

Document Title	SWS_EthernetDriver: 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	SWS_EthernetDriver	4
1.1	Specification Item ECUC_Eth_00006	4
1.2	Specification Item ECUC_Eth_00019	7
1.3	Specification Item ECUC_Eth_00045	8
1.4	Specification Item ECUC_Eth_00051	9
1.5	Specification Item ECUC_Eth_00062	10
1.6	Specification Item ECUC_Eth_00063	13
1.7	Specification Item SWS_Eth_00008	17
1.8	Specification Item SWS_Eth_00016	20
1.9	Specification Item SWS_Eth_00026	23
1.10	Specification Item SWS_Eth_00039	25
1.11	Specification Item SWS_Eth_00043	26
1.12	Specification Item SWS_Eth_00044	29
1.13	Specification Item SWS_Eth_00048	33
1.14	Specification Item SWS_Eth_00049	36
1.15	Specification Item SWS_Eth_00050	39
1.16	Specification Item SWS_Eth_00054	43
1.17	Specification Item SWS_Eth_00055	46
1.18	Specification Item SWS_Eth_00056	49
1.19	Specification Item SWS_Eth_00060	53
1.20	Specification Item SWS_Eth_00061	56
1.21	Specification Item SWS_Eth_00066	60
1.22	Specification Item SWS_Eth_00067	63
1.23	Specification Item SWS_Eth_00068	66
1.24	Specification Item SWS_Eth_00081	70
1.25	Specification Item SWS_Eth_00082	73
1.26	Specification Item SWS_Eth_00083	76
1.27	Specification Item SWS_Eth_00084	80
1.28	Specification Item SWS_Eth_00085	83
1.29	Specification Item SWS_Eth_00090	87
1.30	Specification Item SWS_Eth_00091	90
1.31	Specification Item SWS_Eth_00092	93
1.32	Specification Item SWS_Eth_00093	97
1.33	Specification Item SWS_Eth_00097	100
1.34	Specification Item SWS_Eth_00098	103
1.35	Specification Item SWS_Eth_00103	107
1.36	Specification Item SWS_Eth_00104	110
1.37	Specification Item SWS_Eth_00106	114
1.38	Specification Item SWS_Eth_00129	126
1.39	Specification Item SWS_Eth_00132	129

1.40	Specification Item SWS_Eth_00134	133
1.41	Specification Item SWS_Eth_00136	136
1.42	Specification Item SWS_Eth_00140	139
1.43	Specification Item SWS_Eth_00141	143
1.44	Specification Item SWS_Eth_00142	146
1.45	Specification Item SWS_Eth_00148	149
1.46	Specification Item SWS_Eth_00149	151
1.47	Specification Item SWS_Eth_00164	152
1.48	Specification Item SWS_Eth_00165	155
1.49	Specification Item SWS_Eth_00166	159
1.50	Specification Item SWS_Eth_00168	162
1.51	Specification Item SWS_Eth_00171	163
1.52	Specification Item SWS_Eth_00182	164
1.53	Specification Item SWS_Eth_00183	167
1.54	Specification Item SWS_Eth_00184	171
1.55	Specification Item SWS_Eth_00187	174
1.56	Specification Item SWS_Eth_00188	177
1.57	Specification Item SWS_Eth_00191	181
1.58	Specification Item SWS_Eth_00192	184
1.59	Specification Item SWS_Eth_00193	188
1.60	Specification Item SWS_Eth_00195	191
1.61	Specification Item SWS_Eth_00196	203
1.62	Specification Item SWS_Eth_00197	207
1.63	Specification Item SWS_Eth_00198	210
1.64	Specification Item SWS_Eth_00228	213
1.65	Specification Item SWS_Eth_00229	217
1.66	Specification Item SWS_Eth_00230	220
1.67	Specification Item SWS_Eth_00235	223
1.68	Specification Item SWS_Eth_00236	227
1.69	Specification Item SWS_Eth_00237	230
1.70	Specification Item SWS_Eth_00248	234
1.71	Specification Item SWS_Eth_00249	237
1.72	Specification Item SWS_Eth_00250	240
1.73	Specification Item SWS_Eth_00252	244
1.74	Specification Item SWS_Eth_00253	247
1.75	Specification Item SWS_Eth_00254	250

1 SWS_EthernetDriver

1.1 Specification Item ECUC_Eth_00006

Trace References:

none

Content:

