

<b>Document Title</b>	SWS_SPIHandlerDriver: 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_SPIHandlerDriver	3
1.1	Specification Item ECUC_Spi_00238	3
1.2	Specification Item SWS_Spi_00004	5
1.3	Specification Item SWS_Spi_00011	6
1.4	Specification Item SWS_Spi_00012	8
1.5	Specification Item SWS_Spi_00017	10
1.6	Specification Item SWS_Spi_00035	12
1.7	Specification Item SWS_Spi_00039	14
1.8	Specification Item SWS_Spi_00046	15
1.9	Specification Item SWS_Spi_00130	18
1.10	Specification Item SWS_Spi_00131	20
1.11	Specification Item SWS_Spi_00135	22
1.12	Specification Item SWS_Spi_00140	23
1.13	Specification Item SWS_Spi_00180	24
1.14	Specification Item SWS_Spi_00233	37
1.15	Specification Item SWS_Spi_00237	40
1.16	Specification Item SWS_Spi_00238	41
1.17	Specification Item SWS_Spi_00240	43
1.18	Specification Item SWS_Spi_00241	44
1.19	Specification Item SWS_Spi_00242	45
1.20	Specification Item SWS_Spi_00243	47
1.21	Specification Item SWS_Spi_00245	48
1.22	Specification Item SWS_Spi_00246	49
1.23	Specification Item SWS_Spi_00324	51
1.24	Specification Item SWS_Spi_00373	51
1.25	Specification Item SWS_Spi_00389	54

# 1 SWS\_SPIHandlerDriver

## 1.1 Specification Item ECUC\_Spi\_00238

### Trace References:

none

### Content:

Name	SpiHwUnitSynchronousSpiJob.SpiHwUnitSynchronous		
Parent Container	SpiJob		
Description	<p>If SpiHwUnitSynchronous is set to "SYNCHRONOUS", the SpiJob uses its containing SpiDriver in a synchronous manner. If it is set to "ASYNCHRONOUS", it uses the driver in an asynchronous way. If the parameter is not set, the SpiChannel uses the driver also in an asynchronous way.</p> <p>Tags: atp.Status=obsolete</p>		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ASYNCHRONOUSSpiJob.SpiHwUnitSynchronous.ASYNCHRONOUS	-	
	SYNCHRONOUSSpiJob.SpiHwUnitSynchronous.SYNCHRONOUS	-	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

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

- RfC #76542: Simplifying the driver by removing the prearranged bus requirements.

#### Problem description:

[SWS\_Spi\_00130][SWS\_Spi\_00131][SWS\_Spi\_00140]

The prearrangement of the bus makes the SPI driver too complex. A Job is defined by the CS# signal and it is not important if the job data is transmitted synchronous or asynchronous.

On the other hand of the job is prearranged and assigned to a hardware unit, this unit is locked for other devices.

Example:

As hardware developer i plan to use a CANTRCV together with an external EEP connected to the same hardware. Regarding to the requirements and if i choose a synchronous communication, the communication with the EEP is not possible due to the restriction of the prearranged bus.

Finally the aim is to remove this "restrictions" to the SPI driver.

**Agreed solution:**

\_Page 37, remove the following note:

"This functionality level, based on a mixed usage of synchronous transmission on one prearranged SPI bus and asynchronous transmission on others, generates restrictions on configuration and usage of Sequences and Jobs."

\_Page 38: remove SWS\_Spi\_00130, SWS\_Spi\_00131 and SWS\_Spi\_00140

[SWS\_Spi\_00130] The so-called synchronous Sequences shall only be composed of Jobs that are associated to the prearranged SPI bus. These Sequences shall be used with synchronous services3 only. ()

[SWS\_Spi\_00131] Jobs associated with the prearranged SPI bus shall not belong to Sequences containing Jobs associated with another SPI bus. In other words, mixed Sequences (synchronous with asynchronous Jobs) shall not be allowed. ()

[SWS\_Spi\_00140] If SpiHwUnitSynchronous is set to "Synchronous" for a job, the associated bus defined by SpiHwUnit behave same as prearranged bus. It means that all requirements valid for prearranged bus will be valid also for the bus assigned to this job. ()

\_Page 91: set ECUC\_Spi\_00238 to obsolete

SpiHwUnitSynchronous

Description: If SpiHwUnitSynchronous is set to "SYNCHRONOUS", the SpiJob uses its containing SpiDriver in a synchronous manner. If it is set to "ASYNCHRONOUS", it uses the driver in an asynchronous way. If the parameter is not set, the SpiChannel uses the driver also in an asynchronous way.

–Last change on issue 76542 comment 14–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

## 1.2 Specification Item SWS\_Spi\_00004

### Trace References:

SRS\_BSW\_00327, SRS\_BSW\_00337, SRS\_BSW\_00385

### Content:

SPI Handler/driver shall be able to detect the error SPI\_E\_PARAM\_CHANNEL (0x0A) when API service called with wrong parameter.

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

- RfC #59085: Rollout of 'Runtime errors'

#### Problem description:

Inconsistencies in SWS with semantics of Default errors  
 –Last change on issue 59085 comment 26–

#### Agreed solution:

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

#### Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

\_\_\_\_\_

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRuntimeError

SWS\_LinIf:

\_\_\_\_\_

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

### 1.3 Specification Item SWS\_Spi\_00011

**Trace References:**

none

**Content:**

After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT with the numeric value 0. .

