

<b>Document Title</b>	SWS_LINDriver: Complete Change Documentation 4.3.0 - 4.3.1
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	695

  

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	4.3.1

## Table of Contents

1	SWS_LINDriver	3
1.1	Specification Item SWS_Lin_00099	3
1.2	Specification Item SWS_Lin_00105	6
1.3	Specification Item SWS_Lin_00107	9
1.4	Specification Item SWS_Lin_00129	13
1.5	Specification Item SWS_Lin_00131	16
1.6	Specification Item SWS_Lin_00133	20
1.7	Specification Item SWS_Lin_00135	23
1.8	Specification Item SWS_Lin_00137	26
1.9	Specification Item SWS_Lin_00139	30
1.10	Specification Item SWS_Lin_00140	33
1.11	Specification Item SWS_Lin_00141	36
1.12	Specification Item SWS_Lin_00143	40
1.13	Specification Item SWS_Lin_00144	43
1.14	Specification Item SWS_Lin_00168	47
1.15	Specification Item SWS_Lin_00171	51
1.16	Specification Item SWS_Lin_00191	54
1.17	Specification Item SWS_Lin_00195	66
1.18	Specification Item SWS_Lin_00197	70
1.19	Specification Item SWS_Lin_00198	73
1.20	Specification Item SWS_Lin_00199	77
1.21	Specification Item SWS_Lin_00213	80
1.22	Specification Item SWS_Lin_00214	83
1.23	Specification Item SWS_Lin_00215	87
1.24	Specification Item SWS_Lin_00216	90
1.25	Specification Item SWS_Lin_00218	93
1.26	Specification Item SWS_Lin_00237	97
1.27	Specification Item SWS_Lin_00248	100
1.28	Specification Item SWS_Lin_00249	104
1.29	Specification Item SWS_Lin_00251	107
1.30	Specification Item SWS_Lin_00258	110
1.31	Specification Item SWS_Lin_00259	114
1.32	Specification Item SWS_Lin_00260	117

# 1 SWS\_LINDriver

## 1.1 Specification Item SWS\_Lin\_00099

### Trace References:

none

### Content:

If **default development** error detection for the Lin module is enabled: the function Lin\_Init shall check the parameter Config for being within the allowed range. If Config is not in the allowed range, the function Lin\_Init shall raise the **default development** error LIN\_E\_INVALID\_POINTER.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

– SWS\_RTE –

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

**SRS\_Libraries:**

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_SPALGeneral:**

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_FlashTest:**

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_MFXLibrary:**

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SWS\_MemoryAbstractionInterface:**

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_FlexRayNetworkManagement:**

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_CANStateManager:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"
- Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.2 Specification Item SWS\_Lin\_00105

#### Trace References:

none

#### Content:

If **default development** error detection for the Lin module is enabled: the function Lin\_Init shall check the Lin driver for being in the state LIN\_UNINIT. If the Lin driver is not in the state LIN\_UNINIT, the function Lin\_Init shall raise the **default development** error LIN\_E\_STATE\_TRANSITION.

#### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS\_Rte\_07676]
- Change [SWS\_RTE\_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS\_Rte\_06631]  
[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"



#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.3 Specification Item SWS\_Lin\_00107

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: if the function `Lin_CheckWakeup` is called before the LIN module was initialized, the function `Lin_CheckWakeup` shall raise the **default development** error `LIN_E_UNINIT`.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters `*DevErrorDetect` are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_XXX\_YYYYY]

In case default error detection is enabled for the XXXX module: The XXXX module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_XXX\_YYYYY]

If default error detection is enabled: the function shall check that the service `xxx_Init` was previously called. If the check fails, the function shall raise the default error `XXX_E_NOT_INITIALIZED` otherwise (if DET is disabled) return `E_NOT_OK`. ()"

- "In case default errors are enabled,..."

- "module raises the Default error `XXX_E_TRANSITION`"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.4 Specification Item SWS\_Lin\_00129

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: if the function Lin\_GoToSleep is called before the LIN module was initialized, the function Lin\_GoToSleep shall raise the **default development** error LIN\_E\_UNINIT.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development

error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
–Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.5 Specification Item SWS\_Lin\_00131

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: the function Lin\_GoToSleep shall raise the **default development** error LIN\_E\_INVALID\_CHANNEL if the channel parameter is invalid.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

### Problem description:

The DET was renamed from development error tracer to default error tracer.



This change was most of the time done automatically and unfortunately re-named "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

### **Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service

xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX\_E\_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS\_Rte\_07676]
- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS\_Rte\_06631]  
[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.6 Specification Item SWS\_Lin\_00133

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: if the function Lin\_GoToSleepInternal is called before the LIN module was initialized, the function Lin\_GoToSleepInternal shall raise the **default development** error LIN\_E\_UNINIT.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
–Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.7 Specification Item SWS\_Lin\_00135

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: the function Lin\_GoToSleepInternal shall raise the **default development** error LIN\_E\_INVALID\_CHANNEL if the channel parameter is invalid.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —



- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS\_Rte\_07676]
- Change [SWS\_RTE\_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS\_Rte\_06631]  
[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

#### SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"  
 Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"  
 Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents":  
 Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents":  
 Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents":  
 Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
 –Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.8 Specification Item SWS\_Lin\_00137

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: if the function Lin\_Wakeup is called before the LIN module was initialized, the function Lin\_Wakeup shall raise the **default development** error LIN\_E\_UNINIT.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]"

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_FlashTest:**

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_MFXLibrary:**

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SWS\_MemoryAbstractionInterface:**

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_FlexRayNetworkManagement:**

