

Document Title	SRS_BSWGeneral: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1	SRS_BSWGeneral	3
1.1	Specification Item SRS_BSW_00101	3
1.2	Specification Item SRS_BSW_00158	4
1.3	Specification Item SRS_BSW_00300	6
1.4	Specification Item SRS_BSW_00305	8
1.5	Specification Item SRS_BSW_00348	14
1.6	Specification Item SRS_BSW_00350	24
1.7	Specification Item SRS_BSW_00374	26
1.8	Specification Item SRS_BSW_00378	31
1.9	Specification Item SRS_BSW_00381	33
1.10	Specification Item SRS_BSW_00404	35
1.11	Specification Item SRS_BSW_00406	37
1.12	Specification Item SRS_BSW_00411	39
1.13	Specification Item SRS_BSW_00412	41
1.14	Specification Item SRS_BSW_00429	43
1.15	Specification Item SRS_BSW_00447	44
1.16	Specification Item SRS_BSW_00456	46
1.17	Specification Item SRS_BSW_00464	47
1.18	Specification Item SRS_BSW_00465	49
1.19	Specification Item SRS_BSW_00484	51
1.20	Specification Item SRS_BSW_00485	64
1.21	Specification Item SRS_BSW_00486	76

1 SRS_BSWGeneral

1.1 Specification Item SRS_BSW_00101

Trace References:

RS_BRF_01136

Content:

Type:	
Description:	If a Basic Software Module needs to initialize variables and hardware resources, this should be done in a separate initialization function. This function shall be named <Module name>_Init(). This function shall only be called by the BswM or EcuM.
Rationale:	Interface to ECU state manager
Applies to:	
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #75556: Incomplete Requirements in SRS_General

Problem description:

During completion of tracing I found that a requirement is missing in SRS_General. I could not trace my requirement:

SWS_Det_00024:

If the Default Error Tracer has not been initialized before Det reporting functions are called, the reporting functions shall return immediately without any other action (no Error_Hook shall be used, no implementer specific function shall be performed and no error shall be reported

Against any of the requirements in the SRS_General.

The section on initialization is incomplete and should describe also the behaviour in case the init functions have not been called.

Furthermore the tracing of the init-function in SRS_BSW_00101 is missing to a feature.

Proposal: extend: SRS_BSW_00406 and complete tracing in SRS_BSW_00101

Agreed solution:

SRS_BSW_00406 - add:

v) If Det not initialized before reporting functions, it shall return immediately without any other action.

Requirement tracing SRS_BSW_00101- (RS_BRF_01136)

–Last change on issue 75556 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item SRS_BSW_00158

Trace References:

none

Content:

Type:	
Description:	All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation.
Rationale:	Easy and clear configuration.
Applies to:	
Use Case:	The file Adc_Cfg.h contains the pre-compile time configurable parameters to set the properties of the module Adc.
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS re-

quirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)
- Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item SRS_BSW_00300

Trace References:

RS_BRF_01024

Content:

Type:	
Description:	All AUTOSAR Basic Software Modules shall be identified by an unambiguous name. The module name is always part of related files.
Rationale:	The module name serves as an identifier and classification mechanism in order to group module related files.
Applies to:	
Use Case:	Example: Eep.c, Eep.h, Eep_Cfg.h
Supporting Material:	WP Architecture SW Module List (Module Abbreviations)

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment

(<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.4 Specification Item SRS_BSW_00305

Trace References:

RS_BRF_01024

Content:

Type:	
Description:	All AUTOSAR Basic Software Modules shall label data types according to the following scheme:
Rationale:	Enhance readability and unique classification of data type identifiers.
Applies to:	
Use Case:	
Supporting Material:	BMW Standard Core Programming Guidelines –

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #78035: SRS requirements refer to BMW specifications

Problem description:

AUTOSAR documents shall not relate to any particular company.

The requirements below reference various BMW specifications and have to be corrected.

Document: SRS_EEPROMDriver:

[SRS_Eep_00087] The EEPROM driver shall provide an asynchronous read function

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_00088] The EEPROM driver shall provide an asynchronous write function

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_00089] The EEPROM driver shall provide an asynchronous erase function

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_00090] The EEPROM driver shall provide a synchronous cancel function

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_00091] The EEPROM driver shall provide a synchronous function which returns the job processing status

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_00096] EEPROM driver static shall be configured

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_00092] The EEPROM driver shall only write data if at least one data value of the affected erasable block is different from the data value to be written

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_00094] The EEPROM driver shall handle the EEPROM memory segmentation

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_00095] The EEPROM driver shall handle only one job at the same time

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_12047] The EEPROM driver shall provide a function that has to be called for job processing

"Supporting Material: BMW Specification MCAL V1.0a"

[SRS_Eep_12050] The job processing function of the EEPROM driver shall process only as much data as the EEPROM hardware can handle

"Supporting Material: BMW Specification MCAL V1.0a"

Document: SRS_SPALGeneral:

[SRS_SPAL_12056] All driver modules shall allow the static configuration of notification mechanism

"Supporting Material: BMW Specification MCAL V1.0a, [...]"