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

- RfC #76532: MCAL: Numeric value definition

**Problem description:**

General question: Why we need to define in SWS the numeric values e.g. for Enumeration. If we decide to do that it shall be done as part of the table description and it shall be consistently done for all ranges, type etc.

In general it shall be defined in AUTOSAR whether defined values from UML model shall be extracted to the SWS. Example for a solution is to add the numeric values to the tables in chapter 8.

Numeric value definitions are redundant for the following items:

SWS\_Spi\_00011: After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT with the numeric value 0.(.)

Reason vor the value is coming from SRS\_BSW\_00406. This shall be added for traceability.

[SWS\_Spi\_00012] After reset, the type Spi\_JobResultType shall have the default value SPI\_JOB\_OK with the numeric value 0.()

[SWS\_Spi\_00017] After reset, the type Spi\_SeqResultType shall have the de-fault value SPI\_SEQ\_OK with the numeric value 0.()

**Agreed solution:**

Replace old description by this text:

[SWS\_Spi\_00011]: After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT.()

[SWS\_Spi\_00012]: After reset, the type Spi\_JobResultType shall have the de-fault value SPI\_JOB\_OK.()

[SWS\_Spi\_00017]: After reset, the type Spi\_SeqResultType shall have the de-fault value SPI\_SEQ\_OK.()

SWS\_Spi\_00373:

Spi\_StatusType: Table item "SPI\_UNINIT" shall have the value "0x00". Generation from UML model;

Traceability to be added to "SRS\_BSW\_00406, SRS\_BSW\_00335"

Affected CorTst:

Replace old description by the following :

[SWS\_CorTst\_00138]

For the type CorTst\_ErrOkType, the enumeration value CORTST\_E\_NOT\_TESTED shall be the default value after a reset. CorTstTestIntervalId shall have value zero per default.

Affected FlsTst:

Replace old description by the following :

[SWS\_FlsTst\_00049] For the type FlsTst\_StateType, the enumeration value FLSTST\_UNINIT shall be the default value after a reset.

[SWS\_FlsTst\_00053] For the type FlsTst\_TestResultFgndType, the enumeration value FLSTST\_NOT\_TESTED shall be the default value after a reset.

[SWS\_FlsTst\_00154] For the type FlsTst\_TestResultBgndType, the enumeration value FLSTST\_RESULT\_NOT\_TESTED shall be the default value after a reset.

[SWS\_FlsTst\_00164] UML shall be updated to generate FLSTST\_RESULT\_NOT\_TESTED with "0" value within this table.

Affected RamTst:

[SWS\_RamTst\_00006] For the type RamTst\_ExecutionStatusType, the enumeration value RAMTST\_EXECUTION\_UNINIT shall be the default value after a reset.

[SWS\_RamTst\_00012] For the type RamTst\_TestResultType (of the overall test result), the enumeration value RAMTST\_RESULT\_NOT\_TESTED shall be the default value after a reset.

Remove trace to (SRS\_BSW\_00406)

[SWS\_RamTst\_00013] For the type RamTst\_AlgorithmType, the enumeration value RAMTST\_ALGORITHM\_UNDEFINED shall be the default value after reset.

Remove trace to (SRS\_BSW\_00406)

Other MCAL modules:

Dio, Wdg, Ocu are not affected by this.  
 –Last change on issue 76532 comment 7–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.4 Specification Item SWS\_Spi\_00012

**Trace References:**

none

**Content:**

After reset, the type Spi\_JobResultType shall have the default value SPI\_JOB\_OK with the numeric value 0. .

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

- RfC #76532: MCAL: Numeric value definition

**Problem description:**

General question: Why we need to define in SWS the numeric values e.g. for Enumeration. If we decide to do that it shall be done as part of the table description and it shall be consistently done for all ranges, type etc.

In general it shall be defined in AUTOSAR whether defined values from UML model shall be extracted to the SWS. Example for a solution is to add the numeric values to the tables in chapter 8.

Numeric value definitions are redundant for the following items:

SWS\_Spi\_00011: After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT with the numeric value 0.()

Reason vor the value is coming from SRS\_BSW\_00406. This shall be added for traceability.

[SWS\_Spi\_00012] After reset, the type Spi\_JobResultType shall have the default value SPI\_JOB\_OK with the numeric value 0.()

[SWS\_Spi\_00017] After reset, the type Spi\_SeqResultType shall have the default value SPI\_SEQ\_OK with the numeric value 0.()

#### **Agreed solution:**

Replace old description by this text:

[SWS\_Spi\_00011]: After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT.()

[SWS\_Spi\_00012]: After reset, the type Spi\_JobResultType shall have the default value SPI\_JOB\_OK.()

[SWS\_Spi\_00017]: After reset, the type Spi\_SeqResultType shall have the default value SPI\_SEQ\_OK.()

SWS\_Spi\_00373:

Spi\_StatusType: Table item "SPI\_UNINIT" shall have the value "0x00". Generation from UML model;

Traceability to be added to "SRS\_BSW\_00406, SRS\_BSW\_00335"

Affected CorTst:

Replace old description by the following :

[SWS\_CorTst\_00138]

For the type CorTst\_ErrOkType, the enumeration value CORTST\_E\_NOT\_TESTED shall be the default value after a reset. CorTstTestIntervallId shall have value zero per default.