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_CANStateManager:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_PDURouter:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_EEPROMDriver:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.9 Specification Item SWS\_Lin\_00139

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: the function Lin\_Wakeup shall raise the **default development** error LIN\_E\_INVALID\_CHANNEL if the channel parameter is invalid or the channel is inactive.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance

Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

#### SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"



SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
 –Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.10 Specification Item SWS\_Lin\_00140

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: the function Lin\_Wakeup shall raise the **default development** error LIN\_E\_STATE\_TRANSITION if the LIN channel state-machine is not in the state LIN\_CH\_SLEEP.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX\_E\_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —
  - Change 4.8 Default errors to 4.8 Development errors
  - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
  - Remove [SWS\_Rte\_07676]
  - Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
  - Change [SWS\_Rte\_06631]
- [SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
  - In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"
- |   |    |                                  |
|---|----|----------------------------------|
| Rename "AUTOSAR_SWS_DevelopmentErrorTracer" | to | "AUTOSAR_SWS_DefaultErrorTracer" |
|---|----|----------------------------------|

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.11 Specification Item SWS\_Lin\_00141

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: if the function `Lin_GetStatus` is called before the LIN module was initialized, the function `Lin_GetStatus` shall raise the **default development** error `LIN_E_UNINIT` **and otherwise (if DET is disabled)** return `LIN_NOT_OK`.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters `*DevErrorDetect` are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]"

If default error detection is enabled: the function shall check that the service `xxx_Init` was previously called. If the check fails, the function shall raise the default error `XXX_E_NOT_INITIALIZED` otherwise (if DET is disabled) return `E_NOT_OK`. ()"

- "In case default errors are enabled,..."
- "module raises the Default error XXX\_E\_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX\_E\_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS\_Rte\_07676]
- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS\_Rte\_06631]
- [SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_SPALGeneral:**

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_FlashTest:**

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_MFXLibrary:**

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SWS\_MemoryAbstractionInterface:**

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_FlexRayNetworkManagement:**

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_CANStateManager:**

- In chapter "3.1 Input documents":  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_PDURouter:**

- In chapter "3.1 Input documents":  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.12 Specification Item SWS\_Lin\_00143

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: if the channel parameter is invalid or the channel is inactive, the function Lin\_GetStatus shall raise the **default development** error LIN\_E\_INVALID\_CHANNEL **and otherwise (if DET is disabled)** return LIN\_NOT\_OK.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET



does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename	"AUTOSAR_SWS_DevelopmentErrorTracer"	to	"AU-
	TOSAR_SWS_DefaultErrorTracer"		

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename	"AUTOSAR_SWS_DevelopmentErrorTracer"	to	"AU-
	TOSAR_SWS_DefaultErrorTracer"		

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
 –Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.13 Specification Item SWS\_Lin\_00144

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: the function Lin\_GetStatus shall check the parameter Lin\_SduPtr for not being a NULL pointer. If Lin\_SduPtr is a NULL pointer, the function Lin\_GetStatus shall raise the **default development** error LIN\_E\_PARAM\_POINTER **and otherwise (if DET is disabled)** return LIN\_NOT\_OK.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyy]"

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to

**"Default Error Tracer"**

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_MFXLibrary:**

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SWS\_MemoryAbstractionInterface:**

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_FlexRayNetworkManagement:**

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_CANStateManager:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_PDURouter:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_EEPROMDriver:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.14 Specification Item SWS\_Lin\_00168

### Trace References:

none

### Content:

Service name:	Lin_GetStatusLin_GetStatus	
Syntax:	Lin_StatusType Lin_GetStatus( uint8 Channel, uint8** Lin_SduPtr )	
Service ID[hex]:	0x08	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	ChannelLin_GetStatus.Channel	LIN channel to be checked
Parameters (inout):	None	
Parameters (out):	Lin_SduPtrLin_GetStatus.Lin_SduPtr	Pointer to pointer to a shadow buffer or memory mapped LIN Hardware receive buffer where the current SDU is stored.
Return value:	Lin_StatusType	<p>LIN_NOT_OK: Development or production error occurred</p> <p>LIN_TX_OK: Successful transmission</p> <p>LIN_TX_BUSY: Ongoing transmission (Header or Response)</p> <p>LIN_TX_HEADER_ERROR: Erroneous header transmission such as:</p> <ul style="list-style-type: none"> <li>- Mismatch between sent and read back data</li> <li>- Identifier parity error or Physical bus error</li> </ul> <p>LIN_TX_ERROR: Erroneous response transmission such as:</p> <ul style="list-style-type: none"> <li>- Mismatch between sent and read back data</li> <li>- Physical bus error</li> </ul> <p>LIN_RX_OK: Reception of correct response</p> <p>LIN_RX_BUSY: Ongoing reception: at least one response byte has been received, but the checksum byte has not been received</p> <p>LIN_RX_ERROR: Erroneous response reception such as:</p> <ul style="list-style-type: none"> <li>- Framing error</li> <li>- Overrun error</li> <li>- Checksum error or Short response</li> </ul> <p>LIN_RX_NO_RESPONSE: No response byte has been received so far</p> <p>LIN_OPERATIONAL: Normal operation; the related LIN channel is <b>just initialized or waked woken</b> up from the LIN_CH_SLEEP and no data has been sent.</p> <p>LIN_CH_SLEEP: Sleep state operation; in this state wake-up detection from slave nodes is enabled.</p>
Description:	Gets the status of the LIN driver.	

## RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73095: [LinSM][LinIf][Lin] Inconsistent LIN channel states upon initialization

### Problem description:

As per SWS\_Lin\_00171, LIN channel state upon initialization shall be LIN\_CH\_OPERATIONAL

As per SWS\_LinIf\_00507, LinIf channel state upon initialization shall be either LINIF\_CHANNEL\_OPERATIONAL or LINIF\_CHANNEL\_SLEEP depending on the configuration parameter LinIfStartupState

As per SWS\_LinSM\_00152, LinSM state for all networks upon initialization shall be LINSM\_NO\_COM

This is an inconsistent state within the LIN Stack since LIN driver channel is operational but LinSM is in NO\_COM and LinIf state could be either!

Tracking the history of LinIf (Bug 27547) it seems that when LinIfStartupState was introduced, then Bug 27547 comment 8 was not addressed at that time to avoid this inconsistency.

Is there a use case to start the LIN channels in OPERATIONAL?

Should the LinSM already transition the channels to SLEEP following initialization? (similar to CanSM)

Or should the Lin driver specifications be updated to allow for either state following initialization (similar to LinIf and LinTrcv specifications)

### Agreed solution:

=====  
For CP R4.3.1  
=====

### SRS LIN

=====

1) [SRS\_Lin\_01514], Description: Change  
from

<The LIN Interface shall inform an upper layer if a wake-up request was notified by the underlying LIN Driver

<A wakeup notification shall only be raised in the case a go-to-sleep-command has been issued to the bus.

to

>The LIN Interface shall inform an upper layer if a wake-up request was notified by



the underlying LIN Driver.

## SWS LinIf

=====

1) Sec. 4.1 Limitations: Add following description

+It's assumed that all of connected LIN Slave ECUs can receive a wakeup frame when they are already operational (as LIN Master ECU starts with LINIF\_CHANNEL\_SLEEP state).

2) Chap. 7, [SWS\_LinIf\_00507]: Change

from

<[SWS\_LinIf\_00507] The LIN Interface shall transit from LINIF\_UNINIT to

< LINIF\_CHANNEL\_OPERATIONAL when the function LinIf\_Init is called and the configuration parameter LinIfStartupState is set to NORMAL.

< LINIF\_CHANNEL\_SLEEP when the function LinIf\_Init is called and the configuration parameter LinIfStartupState is set to SLEEP. ()

to

>[SWS\_LinIf\_00507] The LIN Interface shall transit from LINIF\_UNINIT to LINIF\_CHANNEL\_SLEEP without sending go-to-sleep command, when the function LinIf\_Init is called. ()

>Note: it is assumed that automatically slave nodes will enter bus sleep mode earliest after 4s and latest 10s of bus inactivity (as specified in LIN 2.1).

3) Chap. 7, Figure 4 (stm LinIf Node Management): Update the figure as described below

-Remove the transition to LINIF\_CHANNEL\_OPERATIONAL with LinIf\_Init [LinIfStartupState=NORMAL]

-Remove the transition to LINIF\_CHANNEL\_SLEEP with LinIf\_Init [LinIfStartupState=SLEEP]

+Add the initial transition to LINIF\_CHANNEL\_SLEEP inside the state LINIF\_INIT

4) Chap. 10, ECUC\_LinIf\_00069: Set the parameter LinIfStartupState to obsolete

5) Chap. 10, Figure 20: Remove the parameter LinIfStartupState

## SWS LinTrcv

=====

### Chapter 7.1

Update figure: remove transitions from NOT\_ACTIVE to LIN-TRCV\_TRCV\_MODE\_STANDBY and LINTRCV\_TRCV\_MODE\_NORMAL.

Remove the following sentence:

The function LinTrcv\_Init() causes a state change to either LIN-TRCV\_TRCV\_MODE\_SLEEP, LINTRCV\_TRCV\_MODE\_NORMAL or LIN-TRCV\_TRCV\_MODE\_STANDBY (any of these 3 states belong to the upper state ACTIVE). This depends on the configuration and is independent configurable for each channel.

Change SWS\_LinTrcv\_00119 from

The function LinTrcv\_Init shall set the LIN transceiver hardware to the state configured by the configuration parameter LinTrcvInitState.

This can be LINTRCV\_TRCV\_MODE\_NORMAL, LIN-TRCV\_TRCV\_MODE\_STANDBY or LINTRCV\_TRCV\_MODE\_SLEEP.  
to

The function LinTrcv\_Init shall set the LIN transceiver hardware to the state LIN-TRCV\_TRCV\_MODE\_SLEEP.

Remove SWS\_LinTrcv\_00146.

remove configurations note after SWS\_LinTrcv\_00146

set ECUC\_LinTrcv\_00005 to obsolete.

## SWS Lin

=====

- Change SWS\_Lin\_00171 to: "On entering the state LIN\_INIT, the Lin module shall set each channel into state LIN\_CH\_SLEEP, enable the wake-up detection (if enabled by LinChannelWakeupSupport), and optionally set the LIN hardware unit to reduced power operation mode (if supported by HW)."

- Change initial transition in figure 7-1 to LIN\_CH\_SLEEP.

- SWS\_Lin\_00168 Lin\_GetStatus: Change LIN\_OPERATIONAL description to "Normal operation; the related LIN channel is woken up from the LIN\_CH\_SLEEP and no data has been sent."

## SWS LinSM

=====

no changes required

## TPS System Template

=====

1) Appendix C: Remove the rule of the parameter LinIfStartupState

–Last change on issue 73095 comment 46–

**BW-C-Level:**

Application	Specification	Bus
1	4	4

## 1.15 Specification Item SWS\_Lin\_00171

**Trace References:**

none

**Content:**

On entering the state LIN\_INIT, the Lin module shall set each channel into state LIN\_CH\_OPERATIONALSLEEP, enable the wake-up detection (if enabled by LinChannelWakeupSupport), and optionally set the LIN hardware unit to reduced power operation mode (if supported by HW).