[SRS_SPAL_12057] All driver modules shall implement an interface for initialization

"Supporting Material: BMW Specification MCAL V1.0a, MAL1.0.0"

[SRS_SPAL_12063] All driver modules shall only support raw value mode

"Supporting Material: BMW Specification MCAL V1.0a, MAL1.6.0"

[SRS_SPAL_12064] All driver modules shall raise an error if the change of the operation mode leads to degradation of running operations

"Supporting Material: BMW Specification MCAL V1.0a, MAL1.5.2"

SRS_SPAL_12067 All driver modules shall set their wake-up conditions depending on the selected operation mode

"Supporting Material: BMW Specification MCAL V1.0a, MAL1.5.0"

Document: SRS_PWMDriver:

[SRS_Pwm_12293] The PWM driver shall allow the static configuration of PWM channel properties

"Supporting Material: BMW Specification MCAL V1.0a, MAL13.1.4 The idle level configuration covers the active phase requirement from Hella.

[SRS_Pwm_12295] The PWM driver shall provide a service for setting the duty cycle of a selected channel

"Supporting Material: BMW Specification MCAL V1.0a, MAL13.x"

[SRS_Pwm_12297] The PWM driver shall provide a service for setting the period of a selected channel

"Supporting Material: BMW Specification MCAL V1.0a, MAL13.8.0; Kojak movies (in case you do not know Kojak)"

Document: SRS_BSWGeneral:

Chapter 5 "General Requirements on Basic Software"

The ECU application experience is taken from the following concrete applications:

Sunroof and power window ECU

Diesel engine ECU

ESP ECU

BMW, DC and VW standard software packages (Standard Core, Standard Software Platform, Standard Software Core) including OSEK OS, communication modules,

bootloader, basic diagnostic functions for the domains listed above
Infotainment control ECU

[SRS_BSW_00305] Data types naming convention
"Supporting Material: BMW Standard Core Programming Guidelines"

Document: SRS_COM:

[SRS_Com_00177] AUTOSAR COM and LargeDataCOM shall support multiple configuration stages
"Use Case: It must be possible to configure the handled bus frames after compile- or build-time, particularly for future concepts running at BMW (reuse an ECU within another vehicle product line with different and incompatible communication layouts)."

Document: SRS_ADCDriver:

[SRS_Adc_12280] The ADC Driver shall allow a specific result access modes for each ADC Channel Group
"Supporting Material: BMW Specification MCAL V1.0a, MAL14.2.0."

[SRS_Adc_12283] The ADC driver shall mask out information bits from the conversion result not belonging to the ADC value
"Supporting Material: BMW Specification MCAL V1.0a, MAL14.4.1."

[SRS_Adc_12819] The ADC Driver shall provide a synchronous service for reading the last valid conversion results of the selected channel group
"Supporting Material: BMW Specification MCAL V1.0a."

[SRS_Adc_12291] The ADC Driver shall provide a service for querying the status of an ADC Channel Group
"Supporting Material: BMW Specification MCAL V1.0a."

[SRS_Adc_12288] Based on the channel group configuration the ADC driver shall be able to handle the buffers of stream jobs
"Supporting Material: BMW Specification MCAL V1.0a."

[SRS_Adc_12292] If the ADC provides signed values, the ADC driver shall put the sign bit into the MSB of the return value
"Supporting Material: BMW Specification MCAL V1.0a, MAL14.4.2."

Document: SRS_DIODriver:

[SRS_Dio_12003] The DIO Driver shall provide a service that writes a data word to the assigned DIO port

"Supporting Material: BMW Specification MCAL V1.0a, REQ MAL10.3.0"

[SRS_Dio_12004] The DIO Driver shall provide a service that writes a selectable number of adjoining bits to an assigned part of a DIO port

"Supporting Material: BMW Specification MCAL V1.0a, REQ MAL10.3.1"

[SRS_Dio_12005] The DIO Driver shall provide a service for write access to single DIO channels

"Supporting Material: BMW Specification MCAL V1.0a, REQ MAL10.4.0"

[SRS_Dio_12006] The DIO Driver shall provide a service for reading a data word from the assigned DIO port

"Supporting Material: BMW Specification MCAL V1.0a, REQ MAL10.4.0"

[SRS_Dio_12007] The DIO Driver shall provide a service for reading a selectable number of adjoining bits from an assigned part of a DIO port

"Supporting Material: BMW Specification MCAL V1.0a, REQ MAL10.4.1"

[SRS_Dio_12008] The DIO Driver shall provide a service for reading one bit of an assigned DIO channel

"Supporting Material: BMW Specification MCAL V1.0a, REQ MAL10.6.0"

Agreed solution:

CP_SRS_ADCCDriver

SRS_Adc_12280, SRS_Adc_12283, SRS_Adc_12819, SRS_Adc_12291,
SRS_Adc_12288, SRS_Adc_12292

Change from:

Supporting Material: BMW Specification MCAL V1.0a, [...]

to:

Supporting Material: –

====

CP_SRS_BSWGeneral

replace "BMW Standard Core Programming Guidelines" by "–" in SRS_BSW_00305

===

CP_SRS_COM

Update [SRS_Com_00177] AUTOSAR COM and LargeDataCOM shall support multiple configuration stages

"Use Case: It must be possible to configure the handled bus frames after compile- or build-time, for example when reusing an ECU within another vehicle product line with different and incompatible communication layouts."