Affected FlsTst:

Replace old description by the following :

[SWS\_FlsTst\_00049] For the type FlsTst\_StateType, the enumeration value FLSTST\_UNINIT shall be the default value after a reset.

[SWS\_FlsTst\_00053] For the type FlsTst\_TestResultFgndType, the enumeration value FLSTST\_NOT\_TESTED shall be the default value after a reset.

[SWS\_FlsTst\_00154] For the type FlsTst\_TestResultBgndType, the enumeration value FLSTST\_RESULT\_NOT\_TESTED shall be the default value after a reset.

[SWS\_FlsTst\_00164] UML shall be updated to generate FLSTST\_RESULT\_NOT\_TESTED with "0" value within this table.

Affected RamTst:

[SWS\_RamTst\_00006] For the type RamTst\_ExecutionStatusType, the enumeration value RAMTST\_EXECUTION\_UNINIT shall be the default value after a reset.

[SWS\_RamTst\_00012] For the type RamTst\_TestResultType (of the overall test result), the enumeration value RAMTST\_RESULT\_NOT\_TESTED shall be the default value after a reset.

Remove trace to (SRS\_BSW\_00406)

[SWS\_RamTst\_00013] For the type RamTst\_AlgorithmType, the enumeration value RAMTST\_ALGORITHM\_UNDEFINED shall be the default value after reset.

Remove trace to (SRS\_BSW\_00406)

Other MCAL modules:

Dio, Wdg, Ocu are not affected by this.  
 –Last change on issue 76532 comment 7–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.5 Specification Item SWS\_Spi\_00017

**Trace References:**

none

**Content:**

After reset, the type Spi\_SeqResultType shall have the default value SPI\_SEQ\_OK with the numeric value 0.

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

- RfC #76532: MCAL: Numeric value definition

**Problem description:**

General question: Why we need to define in SWS the numeric values e.g. for Enumeration. If we decide to do that it shall be done as part of the table description and it shall be consistently done for all ranges, type etc.

In general it shall be defined in AUTOSAR whether defined values from UML model shall be extracted to the SWS. Example for a solution is to add the numeric values to the tables in chapter 8.

Numeric value definitions are redundant for the following items:

SWS\_Spi\_00011: After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT with the numeric value 0.()

Reason vor the value is coming from SRS\_BSW\_00406. This shall be added for traceability.

[SWS\_Spi\_00012] After reset, the type Spi\_JobResultType shall have the default value SPI\_JOB\_OK with the numeric value 0.()

[SWS\_Spi\_00017] After reset, the type Spi\_SeqResultType shall have the de-fault value SPI\_SEQ\_OK with the numeric value 0.()

#### **Agreed solution:**

Replace old description by this text:

[SWS\_Spi\_00011]: After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT.()

[SWS\_Spi\_00012]: After reset, the type Spi\_JobResultType shall have the de-fault value SPI\_JOB\_OK.()

[SWS\_Spi\_00017]: After reset, the type Spi\_SeqResultType shall have the de-fault value SPI\_SEQ\_OK.()

SWS\_Spi\_00373:

Spi\_StatusType: Table item "SPI\_UNINIT" shall have the value "0x00". Generation from UML model;

Traceability to be added to "SRS\_BSW\_00406, SRS\_BSW\_00335"

Affected CorTst:

Replace old description by the following :

[SWS\_CorTst\_00138]

For the type CorTst\_ErrOkType, the enumeration value CORTST\_E\_NOT\_TESTED shall be the default value after a reset. CorTstTestIntervalId shall have value zero per default.

Affected FlsTst:

Replace old description by the following :

[SWS\_FlsTst\_00049] For the type FlsTst\_StateType, the enumeration value FLSTST\_UNINIT shall be the default value after a reset.

[SWS\_FlsTst\_00053] For the type FlsTst\_TestResultFgndType, the enumeration

value FLSTST\_NOT\_TESTED shall be the default value after a reset.

[SWS\_FlsTst\_00154] For the type FlsTst\_TestResultBgndType, the enumeration value FLSTST\_RESULT\_NOT\_TESTED shall be the default value after a reset.

[SWS\_FlsTst\_00164] UML shall be updated to generate FLSTST\_RESULT\_NOT\_TESTED with "0" value within this table.

Affected RamTst:

[SWS\_RamTst\_00006] For the type RamTst\_ExecutionStatusType, the enumeration value RAMTST\_EXECUTION\_UNINIT shall be the default value after a reset.

[SWS\_RamTst\_00012] For the type RamTst\_TestResultType (of the overall test result), the enumeration value RAMTST\_RESULT\_NOT\_TESTED shall be the default value after a reset.

Remove trace to (SRS\_BSW\_00406)

[SWS\_RamTst\_00013] For the type RamTst\_AlgorithmType, the enumeration value RAMTST\_ALGORITHM\_UNDEFINED shall be the default value after reset.

Remove trace to (SRS\_BSW\_00406)

Other MCAL modules:

Dio, Wdg, Ocu are not affected by this.