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73095: [LinSM][LinIf][Lin] Inconsistent LIN channel states upon initialization

**Problem description:**

As per SWS\_Lin\_00171, LIN channel state upon initialization shall be LIN\_CH\_OPERATIONAL

As per SWS\_LinIf\_00507, LinIf channel state upon initialization shall be either LINIF\_CHANNEL\_OPERATIONAL or LINIF\_CHANNEL\_SLEEP depending on the configuration parameter LinIfStartupState

As per SWS\_LinSM\_00152, LinSM state for all networks upon initialization shall be LINSM\_NO\_COM

This is an inconsistent state within the LIN Stack since LIN driver channel is operational but LinSM is in NO\_COM and LinIf state could be either!

Tracking the history of LinIf (Bug 27547) it seems that when LinIfStartupState was introduced, then Bug 27547 comment 8 was not addressed at that time to avoid this inconsistency.

Is there a use case to start the LIN channels in OPERATIONAL?

Should the LinSM already transition the channels to SLEEP following initialization? (similar to CanSM)

Or should the Lin driver specifications be updated to allow for either state following initialization (similar to LinIf and LinTrcv specifications)

### Agreed solution:

=====  
For CP R4.3.1  
=====

#### SRS LIN

=====

1) [SRS\_Lin\_01514], Description: Change  
from

<The LIN Interface shall inform an upper layer if a wake-up request was notified by the underlying LIN Driver

<A wakeup notification shall only be raised in the case a go-to-sleep-command has been issued to the bus.

to

>The LIN Interface shall inform an upper layer if a wake-up request was notified by the underlying LIN Driver.

#### SWS LinIf

=====

1) Sec. 4.1 Limitations: Add following description

+It's assumed that all of connected LIN Slave ECUs can receive a wakeup frame when they are already operational (as LIN Master ECU starts with LINIF\_CHANNEL\_SLEEP state).

2) Chap. 7, [SWS\_LinIf\_00507]: Change  
from

<[SWS\_LinIf\_00507] The LIN Interface shall transit from LINIF\_UNINIT to

< LINIF\_CHANNEL\_OPERATIONAL when the function LinIf\_Init is called and the configuration parameter LinIfStartupState is set to NORMAL.

< LINIF\_CHANNEL\_SLEEP when the function LinIf\_Init is called and the configuration parameter LinIfStartupState is set to SLEEP. ()

to

>[SWS\_LinIf\_00507] The LIN Interface shall transit from LINIF\_UNINIT to LINIF\_CHANNEL\_SLEEP without sending go-to-sleep command, when the function LinIf\_Init is called. ()

>Note: it is assumed that automatically slave nodes will enter bus sleep mode earliest after 4s and latest 10s of bus inactivity (as specified in LIN 2.1).

3) Chap. 7, Figure 4 (stm LinIf Node Management): Update the figure as de-

scribed below

- Remove the transition to LINIF\_CHANNEL\_OPERATIONAL with LinIf\_Init [LinIfStartupState=NORMAL]
- Remove the transition to LINIF\_CHANNEL\_SLEEP with LinIf\_Init [LinIfStartupState=SLEEP]
- +Add the initial transition to LINIF\_CHANNEL\_SLEEP inside the state LINIF\_INIT

4) Chap. 10, ECUC\_LinIf\_00069: Set the parameter LinIfStartupState to obsolete

5) Chap. 10, Figure 20: Remove the parameter LinIfStartupState

## SWS LinTrcv

=====

### Chapter 7.1

Update figure: remove transitions from NOT\_ACTIVE to LINTRCV\_TRCV\_MODE\_STANDBY and LINTRCV\_TRCV\_MODE\_NORMAL.

Remove the following sentence:

The function LinTrcv\_Init() causes a state change to either LINTRCV\_TRCV\_MODE\_SLEEP, LINTRCV\_TRCV\_MODE\_NORMAL or LINTRCV\_TRCV\_MODE\_STANDBY (any of these 3 states belong to the upper state ACTIVE). This depends on the configuration and is independent configurable for each channel.

Change SWS\_LinTrcv\_00119 from

The function LinTrcv\_Init shall set the LIN transceiver hardware to the state configured by the configuration parameter LinTrcvInitState.

This can be LINTRCV\_TRCV\_MODE\_NORMAL, LINTRCV\_TRCV\_MODE\_STANDBY or LINTRCV\_TRCV\_MODE\_SLEEP.

to

The function LinTrcv\_Init shall set the LIN transceiver hardware to the state LINTRCV\_TRCV\_MODE\_SLEEP.

Remove SWS\_LinTrcv\_00146.

remove configurations note after SWS\_LinTrcv\_00146

set ECUC\_LinTrcv\_00005 to obsolete.

## SWS Lin

=====

- Change SWS\_Lin\_00171 to: "On entering the state LIN\_INIT, the Lin module

shall set each channel into state LIN\_CH\_SLEEP, enable the wake-up detection (if enabled by LinChannelWakeupSupport), and optionally set the LIN hardware unit to reduced power operation mode (if supported by HW)."

- Change initial transition in figure 7-1 to LIN\_CH\_SLEEP.

- SWS\_Lin\_00168 Lin\_GetStatus: Change LIN\_OPERATIONAL description to "Normal operation; the related LIN channel is woken up from the LIN\_CH\_SLEEP and no data has been sent."