====

CP_SRS_DIODriver

SRS_Dio_12003, SRS_Dio_12004, SRS_Dio_12005, SRS_Dio_12006,
SRS_Dio_12007, SRS_Dio_12008

Change from:

Supporting Material: BMW Specification MCAL V1.0a, REQ MAL10.x.x

to:

Supporting Material: –

====

CP_SRS_EEPROMDriver

SRS_Eep_00087, SRS_Eep_00088, SRS_Eep_00089, SRS_Eep_00090,
SRS_Eep_00091, SRS_Eep_00092, SRS_Eep_00094, SRS_Eep_00095,
SRS_Eep_00096, SRS_Eep_12047, SRS_Eep_12050

Change from:

Supporting Material: BMW Specification MCAL V1.0a

to:

Supporting Material: –

====

CP_SRS_PWMDriver

SRS_Pwm_12293:

Change from:

Supporting Material: BMW Specification MCAL V1.0a, MAL13.1.4. The idle level configuration covers the active phase requirement from Hella.

Change to:

Supporting Material: –

SRS_Pwm_12295:

Change from:

Supporting Material: BMW Specification MCAL V1.0a, MAL13.x

Change to:

Supporting Material: –

SRS_Pwm_12297:

Change from:

Supporting Material: BMW Specification MCAL V1.0a, MAL13.8.0. Kojak movies (in case you do not know Kojak)

Change to:

Supporting Material: –

===

CP_SRS_SPALGeneral

SRS_SPAL_12056, SRS_SPAL_12057, SRS_SPAL_12063, SRS_SPAL_12064,
SRS_SPAL_12067

Change from:

Supporting Material: BMW Specification MCAL V1.0a, ...

to:

Supporting Material: –

–Last change on issue 78035 comment 14–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.5 Specification Item SRS_BSW_00348

Trace References:

RS_BRF_01024

Content:

Type:	
Description:	All AUTOSAR standard types and constants shall be placed and organized in a standard type header file.

Rationale:	Provide uniform framework wide access to standard types to be used by all modules.
Applies to:	
Use Case:	Each module that uses AUTOSAR integer data types and/or the standard return type shall include the file Std_Types.h.
Supporting Material:	Important note for implementation of this header file:

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73564: References to OSEK

Problem description:

The OSEK web site is not available.
References need to be updated.

OSEK was pushed to ISO 17356
ISO 17356-2: introduction?
ISO 17356-3: OS
ISO 17356-4: COM
ISO 17356-5: NM (not sure we need to have such reference)
ISO 17356-6: OIL (not sure we need to have such reference)

I tried to select the documents where it would make sense to get a fix.

Agreed solution:

Replace references to
* OSEK web site
* the OSEK file name / version
with an ISO reference (with ISO version)

Check references to OSEK subsection (or avoid such references), to make sure that the section numbering in ISO is the same.

SWS OS
=====

Change references in chapter 3.2.1 to ISO. Remove [16], [18], [19] and [20]

Remove [22] from chapter 3.2.2

Remove "OSEKtime OS [16] and the HIS Protected OSEK [22] are immature specifications that contain concepts necessary for AUTOSAR and satisfy specific

application domains. It is the purpose of this document to identify these needs and to recommend the use of parts (or all) of these specifications as appropriate." from chapter 4.1.

Change "OSEK OS" to new ISO reference ("OSEK OS[..]" - many occurrences, also in chapter 10)

Remove "So called hard and smooth synchronization from OSEKtime [16] are supported by this single unified concept in AUTOSAR OS. Smooth synchronization may be emulated by setting the small adjustment values on the final expiry point. Hard synchronization may be emulated by setting large adjustment values on the final expiry point." from 7.4.2.2.3

Remove "and provides the type of protection given by the OSEKtime Interrupt re-enable schedule event [16]." from 7.7.2.1

Remove chapter 12.3

SWCT:
=====

Replace reference to OSEK COM:

Fifteen filter algorithms formally described by the enumeration type DataFilter-TypeEnum in the meta-model are taken from OSEK COM 3.0.3 specification [18] that is referenced by the RTE specification [2].

by:

Fifteen filter algorithms formally described by the enumeration type DataFilter-TypeEnum in the meta-model are taken from the ISO 17356-4 specification [18] that is referenced by the RTE specification [2].