–Last change on issue 76532 comment 7–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.6 Specification Item SWS\_Spi\_00035

**Trace References:**

SRS\_Spi\_12200, SRS\_Spi\_12201

**Content:**

When the function Spi\_SyncAsyncTransmit is called while a sequence is on transmission and SPI\_SUPPORT\_CONCURRENT\_SYNC\_TRANSMIT is disabled or another sequence is on transmission on same bus, the SPI Handler/Driver shall not take into account this new transmission request and the function shall return the value E\_NOT\_OK (see used with EB and the source data pointer has been provided as NULL using the Spi\_Setup EB method, the default transmit data configured for each channel shall be transmitted. (See also

[SWS\_SpiSPI\_0011400028]). In this case and according to SWS\_BSW\_00042 and SWS\_BSW\_00045, the SPIHandler/Driver shall report the SPI\_E\_SEQ\_IN\_PROCESS error.)

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

- RfC #75019: SWS\_Spi\_00035 has been eclipsed when switching from 4.0.3 to 4.1 release

#### Problem description:

SWS\_Spi\_00035 has been eclipsed when switching from 4.0.3 to 4.1 release and is basically a copy-paste of SWS\_Spi\_00135.

But SWS\_Spi\_00135 makes reference SWS\_Spi\_00100 that doesn't exist, so the text of the SWS\_Spi\_00135 should be taken from SWS\_Spi\_00035 that makes reference to the right SWS\_BSW reqs.

#### 1) 4.0.3

[SPI035] When the function Spi\_AsyncTransmit is used with EB and the source data pointer has been provided as NULL using the Spi\_SetupEB method, the default transmit data configured for each channel will be transmitted. (See also [SPI028]) (BSW12200, BSW12201)

[SPI135] When the function Spi\_SyncTransmit is called while a sequence is on transmission and SPI\_SUPPORT\_CONCURRENT\_SYNC\_TRANSMIT is disabled or another sequence is on transmission on same bus, the SPI Handler/Driver shall not take into account this new transmission request and the function shall return the value E\_NOT\_OK (see [SPI114]). In this case and according to [SPI100], the SPI Handler/Driver shall report the SPI\_E\_SEQ\_IN\_PROCESS error. ()

#### 2) 4.1

[SWS\_Spi\_00035] When the function Spi\_SyncTransmit is called while a sequence is on transmission and SPI\_SUPPORT\_CONCURRENT\_SYNC\_TRANSMIT is disabled or another sequence is on transmission on same bus, the SPI Handler/Driver shall not take into account this new transmission request and the function shall return the value E\_NOT\_OK (see [SWS\_Spi\_00114]). In this case and according to [SWS\_BSW\_00042] and [SWS\_BSW\_00045], the SPI Handler/Driver shall report the SPI\_E\_SEQ\_IN\_PROCESS error.) (SRS\_Spi\_12200, SRS\_Spi\_12201)

[SWS\_Spi\_00135] When the function Spi\_SyncTransmit is called while a sequence is on transmission and SPI\_SUPPORT\_CONCURRENT\_SYNC\_TRANSMIT is disabled or another sequence is on transmission on same bus, the SPI Handler/Driver shall not take into account this new transmission request and the function shall return the value E\_NOT\_OK (see [SWS\_Spi\_00114]). In this case and according to [SWS\_Spi\_00100], the SPI Handler/Driver shall report the SPI\_E\_SEQ\_IN\_PROCESS error. ()  
 –Last change on issue 75019 comment 1–

**Agreed solution:**

Restore SWS\_Spi\_00035 as the following:  
 [SWS\_Spi\_00035] When the function Spi\_AsyncTransmit is used with EB and the source data pointer has been provided as NULL using the Spi\_SetupEB method, the default transmit data configured for each channel shall be transmitted. (See also [SWS\_Spi\_00028]) (SRS\_Spi\_12200, SRS\_Spi\_12201)  
 –Last change on issue 75019 comment 8–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.7 Specification Item SWS\_Spi\_00039

**Trace References:**

[SRS\\_SPAL\\_00157](#), [SRS\\_Spi\\_12104](#)

**Content:**

The function Spi\_GetSequenceResult shall return the last transmission result of the specified Sequence.

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

- RfC #75024: SWS\_Spi\_00324 and SWS\_Spi\_00039 are redundant

**Problem description:**

In chapter 8.3.9 Spi\_GetSequenceResult, SWS\_Spi\_00324 and SWS\_Spi\_00039 are redundant.

[SWS\_Spi\_00324] The function Spi\_GetSequenceResult shall return the last transmission result of the specified Sequence.()

[SWS\_Spi\_00039] The function Spi\_GetSequenceResult shall return the last transmission result of the specified Sequence. (SRS\_SPAL\_00157, SRS\_Spi\_12104)

**Agreed solution:**

1) Remove SWS\_Spi\_00039

2) [SWS\_Spi\_00324] - add traces to SRS\_SPAL\_00157 and SRS\_Spi\_12104  
 –Last change on issue 75024 comment 9–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.8 Specification Item SWS\_Spi\_00046

**Trace References:**

SRS\_BSW\_00406

**Content:**

If **default development** error detection for the SPI module is enabled and the SPI Handler/Driver's environment calls any API function before initialization, an error should be reported to the DET with the error value SPI\_E\_UNINIT according to the configuration.

