appears. e.g. inside of the source code (e.g. Implementation) or inside of the User Manual (e.g. Documentation) or inside of the concerning Generation Tool code.

The Problem description expresses the Issue content, eventually impact, etc.
What happens: Symptoms, consequences and/or the detection way is described.
When does it happen: Ignition, trigger point of the Issue
In which configuration does this happen: Dependencies to a certain functionality or another component

The Resolution description describes a workaround, if available and the resolution of the Issue.

## 5. Quality Management Contact

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

Vector Informatik GmbH
Ingersheimer Str. 24
D-70499 Stuttgart

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