Replace:

[TPS_SWCT_01222] Applicability of DataFilter

This OSEK specification states that filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).(RS_SWCT_03221)

by:

[TPS_SWCT_01222] Applicability of DataFilter

The ISO 17356-4 specification states that filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).(RS_SWCT_03221)

Replace:

[constr_1044] Applicability of DataFilter

According to the origin of DataFilter, i.e. OSEK COM 3.0.3 specification [18], DataFilters can only be applied to values with an integer base type.()

by:

[constr_1044] Applicability of DataFilter

According to the origin of DataFilter, i.e. ISO 17356-4 specification [18], DataFilters can only be applied to values with an integer base type.()

Replace footnote to [constr_1090]:

This constraint is valid at least in the OSEK standard where an extended task (that can have wait points) can only exist a single time in the context of the scheduler.

by:

This constraint is valid at least in the ISO 17356-3 standard where an extended task (that can have wait points) can only exist a single time in the context of the scheduler.

=====

Dem

=====

Replace the reference [17] Communication in Chapter 3.1 Input documents & related standards and norms Bibliography by : ISO 17356-3 in www.iso.org/

=====

Dcm

=====

Replace the reference [8]Communication in Chapter 3.1 Input documents & related standards and norms Bibliography by : ISO 17356-3 in www.iso.org/

=====

EXP_VFB

Add an entry in chapter "13 References" related to ISO 17356-4:
ISO 17356-4
OSEK/VDX Communication (COM)
www.iso.org

Change the references to "OSEK-COM V3.0.3" in Table 4.2 (4.3.2 From the point of view of the receiver) and EXP_Vfb_00028 (4.3.4 Filtering between the sender and the receiver) to "ISO 17356-4". Link the references to the entry in chapter "13 References".

=====
SRS_BSWGeneral

5 General Requirements on Basic Software

Replace "OSEK OS" by "ISO 17356-3"

5.2.3.4 Standard header Files
Replace in [SRS_BSW_00348]

Because E_OK is already defined within OSEK OS, E_OK has to be checked for being already defined:
/* for OSEK compliance this typedef has been added */

by

Because E_OK is already defined within ISO 17356-3, E_OK has to be checked for being already defined
/* for ISO 17356 compliance this typedef has been added */

6.2 Related Standards and Norms

Remove
6.2.1 OSEK
[STD_OSEK_OS] OSEK/VDX Operating System Specification
<http://www.osek-vdx.org>

Replace with ISO 17356-3 norm

=====
SRS_COM

- 1)
Remove [DOC_OSEK_GLOS] and all its references, since a) AUTOSAR has an own wording and glossary b) the references to (old) OSEK-terms most probably create more confusion than guidance
- 2)
Set reference of [DOC_OSEK_COM] to: ISO 17356-4: COM
Move reference to Chapter 7.3 ISO and remove Chapter 7.2 OSEK
- 3)
Remove section references to [DOC_OSEK_COM], just keep the textual references no numbers
- 4)
Search and replace textual references to OSEK COM 3.0.3 by ISO 17356-4: COM
- 5)
[SRS_Com_02084]: Change following text in Description,
from
< The possibilities to define those conditions shall be the same as defined in [DOC_OSEK_COM] reception filter algorithms (see [DOC_OSEK_GLOS], Section 2.2.2).
to
> The possibilities to define those conditions shall be the same as defined in [DOC_ISO_COM] reception filter algorithms (see [DOC_ISO_COM], Section 3.2.3).
- 6)
[SRS_Com_02058]: Change following text in Supporting Material,
from
< If no update bits are used, the AUTOSAR COM module provides the deadline monitoring defined in [DOC_OSEK_COM] (Section 2.5.1).
to
> If no update bits are used, the AUTOSAR COM module provides the deadline monitoring defined in [DOC_ISO_COM] (Section 3.5.1).

=====
SRS_Os

in chapter 6.2.1 OSEK:

Replace "[STD_OSEK_OS] OSEK/VDX Operating System, Version 2.2.3, <http://www.osek-vdx.org/mirror/os223.pdf>"

by "[STD_OSEK_OS] ISO 17356-3: OS"

Replace "[STD_OSEK_OIL] OSEK / VDX Implementation Language (OIL) V2.5,
OSEK Implementation Language, <http://www.osek-vdx.org/mirror/oil25.pdf>"
by "[STD_OSEK_OIL] ISO 17356-6: OIL"

Remove "[STD_OSEK_TTOS] OSEK/VDX Time-Triggered Operating System,
Version 1.0, July 24, 2001, <http://www.osek-vdx.org/mirror/ttos10.pdf>"