**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

## 1.9 Specification Item SWS\_Spi\_00130

**Trace References:**

none

**Content:**

The so-called synchronous Sequences shall only be composed of Jobs that are associated to the prearranged SPI bus. These Sequences shall be used with synchronous services3only.

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

- RfC #76542: Simplifying the driver by removing the prearranged bus requirements.

#### Problem description:

[SWS\_Spi\_00130][SWS\_Spi\_00131][SWS\_Spi\_00140]

The prearrangement of the bus makes the SPI driver too complex. A Job is defined by the CS# signal and it is not important if the job data is transmitted synchronous or asynchronous.

On the other hand of the job is prearranged and assigned to a hardware unit, this unit is locked for other devices.

Example:

As hardware developer i plan to use a CANTRCV together with an external EEP connected to the same hardware. Regarding to the requirements and if i choose a synchronous communication, the communication with the EEP is not possible due to the restriction of the prearranged bus.

Finally the aim is to remove this "restrictions" to the SPI driver.

#### Agreed solution:

\_Page 37, remove the following note:

"This functionality level, based on a mixed usage of synchronous transmission on one prearranged SPI bus and asynchronous transmission on others, generates restrictions on configuration and usage of Sequences and Jobs."

\_Page 38: remove SWS\_Spi\_00130, SWS\_Spi\_00131 and SWS\_Spi\_00140

[SWS\_Spi\_00130] The so-called synchronous Sequences shall only be composed of Jobs that are associated to the prearranged SPI bus. These Sequences shall be used with synchronous services3 only. ()

[SWS\_Spi\_00131] Jobs associated with the prearranged SPI bus shall not belong

to Sequences containing Jobs associated with another SPI bus. In other words, mixed Sequences (synchronous with asynchronous Jobs) shall not be allowed. ()

[SWS\_Spi\_00140] If SpiHwUnitSynchronous is set to "Synchronous" for a job, the associated bus defined by SpiHwUnit behave same as prearranged bus. It means that all requirements valid for prearranged bus will be valid also for the bus assigned to this job. ()

\_Page 91: set ECUC\_Spi\_00238 to obsolete

SpiHwUnitSynchronous

Description: If SpiHwUnitSynchronous is set to "SYNCHRONOUS", the SpiJob uses its contain-

ing SpiDriver in a synchronous manner. If it is set to "ASYNCHRONOUS", it uses the driver in an asynchronous way. If the parameter is not set, the SpiChannel uses the driver also in an asynchronous way.

–Last change on issue 76542 comment 14–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

## 1.10 Specification Item SWS\_Spi\_00131

**Trace References:**

none

**Content:**

Jobs associated with the prearranged SPI bus shall not belong to Sequences containing Jobs associated with another SPI bus. In other words, mixed Sequences (synchronous with asynchronous Jobs) shall not be allowed.

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

- RfC #76542: Simplifying the driver by removing the prearranged bus requirements.

**Problem description:**

[SWS\_Spi\_00130][SWS\_Spi\_00131][SWS\_Spi\_00140]

The prearrangement of the bus makes the SPI driver too complex. A Job is defined by the CS# signal and it is not important if the job data is transmitted synchronous or asynchronous.

On the other hand of the job is prearranged and assigned to a hardware unit, this unit is locked for other devices.

Example:

As hardware developer i plan to use a CANTRCV together with an external EEP connected to the same hardware. Regarding to the requirements and if i choose a synchronous communication, the communication with the EEP is not possible due to the restriction of the prearranged bus.

Finally the aim is to remove this "restrictions" to the SPI driver.

**Agreed solution:**

\_Page 37, remove the following note:

"This functionality level, based on a mixed usage of synchronous transmission on one prearranged SPI bus and asynchronous transmission on others, generates restrictions on configuration and usage of Sequences and Jobs."

\_Page 38: remove SWS\_Spi\_00130, SWS\_Spi\_00131 and SWS\_Spi\_00140

[SWS\_Spi\_00130] The so-called synchronous Sequences shall only be composed of Jobs that are associated to the prearranged SPI bus. These Sequences shall be used with synchronous services3 only. ()

[SWS\_Spi\_00131] Jobs associated with the prearranged SPI bus shall not belong

to Sequences containing Jobs associated with another SPI bus. In other words, mixed Sequences (synchronous with asynchronous Jobs) shall not be allowed. ()

[SWS\_Spi\_00140] If SpiHwUnitSynchronous is set to "Synchronous" for a job, the associated bus defined by SpiHwUnit behave same as prearranged bus. It means that all requirements valid for prearranged bus will be valid also for the bus assigned to this job. ()

\_Page 91: set ECUC\_Spi\_00238 to obsolete

SpiHwUnitSynchronous

Description: If SpiHwUnitSynchronous is set to "SYNCHRONOUS", the SpiJob uses its contain-

ing SpiDriver in a synchronous manner. If it is set to "ASYNCHRONOUS", it uses the driver in an asynchronous way. If the parameter is not set, the SpiChannel uses the driver also in an asynchronous way.