#### SWS LinSM

=====

no changes required

#### TPS System Template

=====

1) Appendix C: Remove the rule of the parameter LinIfStartupState

–Last change on issue 73095 comment 46–

#### BW-C-Level:

Application	Specification	Bus
1	4	4

## 1.16 Specification Item SWS\_Lin\_00191

#### Trace References:

none

#### Content:

Service name:	Lin_SendFrameLin_SendFrame	
Syntax:	Std_ReturnType Lin_SendFrame( uint8 Channel, const Lin_PduType* PduInfoPtr )	
Service ID[hex]:	0x04	
Sync/Async:	Asynchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	ChannelLin_SendFrame.Channel	LIN channel to be addressed
	PduInfoPtrLin_SendFrame.PduInfoPtr	Pointer to PDU containing the PID, checksum model, response type, DI and SDU data pointer
Parameters (inout):	None	

Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: Send command has been accepted. E_NOT_OK: Send command has not been accepted, development or production error occurred.
Description:	Sends a LIN header and a LIN response, if necessary. The direction of the frame response (master response, slave response, slave-to-slave communication) is provided by the PduInfoPtr.	

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68035: [diverse] Introduce rules defining which input parameters shall be passed per value and which ones per const reference

#### Problem description:

SWS\_BSW\_00186 especially states that input pointer parameters shall use the const qualifier (i.e., shall be P2CONST).

In addition to that there shall be a SWS item that states that input parameters of integral and enum type shall be passed by value whereas input parameters of structure type shall be passed by reference.

The various transformer SWS documents shall be adapted accordingly.

–Last change on issue 68035 comment 4–

#### Agreed solution:

BSW UML model

The attachment "Changed Proposal in WP-A meeting" contains a list of changes to the APIs in the model (see column H). Afterwards all related documents (included in impact list) shall update their generated artifacts.

#### General Requirements on Basic Software Modules

~~~~~

Introduce the following requirements prior to SRS\_BSW\_00371:

SRS\_BSW\_XXXXX: Input parameters of scalar and enum types shall be passed as a value.

Type: valid

Description: All input parameters of scalar or enum type shall be passed as a value.  
Rationale:

Use case: For example a function named `<Mip>_SomeFunction` with a return type of `Std_ReturnType` and a single parameter named `SomeParameter` of type `uint8` is defined with the following signature:

`Std_ReturnType <Mip>_SomeFunction(uint8 SomeParameter);`

Dependencies: —

Supporting Material: —

SRS\_BSW\_yyyyy: Input parameters of structure type shall be passed as a reference to a constant structure

Type: valid

Description: All input parameters of structure type shall be passed as a reference constant structure

Rationale: Passing input parameters of structure type by value would result in additional run-time overhead due to efforts for copying the whole structure.

Use case: For example a function named `<Mip>_SomeFunction` with a return type of `Std_ReturnType` and a single parameter named `SomeParameter` of type `SomeStructure` (where `SomeStructure` is a struct) is defined with the following signature:

`Std_ReturnType <Mip>_SomeFunction(P2CONST(SomeStructure, AUTOMATIC, <MIP>_APPL_DATA) SomeParameter);`

Dependencies: —

Supporting Material: —

SRS\_BSW\_zzzzz: Input parameters of array type shall be passed as a reference to the constant array base type

Type: valid

Description: All input parameters of array type shall be passed as a reference to the constant array base type

Rationale: This effectively matches the behavior specified in the ISO-C:90 namely that a "declaration of a parameter as 'array of type' shall be adjusted to 'qualified pointer to type'".

Use case: For example a function named `<Mip>_SomeFunction` with a return type of `Std_ReturnType` and a single parameter named `SomeParameter` of type array of `uint8` is defined with the following signature:



Std\_ReturnType      <Mip>\_SomeFunction(P2CONST(uint8,      AUTOMATIC,  
<MIP>\_APPL\_DATA) SomeParameter);

Dependencies: –

Supporting Material: —

## General Specification of Transformers

~~~~~

In SWS\_Xfrm\_00036 change

const <type>\* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules  
rules defined by the SRS BSW General (see SRS\_BSW\_XXXXX, SRS\_BSW\_YYYYY,  
and SRS\_BSW\_ZZZZZ) and SWS BSW General (see SWS\_BSW\_00186 and  
SWS\_BSW\_00187).

In SWS\_Xfrm\_00038 change

[<type> data\_1,] ...

[<type> data\_n]

to

[<paramtype> data\_1,] ...

[<paramtype> data\_n]

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules  
rules defined by the SRS BSW General (see SRS\_BSW\_XXXXX, SRS\_BSW\_YYYYY,

and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and SWS\_BSW\_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data\_1, ..., data\_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS\_Rte\_01017], [SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

In SWS\_Xfrm\_00040 change

[<originalData1>, ...  
<originalDataN>]

to

[<paramtype> originalData1,] ...  
[<paramtype> originalDataN]

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy, and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and SWS\_BSW\_00187).