Remove "[STD_OSEK_ORTI] OSEK/VDX ORTI (OSEK RunTime Interface)
Part A Version 2.1.1, Part B Version 2.1, <http://www.osek-vdx.org/mirror/ORTI-A-211.pdf>"

in [SRS_Os_11002]: remove [STD_OSEK_TTOS] from Supporting Material

=====
SWS_COM

1)
Set reference of [17] to ISO 17356-4: COM
Set reference of [18] to ISO 17356-6: OIL

2)
Search and replace textual references to OSEK COM 3.0.3 by ISO 17356-4: COM
or [17] (if suitable)
and textual references to OSEK OIL by ISO 17356-6: OIL or [18] (if suitable)

=====
SWS_StandardTypes:
In Section 3.2: replace
[7] OSEK/VDX Operating System, Version 2.2.2 www.osek-vdx.org/os222.pdf
by
[7] OSEK/VDX Operating System, ISO 17356-3: OS

=====
CP_TR_AutosarModelConstraints
see PS for the SWCT.

=====
SRS_NetworkManagement:

1) Change Section 7.2.1 name from "OSEK" to "ISO 17356-5"

Remove the description:

[5] [STD_OSEK_NM]

OSEK/VDX NM Specification (ISO 17356-5), Version 2.5.3

[STD_OSEK_NM] OSEK/VDX NM Specification (ISO 17356-5), V2.5.3

<http://www.osek-vdx.org/>

Change to :

OSEK/VDX NM Specification

www.iso.org

====>

7.2.1 ISO 17356-5

[5] ISO 17356-5: NM Specification

www.iso.org

2) Replace "OSEK-NM" to "ISO 17356-5: NM Specification" in [SRS_Nm_02515]

3) Replace "OSEK NM 2.5.3" to "ISO 17356-5: NM Specification" in [SRS_Nm_00142]

–Last change on issue 73564 comment 28–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #75557: Missing Requirements for STD_HIGH, STD_LOW and STD_ACTIVE, STD_IDLE and STD_ON, STD_OFF

Problem description:

Name: Oscar Slotosch

Phone: +49 89 538866911

Role: Doc Owner SWS_StdTypes

Description/Motivation:

During requirements tracing completion I noticed that there is missing two requirements for the definition of the types:

SWS_Std_00007:STD_HIGH, STD_LOW

SWS_Std_00013:STD_ACTIVE, STD_IDLE

SWS_Std_00010: STD_ON, STD_OFF

If we do not find/add a requirement for those types to SRS_General we would have to remove the types.

This could cause many compiler errors all over the world.

Proposed Solution: Elther point me to some requirements that I can use for tracing, or create new requirements (preferable with traces to RS_Features ;-)

Was there already a decision? No

Agreed solution:

Update description of SRS_BSW_00348 with a new bullet:
"- define values for STD_HIGH, STD_LOW, STD_ACTIVE, STD_IDLE"
–Last change on issue 75557 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A
Phone:
Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)
–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.6 Specification Item SRS_BSW_00350

Trace References:

RS_BRF_01028

Content:

Type:	
Description:	All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors. It shall be configurable and the default value of the configuration shall be that those error type is disabled.
Rationale:	Provide module wide debug instrumentation facilities. Each defined keyword has to be properly documented.
Applies to:	
Use Case:	Example:
Supporting Material:	< MODULENAME > shall be derived from WP Architecture "List of Basic Software Modules", [DOC_MOD_LIST] (2...8 characters)

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming

rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.7 Specification Item SRS_BSW_00374

Trace References:

RS_BRF_01032

Content:

Type:	
Description:	All Basic Software Modules shall provide a readable module vendor identification (according to HIS) in their published parameters.
Rationale:	Allow identification of module vendor
Applies to:	
Use Case:	EEP_VENDOR_ID
Supporting Material:	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73567: [diverse] references to HIS

Problem description:

The HIS website is no more available.

We should check what is the impact on the AUTOSAR documents.

There is already RfC 64155 which will make us rid of some references, but I assume some will remain:

- * SHE
- * list of vendor IDs
- * IO driver API
- * bootloader
- ...

I tried to list the document where the HIS reference will survive RfC 64155 to avoid duplicate ITs.

Agreed solution:

Generic approach for the solutions (to be refined per document by documents owners (see below)

- * Check the need for an external reference
- * Try to find a replacement
- * Backup: ask PL solution

AUTOSAR_SRS_BSW_GENERAL

=====

Create new reference to the AUTOSAR Vendor ID List in §6.1:

[VENDOR_ID_LIST] AUTOSAR Vendor ID List

<https://www.autosar.org/documents/vendor-id/>

SRS_BSW_00374:

Remove reference "(according to HIS)" in the Description

Supporting Material:

replace reference "HIS Software Supplier Identifications

[STD_HIS_SUPPLIER_IDS]"

to "AUTOSAR Vendor ID List [VENDOR_ID_LIST]"

Remove Chapter 6.2.2 HIS

SRS_COM

=====

Remove Chapter 7.4 HIS

(the reference is not used anymore)

SRS_CryptoStack

=====

Remove abbreviation SHE from table in chapter 3 Acronyms and abbreviations

SRS_DIODriver

=====

Remove abbreviation HIS from table in chapter 3 Acronyms and abbreviations.