–Last change on issue 76542 comment 14–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

## 1.11 Specification Item SWS\_Spi\_00135

### Trace References:

SRS\_Spi\_12093

### Content:

When the function Spi\_SyncTransmit is called while a sequence is on transmission and SPI\_SUPPORT\_CONCURRENT\_SYNC\_TRANSMIT is disabled or another sequence is on **transmission transmission** on same bus, the SPI Handler/Driver shall not take into account this new transmission request and the function shall return the value E\_NOT\_OK (see [SWS\_Spi\_00114]). In this case **and according to SWS\_Spi\_00100**, the SPI Handler/Driver shall report the SPI\_E\_SEQ\_IN\_PROCESS error **according to [SWS\_BSW\_00042] and [SWS\_BSW\_00045]**.

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

- RfC #76397: [Spi] SWS\_Spi\_00135 refers to SWS\_Spi\_00100

#### Problem description:

SWS\_Spi\_00135 refers to SWS\_Spi\_00100.

Is this reference made for more clarity on SPI\_E\_SEQ\_IN\_PROCESS? If so, please refer SWS\_Spi\_00245.

In other case, is this reference made for SpiDevErrorDetect, then please refer to ECUC\_Spi\_00228.

Note: In historical SPIHandlerDriver SWS, SPI100 used to refer to SpiDevErrorDetect.

#### Agreed solution:

[SWS\_Spi\_00135] When the function Spi\_SyncTransmit is called while a sequence is on transmission and SPI\_SUPPORT\_CONCURRENT\_SYNC\_TRANSMIT is disabled or another sequence is on transmission on same bus, the SPI Handler/Driver shall not take into account this new transmission request and the function shall return the value E\_NOT\_OK (see [SWS\_Spi\_00114]). In this case, the SPI Handler/Driver shall report the SPI\_E\_SEQ\_IN\_PROCESS error according to [SWS\_BSW\_00042] and [SWS\_BSW\_00045]. (SRS\_Spi\_12093)  
–Last change on issue 76397 comment 10–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.12 Specification Item SWS\_Spi\_00140

### Trace References:

none

### Content:

If SpiHwUnitSynchronous is set to "Synchronous" for a job, the associated bus defined by SpiHwUnit behave same as prearranged bus. It means that all requirements valid for prearranged bus will be valid also for the bus assigned to this job.

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

- RfC #76542: Simplifying the driver by removing the prearranged bus requirements.

#### Problem description:

[SWS\_Spi\_00130][SWS\_Spi\_00131][SWS\_Spi\_00140]

The prearrangement of the bus makes the SPI driver too complex. A Job is defined by the CS# signal and it is not important if the job data is transmitted synchronous or asynchronous.

On the other hand of the job is prearranged and assigned to a hardware unit, this unit is locked for other devices.

Example:

As hardware developer i plan to use a CANTRCV together with an external EEP connected to the same hardware. Regarding to the requirements and if i choose a synchronous communication, the communication with the EEP is not possible due to the restriction of the prearranged bus.

Finally the aim is to remove this "restrictions" to the SPI driver.

#### Agreed solution:

\_Page 37, remove the following note:

"This functionality level, based on a mixed usage of synchronous transmission on one prearranged SPI bus and asynchronous transmission on others, generates restrictions on configuration and usage of Sequences and Jobs."

\_Page 38: remove SWS\_Spi\_00130, SWS\_Spi\_00131 and SWS\_Spi\_00140  
 [SWS\_Spi\_00130] The so-called synchronous Sequences shall only be composed of Jobs that are associated to the prearranged SPI bus. These Sequences shall be used with synchronous services3 only. ()

[SWS\_Spi\_00131] Jobs associated with the prearranged SPI bus shall not belong to Sequences containing Jobs associated with another SPI bus. In other words, mixed Sequences (synchronous with asynchronous Jobs) shall not be allowed. ()

[SWS\_Spi\_00140] If SpiHwUnitSynchronous is set to "Synchronous" for a job, the associated bus defined by SpiHwUnit behave same as prearranged bus. It means that all requirements valid for prearranged bus will be valid also for the bus assigned to this job. ()

\_Page 91: set ECUC\_Spi\_00238 to obsolete  
 SpiHwUnitSynchronous

Description: If SpiHwUnitSynchronous is set to "SYNCHRONOUS", the SpiJob uses its containing SpiDriver in a synchronous manner. If it is set to "ASYNCHRONOUS", it uses the driver in an asynchronous way. If the parameter is not set, the SpiChannel uses the driver also in an asynchronous way.

–Last change on issue 76542 comment 14–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

### 1.13 Specification Item SWS\_Spi\_00180

**Trace References:**

none

**Content:**

Service name:	Spi_SetupEBSpi_SetupEB
Syntax:	Std_ReturnType Spi_SetupEB( Spi_ChannelType Channel, const Spi_DataBufferType* SrcDataBufferPtr, Spi_DataBufferType* DesDataBufferPtr, Spi_NumberOfDataType Length )

Service ID[hex]:	0x05	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ChannelSpi_SetupEB.Channel	Channel ID.
	SrcDataBufferPtrSpi_SetupEB.SrcData BufferPtr	Pointer to source data buffer.
	DesDataBufferPtrSpi_SetupEB.DesData BufferPtr	Pointer to destination data buffer in RAM.
	LengthSpi_SetupEB.Length	Length (number of data elements) of the data to be transmitted from SrcData BufferPtr and/or received from DesData BufferPtr Min.: 1 Max.: Max of data specified at configuration for this channel

**None**

Parameters (inout):	DesDataBufferPtrSpi_SetupEB.DesData BufferPtr	Pointer to destination data buffer in RAM.
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: Setup command has been accepted E_NOT_OK: Setup command has not been accepted
Description:	Service to setup the buffers and the length of data for the EB SPI Handler/Driver Channel specified.	