Container Name	EthCtrlConfigEthCtrlConfig
Description	Configuration of the individual controller
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
EthCtrlEnableMii	ECUC_Eth_00012
EthCtrlEnableRxInterrupt	ECUC_Eth_00010
EthCtrlEnableTxInterrupt	ECUC_Eth_00011
EthCtrlIdx	ECUC_Eth_00007
EthCtrlMacLayerSpeed	ECUC_Eth_00063
EthCtrlMacLayerSubType	ECUC_Eth_00062
EthCtrlMacLayerType	ECUC_Eth_00039
EthCtrlPhyAddress	ECUC_Eth_00020

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
EthCtrlConfigEgress	1	Configuration of one Ethernet controller egress behavior.
EthCtrlConfigIngress	1	Configuration of one Ethernet controller ingress behavior.
EthDemEventParameterRefs	0..1	Container for the references to Dem EventParameter elements which shall be invoked using the API Dem_SetEvent Status in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's Dem EventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76343: Configuration parameters for Ethernet MAC layer types is incomplete

Problem description:

With https://www.autosar.org/bugzilla/show_bug.cgi?id=73074 the configuration of the MAC layer type was separated from the physical layer type. The configuration parameter `EthSwtPortMacLayerType` allows for the configuration of "families" of MAC layer types, namely xMII, xGMII, and xxGMII.

The concrete MAC layer type (e.g., SGMII vs. RGMII) however cannot be configured by this parameter. - Available switches (e.g., Broadcom's "Leo") however require the configuration of a *concrete* MAC layer type (and not only the configuration of a "family" of MAC layer types) for the individual port.

The same holds probably true for the Ethernet Driver and the Ethernet Transceiver driver.

One possibility to solve this is to introduce an additional config parameter named something like `EthSwtPortMacLayerSubType` with the following enum values "standard", "reduced", "reversed", "serial", "universal serial". - Thus for example the combination of `EthSwtPortMacLayerType=xGMII` and `EthSwtPortMacLayerSubType="reversed"` would yield RvGMII.

Agreed solution:

EthSwt:

add config parameter to container `EthSwtPort` named `EthSwtPortMacLayerSubType` with the following enum values "STANDARD: standard media-independent interface", "REDUCED: Reduced media-independent interface", "REVERSED: reversed media-independent interface (to provide direct connection between two Ethernet MACs)", "SERIAL: low-power and low pin-count serial 8b/10b-coded media-independent interface", "UNIVERSAL SERIAL: Universal low-power and low pin-count serial 8b/10b-coded media-independent interface".

add config parameter to container `EthSwtPort` named `EthSwtPortMacLayerSpeed` with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

All parameters shall have the same Multiplicity, Variant and config class as `EthSwtPortMacLayerType`.

add description for `EthSwt_PortMacLayerSubTypes`:
Defines the MAC layer subtype of a switch port

add description for EthSwt_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

Eth:

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthCtrlMacLayerType=xGMII and EthCtrlMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for Eth_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for Eth_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

EthTrcv:

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthTrcvPortMacLayerType=xGMII and EthTrcvPortMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for EthTrcv_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for EthTrcv_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

No upstream Mapping as this decision is made on ECU-Configuration.
–Last change on issue 76343 comment 30–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item ECUC_Eth_00019

Trace References:

none

Content:

Name	EthUpdatePhysAddrFilterEthGeneral.EthUpdatePhysAddrFilter		
Description	Enables/Disables optional API Eth_UpdatePhysAddrFilter. Tags: atp.Status=obsolete		
Multiplicity	1 0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76418: [Eth] Superfluous parameter EthUpdatePhysAddrFilter

Problem description:

The configuration parameter EthUpdatePhysAddrFilter is superfluous, since the API seems not to be optional anymore.

Also the corresponding API in EthIf is not optional.

Therefore, the configuration parameter EthUpdatePhysAddrFilter should be removed.

–Last change on issue 76418 comment 14–

Agreed solution:

set to obsolete ECUC_Eth_00019

–Last change on issue 76418 comment 11–

BW-C-Level:

Application	Specification	Bus
1	3	1

1.3 Specification Item ECUC_Eth_00045

Trace References:

none

Content:

Name	EthCtrlConfigIngressFifoBufLenByte		
Description	Fifo buffer length Length of Fifo elements in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	—		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76274: [Eth] Reformulate the description of EthCtrlConfig[Ingress|Egress]FifoBufLenByte

Problem description:

The description could lead to an misunderstanding. E.g.

Param : "EthCtrlConfigIngressFifoBufLenByte "

Description: "Fifo buffer length in bytes."

The parameter is used to configure the length of one element in the fifo. The formulation could be interpreted as the complete length of the fifo. What do you think?