SRS_EEPROMDriver

=====

Remove abbreviation HIS from table in chapter 3 Acronyms and abbreviations.

Remove chapter 6.2 Related standards and norms.

SRS_FlashDriver

=====

- remove HIS from list of abbreviations
- remove requirement SRS_FIs_12083
- remove HIS Flash Driver document from reference list

SRS_FlashTest

=====

- Remove HIS from the table of acronyms in chapter 3

SRS_GPTDriver

=====

- Remove abbreviation HIS from table in chapter 4 Acronyms and abbreviations
- Remove from Chapter 7.2 the reference "[5] HIS API I/O Driver Specification [www.automotive-his.de/results/ API_IODriver_2.1.3.pdf](http://www.automotive-his.de/results/API_IODriver_2.1.3.pdf)".

SRS_ICUDriver

=====

- Remove abbreviation HIS from table in chapter 3 Acronyms and abbreviations
- Remove Chapter 7.1.1 HIS

SRS_IOHWAbstraction

=====

Remove chapter 7.2 Related standards and norms.

SRS_MCUDriver

=====

- Remove abbreviation HIS from table in chapter 3 Acronyms and abbreviations
- Remove from Chapter 7.2 the reference "[5] HIS API I/O Driver Specification [www.automotive-his.de/results/ API_IODriver_2.1.3.pdf](http://www.automotive-his.de/results/API_IODriver_2.1.3.pdf)".

SRS_NetworkManagement

=====

- Remove section 7.2.2 "HIS"

SRS_OCUDriver

=====

- Remove abbreviation HIS from table in chapter 2 Acronyms and abbreviations
- Remove whole chapter 5.2 Related standards and norms (including the single subchapter 5.2.1 HIS)

SRS_OS

=====

- Remove abbreviation HIS from table in chapter 2.2 Acronyms and abbreviations
- Remove chapter 5.2.2 HIS

SRS_PortDriver

=====

- Remove abbreviation HIS from table in chapter 3 Acronyms and abbreviations
- Remove reference [STD_HIS_IO_DRIVER] in chapter 6.1 Deliverables of AUTOSAR "[STD_HIS_IO_DRIVER] HIS API IO Driver, V2.1.3, April 29th, 2004, http://www.automotive-his.de/download/API_IODriver_2_1_3.pdf"
- Remove chapter 6.2 Related standards and norms

SRS_PWMDriver

=====

- Remove abbreviation HIS from table in chapter 4 Acronyms and abbreviations
- Remove Supporting Material for requirement SRS_Pwm_12299

SRS_RAMTest

=====

- Remove abbreviation HIS from table in chapter 3 Acronyms and abbreviations

SRS_SPALGeneral

=====

- Remove HIS from acronym table in chapter 3
- Set "Supporting Material:" of SRS_SPAL_12163 to –
- Remove section 7.2 "Related standards and norms"

SRS_SPIHandlerDriver

=====

- Remove abbreviation HIS from table in chapter 3 Acronyms and abbreviations

SRS_WatchdogDriver

=====

Remove from Chapter 7.2 the reference "[5] HIS API I/O Driver Specification www.automotive-his.de/results/API_IODriver_2.1.3.pdf".

SRS_ADCCDriver

=====

- Remove chapter 7.2 Related standards and norms.
- Remove abbreviation HIS from table in chapter 4 Acronyms and abbreviations

SRS_DIAGNOSTIC

=====

Remove in SRS_Diag_04098 the part of the sentence : "(according HIS [FL-504])"

SWS_BSWGeneral

=====

- Remove HIS from section 3.2 "Related standards and norms"
- Remove the sentence "The ID is the same as in HIS Software Supplier Identifications [19]." from the description of the line "<MIP>_VENDOR_ID" in SWS_BSW_00059

SWS_DiagnosticCommunicationManager

=====

Remove last 3 occurrences sentences in SWS_Dcm_00988:
"HIS representation of FL_ExtProgRequestType".
"HIS representation of FL_ApplicationUpdateType."
"HIS representation of FL_ResponseRequiredType."

SWS_DIODriver

=====

Remove chapter 3.2 Related standards and norms.

SWS_EEPROMDriver

=====

Remove chapter 3.2 Related standards and norms.