**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 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\_SomelpXf\_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  
Frlf\_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 Parameters (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\* Authenti-  
fied, 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\_LdComCbkJCopyTxData\_<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   |

## 1.14 Specification Item SWS\_Spi\_00233

**Trace References:**

none

**Content:**

If **default development** error detection for the SPI module is enabled, the calling of the routine SPI\_Init() while the SPI driver is already initialized will cause a development error SPI\_E\_ALREADY\_INITIALIZED and the desired functionality shall be left without any action.

**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 "AU-

TOSAR\_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   |

## 1.15 Specification Item SWS\_Spi\_00237

**Trace References:**

none

**Content:**

SPI Handler/driver shall be able to detect the error SPI\_E\_PARAM\_JOB (0x0B) when API service called with wrong parameter.

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

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors  
 –Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

\_\_\_\_\_

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRuntimeError

SWS\_LinIf:

\_\_\_\_\_

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

\_\_\_\_\_

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |

## 1.16 Specification Item SWS\_Spi\_00238

**Trace References:**

none

**Content:**

SPI Handler/driver shall be able to detect the error SPI\_E\_PARAM\_SEQ (0x0C) when API service called with wrong parameter.

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

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors  
 –Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

**Notes:**

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

---

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

SWS\_LinIf:

---

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

---

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW

General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |

## 1.17 Specification Item SWS\_Spi\_00240

**Trace References:**

none

**Content:**

SPI Handler/driver shall be able to detect the error SPI\_E\_PARAM\_LENGTH (0x0D) when API service called with wrong parameter.

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

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors  
 –Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

**Notes:**

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

**SWS\_LinIf:**

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

**SWS\_UdpNm:**

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |

## 1.18 Specification Item SWS\_Spi\_00241

**Trace References:**

none

**Content:**

SPI Handler/driver shall be able to detect the error SPI\_E\_PARAM\_UNIT (0x0E) when API service called with wrong parameter.

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

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment <https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

---

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

SWS\_LinIf:

---

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

---

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |

## 1.19 Specification Item SWS\_Spi\_00242

**Trace References:**

none

**Content:**

SPI Handler/driver shall be able to detect the error SPI\_E\_UNINIT (0x1A) when API service used without module initialization.

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

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors  
 –Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

**Notes:**

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

**\*\*\* BSW UML Model \*\*\***

**SWS\_CanNm:**

\_\_\_\_\_

**Chapter 8.6.1 Optional Interfaces:**

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

**SWS\_LinIf:**

\_\_\_\_\_

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

**SWS\_UdpNm:**

\_\_\_\_\_

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API  
 UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

**\*\*\* ECUC XML \*\*\***

Not affected. No configuration of runtime error reporting required (see SWS BSW  
 General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |

## 1.20 Specification Item SWS\_Spi\_00243

### Trace References:

none

### Content:

SPI Handler/driver shall be able to detect the error SPI\_E\_SEQ\_PENDING (0x2A) when services called in a wrong sequence.

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

- RfC #59085: Rollout of 'Runtime errors'

#### Problem description:

Inconsistencies in SWS with semantics of Default errors  
 –Last change on issue 59085 comment 26–

#### Agreed solution:

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

#### Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

\_\_\_\_\_

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRuntimeError

SWS\_LinIf:

\_\_\_\_\_

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |

## 1.21 Specification Item SWS\_Spi\_00245

**Trace References:**

none

**Content:**

SPI Handler/driver shall be able to detect the error SPI\_E\_SEQ\_IN\_PROCESS (0x3A) when synchronous transmission service called at wrong time.

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

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment <https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.

- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

\_\_\_\_\_

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

SWS\_LinIf:

\_\_\_\_\_

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

\_\_\_\_\_

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |

## 1.22 Specification Item SWS\_Spi\_00246

**Trace References:**

none

**Content:**

SPI Handler/driver shall be able to detect the error SPI\_E\_ALREADY\_INITIALIZED (0x4A) when API SPI\_Init service called while the SPI driver has already been initialized time.

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

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors  
 –Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

**Notes:**

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

---

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

SWS\_LinIf:

---

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

---

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |

## 1.23 Specification Item SWS\_Spi\_00324

### Trace References:

[SRS\\_SPAL\\_00157](#), [SRS\\_Spi\\_12104](#)

### Content:

The function Spi\_GetSequenceResult shall return the last transmission result of the specified Sequence.

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

- RfC #75024: SWS\_Spi\_00324 and SWS\_Spi\_00039 are redundant

#### Problem description:

In chapter 8.3.9 Spi\_GetSequenceResult, SWS\_Spi\_00324 and SWS\_Spi\_00039 are redundant.