In SWS\_Xfrm\_00044 change

<type> \*data\_1, ...  
<type> \*data\_n

to

[<paramtype> data\_1,] ...  
[<paramtype> data\_n]

and add the following to the where clause after the API table after the bullet

"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS\_BSW\_XXXXX, SRS\_BSW\_YYYYY, and SRS\_BSW\_ZZZZZ) and SWS BSW General (see SWS\_BSW\_00186 and SWS\_BSW\_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data\_1, ..., data\_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS\_Rte\_01017], [SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

Speci?cation of SOME/IP Transformer

~~~~~

In SWS\_SomeIpXf\_00138 change

const <type>\* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS\_BSW\_XXXXX, SRS\_BSW\_YYYYY, and SRS\_BSW\_ZZZZZ) and SWS BSW General (see SWS\_BSW\_00186 and SWS\_BSW\_00187).

In SWS\_SomeIpXf\_00141 change

[<type> data\_1,] ...

[<type> data\_n]

to

[<paramtype> data\_1,] ...  
[<paramtype> data\_n]

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules  
rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy,  
and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and  
SWS\_BSW\_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the  
transformer as data\_1, ..., data\_n the requirements to API parameters stated in  
chapter API Parameters of [5, SWS RTE] are valid (especially [SWS\_Rte\_01017],  
[SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

In SWS\_SomelpXf\_00145 change

<type> \*data\_1, ...  
<type> \*data\_n

to

[<paramtype> data\_1,] ...  
[<paramtype> data\_n]

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules  
rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy,  
and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and  
SWS\_BSW\_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data\_1, ..., data\_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS\_Rte\_01017], [SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

### Specification of COM Based Transformer

~~~~~

In SWS\_ComXf\_00007 change

const <type>\* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet "type is data type of the data element"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy, and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and SWS\_BSW\_00187).

### Specification of Time Sync over Ethernet

~~~~~

In SWS\_EthTSyn\_00040 make the parameter DataPtr of EthTSyn\_RxIndication const.

### Specification of SWS FlexRay Interface

~~~~~

Change SWS\_Frlf\_05073 from  
Frlf\_NumOfStartupFramesPtr (IN)

to  
FrIf\_NumOfStartupFramesPtr (OUT)

### Specification of ADC

~~~~~

~[SWS\_Adc\_00419] Adc\_SetupResultBuffer: change Adc\_ValueGroupType\* to  
const Adc\_ValueGroupType\*

~[SWS\_Adc\_00369] Adc\_ReadGroup: move Adc\_ValueGroupType \* from Parame-  
ters (in) to Parameters (out)

There is no need to change parameter from IN to INOUT in Adc\_SetupResultBuffer

### Specification of Com

~~~~~

Change type of parameter MetaData of Com\_TriggerIPDUSendWithMetaData from  
uint8\* to const uint8\*

### Specification of ComM

~~~~~

no change required

### Specification of Dem

~~~~~

no change required

### Specification of DLT

~~~~~

no change required

### Specification of DoIP

~~~~~

From:

Std\_ReturnType <User>\_DoIPRoutingActivationConfirmation(boolean\* Confirmed,  
uint8\* ConfirmationReqData, uint8\* ConfirmationResData)

Std\_ReturnType <User>\_DoIPRoutingActivationAuthentication(boolean\* Authenti-  
fied, uint8\* AuthenticationReqData, uint8\* AuthenticationResData)

To:

Std\_ReturnType <User>\_DoIPRoutingActivationConfirmation(boolean\* Confirmed, const uint8\* ConfirmationReqData, uint8\* ConfirmationResData)

Std\_ReturnType <User>\_DoIPRoutingActivationAuthentication(boolean\* Authenticated, const uint8\* AuthenticationReqData, uint8\* AuthenticationResData)

#### Specification of E2ELibrary

~~~~~

no change required

#### Specification of Eth

~~~~~

no change required

#### Specification of EthIf

~~~~~

no change required

#### Specification of EthSwitchDriver

~~~~~

no change required

#### Specification of ICUDriver

~~~~~

SWS\_Icu\_00201: Icu\_StartTimestamp

Parameter (IN): Icu\_ValueType\* BufferPtr shall be changed to Parameters (out) type

#### Specification of LdCom

~~~~~

[SWS\_LDCOM\_00027]: LdCom\_CopyTxData

BufReq\_ReturnType LdCom\_CopyTxData( PduIdType id, const PduInfoType\* info, RetryInfoType\* retry, PduLengthType\* availableDataPtr ) shall be changed to

BufReq\_ReturnType LdCom\_CopyTxData( PduIdType id, const PduInfoType\* info, const RetryInfoType\* retry, PduLengthType\* availableDataPtr )

[SWS\_LDCOM\_00036]: Rte\_LdComCbkJCopyTxData\_<sn>

BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_<sn>( const PduInfoType\* info,

RetryInfoType\* retry, PduLengthType\* availableDataPtr ) shall be changed to  
BufReq\_ReturnType Rte\_LdComCbKCopyTxData\_<sn>( const PduInfoType\* info,  
const RetryInfoType\* retry, PduLengthType\* availableDataPtr )

### Specification of Lin

~~~~~

PduInfoPtr needs to be const in Std\_ReturnType Lin\_SendFrame( uint8 Channel,  
const Lin\_PduType\* PduInfoPtr )

### Specification of PduR

~~~~~

\* PduR\_<User:LoTp>CopyTxData  
add const to "RetryInfoType\* retry"

### Specification of J1939Nm

~~~~~

Change parameter 'name' of User\_AddressClaimedIndication to type 'const uint8'

### Specification of SoAd

~~~~~

=> everything already fixed with RfC 65633

### Specification of SPIHandlerDriver

~~~~~

=> nothing to change for SWS SPI

### Specification of SynchronizedTimeBaseManager

~~~~~

"StbM not affected. All issues listed in the WP-A attachment have been already  
implemented by IT 69124 in context of RfC 65633"

### Specification of Tcplp

~~~~~

~[SWS\_TCPIP\_00040] Tcplp\_DhcpReadOption: change DataPtr from (IN) to  
(OUT)

~[SWS\_TCPIP\_00189] Tcplp\_DhcpV6ReadOption: change DataPtr from (IN) to



(OUT)

=> everything else already fixed with RfC 65633

#### Specification of TimeSyncOverFlexRay