SWS_FlashDriver

=====

- remove SRS_Fls_12083 from traceability Matrix (should be done automatically by the update)

- remove HIS Flash Driver document from reference list

SWS_FlexRayDriver

=====

- Remove HIS from abbreviation table in chapter 2

SWS_FlexRayISOTransportLayer

=====

- Remove HIS from acronym table in chapter 2
- Remove reference to HIS MISRA subset from section 3.2

SWS_FlexRayTransceiverDriver

=====

- remove HIS from table Acronyms and abbreviations

SWS_OS

=====

- remove HIS from table in chapter 2 "Acronyms and abbreviations"
- remove chapter 3.2.2 "HIS" completely
- remove reference to HIS document in chapter 4.1 ("HIS Protected OSEK")

SWS_XCP

=====

- Remove abbreviation HIS from table in chapter 2 Acronyms and abbreviations
- Last change on issue 73567 comment 49-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.8 Specification Item SRS_BSW_00378

Trace References:

none

Content:

Type:	
Description:	For simple logical values and for API return values (if
Rationale:	Repeating requests of several WPs to define a boolean data type.
Applies to:	
Use Case:	API return value. Example:
Supporting Material:	Compiler vendors that provide a boolean data type that cannot be disabled have to change their compiler (i.e. make it ANSI C compliant).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.9 Specification Item SRS_BSW_00381

Trace References:

none

Content:

Type:	
Description:	The pre-compile time parameters shall be placed into a separate configuration header file.
Rationale:	Keep the configuration data separate.

Applies to:	
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)
–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.10 Specification Item SRS_BSW_00404

Trace References:

none

Content:

Type:	
Description:	Post-build configuration phase shall be supported. Post-build parameters are optional
Rationale:	Change ECU configuration after ECU production without an update of the whole application.
Applies to:	
Use Case:	type declaration of the Config Type

Supporting Material:	–
-----------------------------	---

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.11 Specification Item SRS_BSW_00406

Trace References:

RS_BRF_01136

Content:

Type:	
Description:	If the Default Error Tracer (DeT) Error is enabled, module APIs shall check if the module is initialized.
Rationale:	Wrong control flows shall be detected (and happen only) during integration phase. Therefore DeT must be called and stop execution if an uninitialized module is called.
Applies to:	
Use Case:	During optimization of init phase for fast startup, wrong init order has been configured and needs correction.

Supporting Material:	–
-----------------------------	---

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #75556: Incomplete Requirements in SRS_General

Problem description:

During completion of tracing I found that a requiement is missing in SRS_General.
I could not trace my requirement:

SWS_Det_00024:

If the Default Error Tracer has not been initialized before Det reporting functions are called, the reporting functions shall return immediately without any other action (no Error_Hook shall be used, no implementer specific function shall be performed and no error shall be reported

Against any of the requiements in the SRS_General.

The section on initailization is incomplete and should describe also the behaviour in case the init functions have not been called.

Furthermore the tracing of the init-function in SRS_BSW_00101 is missing to a feature.

Proposal: extend: SRS_BSW_00406 and complete tracing in SRS_BSW_00101

Agreed solution:

SRS_BSW_00406 - add:

v) If Det not initialized before reporting functions, it shall return immediately without any other action.

Requirement tracing SRS_BSW_00101- (RS_BRF_01136)

–Last change on issue 75556 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.12 Specification Item SRS_BSW_00411

Trace References:

RS_BRF_01028

Content:

Type:	
Description:	All AUTOSAR Basic Software Modules shall apply the following naming rule for enabling/disabling the existence of the API. It shall be configurable and the default value of the configuration shall be that this API is not available.
Rationale:	Enable/Disable the reading out of version information
Applies to:	
Use Case:	Example:
Supporting Material:	< MODULENAME > shall be derived from WP Architecture "List of Basic Software Modules", [DOC_MOD_LIST] (2...8 characters)

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.13 Specification Item SRS_BSW_00412

Trace References:

RS_BRF_01056

Content:

Type:	
Description:	References to c-configuration parameters (link time and post-build time) shall be placed into a separate h-file. The h-file shall be the same as pre-compile time parameters.
Rationale:	Put the references to c-configuration parameters in the same header file as pre-compile time parameters to enable access to the configuration data.
Applies to:	
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)
–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.14 Specification Item SRS_BSW_00429

Trace References:

none

Content:

Type:	
Description:	BSW modules are shall only be allowed to use certain OS objects and/or related OS services according to the following table:. The services and their access shall be defined in SWS_BSW_General.
Rationale:	Simplification of the OS integration of BSW modules.
Applies to:	
Use Case:	Integration of different BSW modules in one ECU.
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74366: [S[R/W]S BSWGeneral] Duplicate Requirement on SRS and SWS Level

Problem description:

SWS_BSW_00138 and SRS_BSW_00429 are duplicates of each other.
At least the contained table should exist only once.

Agreed solution:

SRS_BSW_00429 Access to OS is restricted

Type: Valid

Description: BSW modules shall only be allowed to use certain OS services. The services and their access shall be defined in SWS_BSW_General.

Rationale: Simplification of the OS integration of BSW modules.

Use Case: Integration of different BSW modules in one ECU.