Agreed solution:

Reformulate the following description
from "Fifo buffer length in bytes."

to "Length of Fifo elements in bytes."

for the following parameters:

EthCtrlConfigEgressFifoBufLenByte

EthCtrlConfigIngressFifoBufLenByte

–Last change on issue 76274 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.4 Specification Item ECUC_Eth_00051

Trace References:

none

Content:

Name	EthCtrlConfigEgressFifoBufLenByte		
Description	Fifo buffer length Length of Fifo elements in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76274: [Eth] Reformulate the description of EthCtrlConfig[Ingress|Egress]FifoBufLenByte

Problem description:

The description could lead to an misunderstanding. E.g.

Param : "EthCtrlConfigIngressFifoBufLenByte "

Description: "Fifo buffer length in bytes."

The parameter is used to configure the length of one element in the fifo. The

formulation could be interpreted as the complete length of the fifo. What do you think?

Agreed solution:

Reformulate the following description
from "Fifo buffer length in bytes."
to "Length of Fifo elements in bytes."

for the following parameters:
EthCtrlConfigEgressFifoBufLenByte
EthCtrlConfigIngressFifoBufLenByte
–Last change on issue 76274 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.5 Specification Item ECUC_Eth_00062

Trace References:

none

Content:

Name	EthCtrlMacLayerSubTypeEthCtrlConfig.EthCtrlMacLayerSubType
Description	Defines the MAC layer subtype of a switch port
Multiplicity	0..1
Type	EcucEnumerationParamDef

Range	REDUCEDEthCtrlConfig.EthCtrlMacLayerSubType.REDUCED	—	
	REVERSEDEthCtrlConfig.EthCtrlMacLayerSubType.REVERSED	—	
	SERIEthCtrlConfig.EthCtrlMacLayerSubType.SERIAL	—	
	STANDARDEthCtrlConfig.EthCtrlMacLayerSubType.STANDARD	—	
	UNIVERSAL_SERIEthCtrlConfig.EthCtrlMacLayerSubType.UNIVERSAL_SERIAL	—	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	—	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	—	
Scope / Dependency	scope: ECU		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76343: Configuration parameters for Ethernet MAC layer types is incomplete

Problem description:

With https://www.autosar.org/bugzilla/show_bug.cgi?id=73074 the configuration of the MAC layer type was separated from the physical layer type. The configuration parameter EthSwtPortMacLayerType allows for the configuration of "families" of MAC layer types, namely xMII, xGMII, and xxGMII.

The concrete MAC layer type (e.g., SGMII vs. RGMII) however cannot be configure by this parameter. - Available switches (e.g., Broadcom's "Leo") however require the configuration of a *concrete* MAC layer type (and not only the configuration of a "family" of MAC layer types) for the individual port.

The same holds probably true for the Ethernet Driver and the Ethernet Transceiver driver.

One possibility to solve this is to introduce an additional config parameter named something like EthSwtPortMacLayerSubType with the following enum values "standard", "reduced", "reversed", "serial", "universal serial". - Thus for example the combination of EthSwtPortMacLayerType=xGMII and EthSwtPortMacLayerSubType="reversed" would yield RvGMII.

Agreed solution:

EthSwt:

add config parameter to container EthSwtPort named EthSwtPortMacLayerSubType with the following enum values "STANDARD: standard media-independent interface", "REDUCED: Reduced media-independent interface", "REVERSED: reversed media-independent interface (to provide direct connection between two Ethernet MACs)", "SERIAL: low-power and low pin-count serial 8b/10b-coded media-independent interface", "UNIVERSAL SERIAL: Universal low-power and low pin-count serial 8b/10b-coded media-independent interface".

add config parameter to container EthSwtPort named EthSwtPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

add description for EthSwt_PortMacLayerSubTypes:

Defines the MAC layer subtype of a switch port

add description for EthSwt_PortMacLayerSpeed:

Defines the baud rate of the MAC layer

Eth:

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthCtrlMacLayerType=xGMII and EthCtrlMacLayerSubType ="REVERSED" would yield RvGMII.