~~~~~

"Change SWS\_FrTSyn\_00064: parameter versioninfo of type Std\_VersionInfoType\* is marked wrongly as IN. Change to OUT"

#### Specification of EFX

~~~~~

~ [SWS\_Efx\_00355] Efx\_Debounce\_u8\_u8: Include constant for pointer Input-parameter as like below.

uint8 Efx\_Debounce\_u8\_u8( boolean X, Efx\_DebounceState\_Type \* State, const Efx\_DebounceParam\_Type \* Param, sint32 dT )

~ [SWS\_Efx\_00376] Efx\_MedianSort: The parameter <InType>\* Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS\_Efx\_00309] Efx\_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Efx\_RampCheckActivity(const Efx\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Efx\_00307] Efx\_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Efx\_RampGetSwitchPos(const Efx\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Efx\_00193] Efx\_Array\_Average: Include constant for pointer Input-parameter as like below.

<OutType> Efx\_Array\_Average\_<InTypeMn>\_<OutTypeMn>( const <InType>\* Array, uint16 Count)

#### Specification of MFL

~~~~~

~ [SWS\_Mfl\_00192] Mfl\_Debounce\_u8\_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl\_Debounce\_u8\_u8( boolean X, Mfl\_DebounceState\_Type\* State, const Mfl\_DebounceParam\_Type\* Param, float32 dT)

~ [SWS\_Mfl\_00266] Mfl\_DebounceInit: The parameter Mfl\_DebounceState\_Type\* State should be Out instead of In parameter as like below.

Parameters (in): X Initial value for the input state

Parameters (out): State Pointer to structure for debouncing state variables

~ [SWS\_Mfl\_00246] Mfl\_HystDeltaRight\_f32\_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl\_HystDeltaRight\_f32\_u8( float32 X, float32 Delta, float32 Rsp, const uint8\* State)

~ [SWS\_Mfl\_00285] Mfl\_MedianSort\_f32\_f32: The parameter Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS\_Mfl\_00037] Mfl\_PT1SetState: The parameter State\_cpst should be Out instead of In parameter as like below.

Parameters (in): X1\_f32 Initial value for input state

Y1\_f32 Initial value for output state

Parameters (out): State\_cpst Pointer to internal state structure

~ [SWS\_Mfl\_00225] Mfl\_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Mfl\_RampCheckActivity( const Mfl\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Mfl\_00223] Mfl\_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Mfl\_RampGetSwitchPos(const Mfl\_StateRamp\_Type\* State\_cpst)

–Last change on issue 68035 comment 135–

#### BW-C-Level:

Application	Specification	Bus
1	4	1

## 1.17 Specification Item SWS\_Lin\_00195

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: if the function Lin\_SendFrame is called before the LIN module was initialized, the function Lin\_SendFrame shall raise the **default development** error LIN\_E\_UNINIT and return with **otherwise** (if DET is disabled) return E\_NOT\_OK.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer"

to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.18 Specification Item SWS\_Lin\_00197

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: if the channel parameter is invalid, the function Lin\_SendFrame shall raise the **default development** error LIN\_E\_INVALID\_CHANNEL **and return with otherwise** (if DET is disabled) **return E\_NOT\_OK**.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS\_Rte\_06631]  
[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

**SRS\_Libraries:**

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_SPALGeneral:**

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_FlashTest:**

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_MFXLibrary:**

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SWS\_MemoryAbstractionInterface:**

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"



**SWS\_FlexRayNetworkManagement:**

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_CANStateManager:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_PDURouter:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_EEPROMDriver:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.19 Specification Item SWS\_Lin\_00198

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: the function Lin\_SendFrame shall check the parameter PduInfoPtr for not being a NULL pointer. If PduInfoPtr is a NULL pointer, the function Lin\_SendFrame shall raise the **default development** error LIN\_E\_PARAM\_POINTER **and return with otherwise** (if DET is disabled) **return E\_NOT\_OK**.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx

module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]"

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.20 Specification Item SWS\_Lin\_00199

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: if the LIN channel state-machine is in the state LIN\_CH\_SLEEP, the function Lin\_SendFrame shall raise the **default development** error LIN\_E\_STATE\_TRANSITION and return with **otherwise** (if DET is disabled) return E\_NOT\_OK.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"
- "[SWS\_XXX\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module

shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

#### SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
-Last change on issue 73570 comment 47-

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.21 Specification Item SWS\_Lin\_00213

**Trace References:**

none

**Content:**

The LIN Driver module shall report the **default development** error "LIN\_E\_STATE\_TRANSITION (0x04)", when Invalid state transition occurs from the current state.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the



correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_XXX\_YYYYY]

In case default error detection is enabled for the XXXX module: The XXXX module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_XXX\_YYYYY]

If default error detection is enabled: the function shall check that the service XXX\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_XXX\_YYYYY]

In case development error detection is enabled for the XXXX module: The XXXX module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_XXX\_YYYYY]

If development error detection is enabled: the function shall check that the service XXX\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

#### – SWS\_RTE –

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS\_Rte\_07676]
- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS\_Rte\_06631]  
[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

#### SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.22 Specification Item SWS\_Lin\_00214

**Trace References:**

none

**Content:**

The LIN Driver module shall report the default development error "LIN\_E\_UNINIT (0x00)", when the API Service is used without module initialization.

## RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]"

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS RTE:

- SWS RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_FlashTest:**

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_MFXLibrary:**

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SWS\_MemoryAbstractionInterface:**

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_FlexRayNetworkManagement:**

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_CANStateManager:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_PDURouter:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_EEPROMDriver:**

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.23 Specification Item SWS\_Lin\_00215

**Trace References:**

none

**Content:**

The LIN Driver module shall report the **default** **development** error "LIN\_E\_INVALID\_CHANNEL (0x02)", when API Service used with an invalid or inactive channel parameter.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance



Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

#### SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
–Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.24 Specification Item SWS\_Lin\_00216

### Trace References:

none

### Content:

The LIN Driver module shall report the **default development** error "LIN\_E\_INVALID\_POINTER (0x03)", when API Service is called with invalid configuration pointer.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

### **Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX\_E\_TRANSITION"
- "The DET provides services to store development errors"

#### Solution for SWS\_RTE:

- SWS\_RTE —
  - Change 4.8 Default errors to 4.8 Development errors
  - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
  - Remove [SWS\_Rte\_07676]
  - Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
  - Change [SWS\_Rte\_06631]
- [SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

#### SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
  - In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"
- |   |    |                                  |
|---|----|----------------------------------|
| Rename "AUTOSAR_SWS_DevelopmentErrorTracer" | to | "AUTOSAR_SWS_DefaultErrorTracer" |
|---|----|----------------------------------|

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.25 Specification Item SWS\_Lin\_00218

**Trace References:**

none

**Content:**

The LIN Driver module shall report the production or **default development** error "LIN\_E\_TIMEOUT (value assigned by DEM)", when Timeout caused by hardware error.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_XXX\_YYYYY]"

In case default error detection is enabled for the XXXX module: The XXXX module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_XXX\_YYYYY]"

If default error detection is enabled: the function shall check that the service XXX\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer"

to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"



SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.26 Specification Item SWS\_Lin\_00237

**Trace References:**

none

**Content:**

(Obsolete) If the LIN module detects an error and calls the Default Error Tracer, the LIN module's function that raised the **default development** error shall return immediately.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development

error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
–Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.27 Specification Item SWS\_Lin\_00248

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: If the parameter versioninfo is a NULL pointer, the function Lin\_GetVersionInfo shall raise the error LIN\_E\_PARAM\_POINTER.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### **Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service

xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX\_E\_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS\_Rte\_07676]
- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS\_Rte\_06631]  
[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.28 Specification Item SWS\_Lin\_00249

### Trace References:

none

### Content:

The LIN Driver module shall report the **default development** error "LIN\_E\_PARAM\_POINTER (0x05)", when API Service is called with a NULL pointer. In case of this error, the API service shall return immediately without any further action, beside reporting this **default development** error.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_XXX\_yyyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module



shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

— SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

#### SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.29 Specification Item SWS\_Lin\_00251

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: if the channel parameter is invalid, the function Lin\_CheckWakeup shall raise the **default development** error LIN\_E\_INVALID\_CHANNEL **and return with otherwise (if DET is disabled) return E\_NOT\_OK.**

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted docu-

ment, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

**Solution for SWS\_RTE:****– SWS\_RTE —**

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS\_Rte\_07676]
- Change [SWS\_RTE\_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS\_Rte\_06631]  
[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

**SRS\_Libraries:**

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_SPALGeneral:**

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

**SRS\_FlashTest:**

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":  
Rename "Development Error Tracer" to "Default Error Tracer"  
Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

**SWS\_MFXLibrary:**

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
 –Last change on issue 73570 comment 47–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.30 Specification Item SWS\_Lin\_00258

### Trace References:

none

### Content:

If **default development** error detection for the LIN module is enabled: if the function `Lin_WakeupInternal` is called before the LIN module was initialized, the function `Lin_WakeupInternal` shall raise the **default development** error `LIN_E_UNINIT`.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

#### Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters `*DevErrorDetect` are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_XXX\_YYYYY]

In case default error detection is enabled for the XXXX module: The XXXX module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_XXX\_YYYYY]

If default error detection is enabled: the function shall check that the service `xxx_Init` was previously called. If the check fails, the function shall raise the default error `XXX_E_NOT_INITIALIZED` otherwise (if DET is disabled) return `E_NOT_OK`. ()"

- "In case default errors are enabled,..."

- "module raises the Default error `XXX_E_TRANSITION`"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"



- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

#### SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

#### SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

#### SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.31 Specification Item SWS\_Lin\_00259

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: the function Lin\_WakeupInternal shall raise the **default development** error LIN\_E\_INVALID\_CHANNEL if the channel parameter is invalid or the channel is inactive.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

**Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

- SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development

error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"  
–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.32 Specification Item SWS\_Lin\_00260

**Trace References:**

none

**Content:**

If **default development** error detection for the LIN module is enabled: the function Lin\_WakeupInternal shall raise the **default development** error LIN\_E\_STATE\_TRANSITION if the LIN channel state-machine is not in the state LIN\_CH\_SLEEP.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #73570: No "default error" in AUTOSAR

**Problem description:**

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters \*DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

#### **Agreed solution:**

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS\_xxx\_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If default error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the default error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX\_E\_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS\_xxx\_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS\_xxx\_yyyyy]

If development error detection is enabled: the function shall check that the service xxx\_Init was previously called. If the check fails, the function shall raise the development error XXX\_E\_NOT\_INITIALIZED otherwise (if DET is disabled) return E\_NOT\_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX\_E\_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS\_RTE:

– SWS\_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS\_Rte\_07676]

- Change [SWS\_RTE\_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS\_Rte\_06631]

[SWS\_Rte\_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS\_BSW\_00337)

SRS\_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS\_SPAL\_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS\_SPAL\_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS\_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS\_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

SWS\_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR\_SWS\_DevelopmentErrorTracer" to "AUTOSAR\_SWS\_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

**BW-C-Level:**

Application	Specification	Bus
1	1	1