Dependencies:

Supporting Material:

–Last change on issue 74366 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.15 Specification Item SRS_BSW_00447

Trace References:

none

Content:

Type:	
Description:	Data Type NvM_RequestResultType used in BSW C-API "NvM_GetErrorStatus" and in the AUTOSAR Interface "NvMService" operation GetErrorStatus (OUT NvM_RequestResultType RequestResultPtr); is same.
Rationale:	Standardizing Include Header file structure will allow common data types to be defined in RTE Types header files. This will avoid double and inconsistent definition of data types in both BSW and Software Component. This will also avoid type casts if SW-Cs are communicating with Autosar Services.
Applies to:	
Use Case:	All BSW Services which are called by Application SW-C and share data types. E.g.
Supporting Material:	Please see the Figure "Relationships between RTE Header Files" and related information in Chapter "RTE Modules" of RTE_SWS

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the en-

abling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.16 Specification Item SRS_BSW_00456

Trace References:

RS_BRF_01016

Content:

Type:	
Description:	If more than one implementation of a BSW Module is linked into an Autosar
Rationale:	BSW systems with multiple implementations of one BSW Module will mostly get an error at compile time or link time, if they are not harmonized.
Applies to:	
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the

filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.17 Specification Item SRS_BSW_00464

Trace References:

RS_BRF_01024

Content:

Type:	
Description:	File names shall be considered case sensitive regardless of the filesystem in which they are used.
Rationale:	Some file systems do not distinguish between file names spelled with the same letters but with different cases. Allowing such variability in the definitions can cause ambiguities.
Applies to:	
Use Case:	If different implementers implement modules using same names with different cases, the compile and link process shall have unpredictable results depending on the file system on which they are executed, leading eventually to errors (source or object file not found).
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the

filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.18 Specification Item SRS_BSW_00465

Trace References:

RS_BRF_01024

Content:

Type:	
Description:	It shall not be allowed to name any two files so that they only differ by the cases of their letters.
Rationale:	Problems deriving potentially ambiguous name definitions must be avoided already in the specification phase
Applies to:	
Use Case:	In a SWS the include files:
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77585: Life cycle changes for header file cleanup

Problem description:

Name: WP-A

Phone:

Role:

Description/Motivation:

For R4.4.0 a cleanup for the header files is planned. This will impact SRS requirements and therefore it is best to set relevant requirements in R4.3.1 either back to draft (indicating that they will be reworked) or obsolete (indicating that they will be dropped)

The following requirements are impacted:

- Set to draft for R4.3.1:

SRS_BSW_00456 "A Header file shall be defined in order to harmonize BSW Modules"

SRS_BSW_00300 "All AUTOSAR Basic Software Modules shall be identified by an unambiguous name"

SRS_BSW_00350 "All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors."

SRS_BSW_00411 "All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API"

SRS_BSW_00464 "File names shall be considered case sensitive regardless of the filesystem in which they are used"

SRS_BSW_00465 "It shall not be allowed to name any two files so that they only differ by the cases of their letters"

SRS_BSW_00158 "All modules of the AUTOSAR Basic Software shall strictly separate configuration from implementation"

SRS_BSW_00447 "Standardizing Include file structure of BSW Modules Implementing Autosar Service"

SRS_BSW_00348 "All AUTOSAR standard types and constants shall be placed and organized in a standard type header file"

SRS_BSW_00378 "AUTOSAR shall provide a boolean type"

SRS_BSW_00404 "BSW Modules shall support post-build configuration"

SRS_BSW_00389 "Containers shall have names "

- Set to obsolete for R4.3.1:

SRS_BSW_00380 Configuration parameters being stored in memory shall be placed into separate c-files

SRS_BSW_00419 If a pre-compile time configuration parameter is implemented as const it should be placed into a separate c-file

SRS_BSW_00381 The pre-compile time parameters shall be placed into a separate configuration header file

SRS_BSW_00412 References to c-configuration parameters shall be placed into a separate h-file

Agreed solution:

for R4.3.1:

- please update only the "Type" field of those requirements of the SRS BSW General document, where the "Type" field changed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

for R4.4.0:

- perform the real changes to the requirements proposed in the attachment (<https://www.autosar.org/bugzilla/attachment.cgi?id=4595>)

–Last change on issue 77585 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.19 Specification Item SRS_BSW_00484

Trace References:

[RS_BRF_01056](#)

Content:

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:	—

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\_SomelpXf\_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 Param-  
eters (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_LdComCbKCopyTxData_<sn>
BufReq_ReturnType Rte_LdComCbKCopyTxData_<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

1.20 Specification Item SRS_BSW_00485

Trace References:

RS_BRF_01056

Content:

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:	—

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 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\_SomelpXf\_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_LdComCbkCopyTxData_<sn>

BufReq_ReturnType Rte_LdComCbkCopyTxData_<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

1.21 Specification Item SRS_BSW_00486

Trace References:

RS_BRF_01056

Content:

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:	–

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\_zzzzzz: 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 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\_SomelpXf\_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 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* 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_LdComCbKCopyTxData_<sn>
BufReq_ReturnType Rte_LdComCbKCopyTxData_<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