[SWS\_Spi\_00324] The function Spi\_GetSequenceResult shall return the last transmission result of the specified Sequence.()

[SWS\_Spi\_00039] The function Spi\_GetSequenceResult shall return the last transmission result of the specified Sequence. (SRS\_SPAL\_00157, SRS\_Spi\_12104)

#### Agreed solution:

1) Remove SWS\_Spi\_00039

2) [SWS\_Spi\_00324] - add traces to SRS\_SPAL\_00157 and SRS\_Spi\_12104  
 –Last change on issue 75024 comment 9–

#### BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 1             | 1   |

## 1.24 Specification Item SWS\_Spi\_00373

### Trace References:

[SRS\\_BSW\\_00406](#), [SRS\\_BSW\\_00335](#)

### Content:

|              |                                                                      |        |                                                               |
|--------------|----------------------------------------------------------------------|--------|---------------------------------------------------------------|
| Name:        | Spi_StatusTypeSpi_StatusType                                         |        |                                                               |
| Type:        | Enumeration                                                          |        |                                                               |
| Range:       | SPI_UNINITSpi_Status<br>Type.SPI_UNINIT                              | - 0x00 | The SPI Handler/Driver is not initialized or not usable.      |
|              | SPI_IDLESpi_Status<br>Type.SPI_IDLE                                  | - 0x01 | The SPI Handler/Driver is not currently transmitting any Job. |
|              | SPI_BUSYSpi_Status<br>Type.SPI_BUSY                                  | - 0x02 | The SPI Handler/Driver is performing a SPI Job (transmit).    |
| Description: | This type defines a range of specific status for SPI Handler/Driver. |        |                                                               |

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

- RfC #76532: MCAL: Numeric value definition

#### Problem description:

General question: Why we need to define in SWS the numeric values e.g. for Enumeration. If we decide to do that it shall be done as part of the table description and it shall be consistently done for all ranges, type etc.

In general it shall be defined in AUTOSAR whether defined values from UML model shall be extracted to the SWS. Example for a solution is to add the numeric values to the tables in chapter 8.

Numeric value definitions are redundant for the following items:

SWS\_Spi\_00011: After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT with the numeric value 0.()

Reason vor the value is coming from SRS\_BSW\_00406. This shall be added for traceability.

[SWS\_Spi\_00012] After reset, the type Spi\_JobResultType shall have the default value SPI\_JOB\_OK with the numeric value 0.()

[SWS\_Spi\_00017] After reset, the type Spi\_SeqResultType shall have the de-fault value SPI\_SEQ\_OK with the numeric value 0.()

#### Agreed solution:

Replace old description by this text:

[SWS\_Spi\_00011]: After reset, the type Spi\_StatusType shall have the default value SPI\_UNINIT.()

[SWS\_Spi\_00012]: After reset, the type Spi\_JobResultType shall have the de-fault value SPI\_JOB\_OK.()

[SWS\_Spi\_00017]: After reset, the type Spi\_SeqResultType shall have the de-fault

value SPI\_SEQ\_OK.()

SWS\_Spi\_00373:

Spi\_StatusType: Table item "SPI\_UNINIT" shall have the value "0x00". Generation from UML model;

Traceability to be added to "SRS\_BSW\_00406, SRS\_BSW\_00335"

Affected CorTst:

Replace old description by the following :

[SWS\_CorTst\_00138]

For the type CorTst\_ErrOkType, the enumeration value CORTST\_E\_NOT\_TESTED shall be the default value after a reset. CorTstTestIntervallId shall have value zero per default.

Affected FlsTst:

Replace old description by the following :

[SWS\_FlsTst\_00049] For the type FlsTst\_StateType, the enumeration value FLSTST\_UNINIT shall be the default value after a reset.

[SWS\_FlsTst\_00053] For the type FlsTst\_TestResultFgndType, the enumeration value FLSTST\_NOT\_TESTED shall be the default value after a reset.

[SWS\_FlsTst\_00154] For the type FlsTst\_TestResultBgndType, the enumeration value FLSTST\_RESULT\_NOT\_TESTED shall be the default value after a reset.

[SWS\_FlsTst\_00164] UML shall be updated to generate FLSTST\_RESULT\_NOT\_TESTED with "0" value within this table.

Affected RamTst:

[SWS\_RamTst\_00006] For the type RamTst\_ExecutionStatusType, the enumeration value RAMTST\_EXECUTION\_UNINIT shall be the default value after a reset.

[SWS\_RamTst\_00012] For the type RamTst\_TestResultType (of the overall test result), the enumeration value RAMTST\_RESULT\_NOT\_TESTED shall be the default value after a reset.

Remove trace to (SRS\_BSW\_00406)

[SWS\_RamTst\_00013] For the type RamTst\_AlgorithmType, the enumeration value RAMTST\_ALGORITHM\_UNDEFINED shall be the default value after reset.

Remove trace to (SRS\_BSW\_00406)

Other MCAL modules:

Dio, Wdg, Ocu are not affected by this.

–Last change on issue 76532 comment 7–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 1             | 1   |

## 1.25 Specification Item SWS\_Spi\_00389

**Trace References:**

none

**Content:**

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

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors  
 –Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

**Notes:**

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

---

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

SWS\_LinIf:

---

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

**SWS\_UdpNm:**

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

**\*\*\* ECUC XML \*\*\***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1           | 4             | 1   |