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for Eth_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for Eth_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

EthTrcv:

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthTrcvPortMacLayerType=xGMII and EthTrcvPortMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for EthTrcv_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for EthTrcv_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

No upstream Mapping as this decision is made on ECU-Configuration.
–Last change on issue 76343 comment 30–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.6 Specification Item ECUC_Eth_00063

Trace References:

none

Content:

Name	EthCtrlMacLayerSpeedEthCtrlConfig.EthCtrlMacLayerSpeedin container EthCtrlConfig		
Description	Defines the baud rate of the MAC layer.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ETH_MAC_LAYER_SPEED_100MEthCtrlConfig.EthCtrlMacLayerSpeed.ETH_MAC_LAYER_SPEED_100M		
	ETH_MAC_LAYER_SPEED_10GEthCtrlConfig.EthCtrlMacLayerSpeed.ETH_MAC_LAYER_SPEED_10G		
	ETH_MAC_LAYER_SPEED_10MEthCtrlConfig.EthCtrlMacLayerSpeed.ETH_MAC_LAYER_SPEED_10M		
	ETH_MAC_LAYER_SPEED_1GEthCtrlConfig.EthCtrlMacLayerSpeed.ETH_MAC_LAYER_SPEED_1G		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	—	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	—	
Scope / Dependency	scope: ECU		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76343: Configuration parameters for Ethernet MAC layer types is incomplete

Problem description:

With https://www.autosar.org/bugzilla/show_bug.cgi?id=73074 the configuration of the MAC layer type was separated from the physical layer type. The configuration parameter EthSwtPortMacLayerType allows for the configuration of "families" of MAC layer types, namely xMII, xGMII, and xxGMII.

The concrete MAC layer type (e.g., SGMII vs. RGMII) however cannot be configured by this parameter. - Available switches (e.g., Broadcom's "Leo") however require the configuration of a *concrete* MAC layer type (and not only the configuration of a "family" of MAC layer types) for the individual port.

The same holds probably true for the Ethernet Driver and the Ethernet Transceiver driver.

One possibility to solve this is to introduce an additional config parameter named something like `EthSwtPortMacLayerSubType` with the following enum values "standard", "reduced", "reversed", "serial", "universal serial". - Thus for example the combination of `EthSwtPortMacLayerType=xGMII` and `EthSwtPortMacLayerSubType="reversed"` would yield RvGMII.

Agreed solution:

EthSwt:

add config parameter to container `EthSwtPort` named `EthSwtPortMacLayerSubType` with the following enum values "STANDARD: standard media-independent interface", "REDUCED: Reduced media-independent interface", "REVERSED: reversed media-independent interface (to provide direct connection between two Ethernet MACs)", "SERIAL: low-power and low pin-count serial 8b/10b-coded media-independent interface", "UNIVERSAL SERIAL: Universal low-power and low pin-count serial 8b/10b-coded media-independent interface".

add config parameter to container `EthSwtPort` named `EthSwtPortMacLayerSpeed` with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

All parameters shall have the same Multiplicity, Variant and config class as `EthSwtPortMacLayerType`.

add description for `EthSwt_PortMacLayerSubTypes`:
Defines the MAC layer subtype of a switch port

add description for `EthSwt_PortMacLayerSpeed`:
Defines the baud rate of the MAC layer

Eth:

add config parameter to container `EthCtrlConfig` named `EthCtrlMacLayerSubType` with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of `EthC-`

trlMacLayerType=xGMII and EthCtrlMacLayerSubType = "REVERSED" would yield RvGMII.

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for Eth_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for Eth_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

EthTrcv:

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthTrcvPortMacLayerType=xGMII and EthTrcvPortMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for EthTrcv_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for EthTrcv_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

No upstream Mapping as this decision is made on ECU-Configuration.
–Last change on issue 76343 comment 30–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.7 Specification Item SWS_Eth_00008

Trace References:

none

Content:

In case **default development** error detection is enabled for the Ethernet Driver module: The Ethernet Driver module shall check API parameters for validity and report detected errors to the DET.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]"

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

- "[SWS_xxx_yyyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyyy]

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

- "[SWS_xxx_yyyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.8 Specification Item SWS_Eth_00016

Trace References:

none

Content:

Type or error	Relevance	Related error code	Value [hex]
Invalid controller index	Default Development error	ETH_E_INV_CTRL_IDX	0x01
Eth module or controller was not initialized	Default Development error	ETH_E_NOT_INITIALIZED UNINIT	0x02
Invalid pointer in parameter list	Default Development error	ETH_E_PARAM_POINTER	0x03
Invalid parameter	Default Development error	ETH_E_INV_PARAM	0x04
Invalid mode	Default Development error	ETH_E_INV_MODE	0x05

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

Solution for SWS_RTE:

- SWS_RTE —
 - Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.9 Specification Item SWS_Eth_00026

Trace References:

SRS_Eth_00127

Content:

Module	Imported Type
ComStack_Types	BufReq_ReturnType
Dem	Dem_EventIdType
	Dem_EventStatusType
Eth_GeneralTypes	Eth_BufIdxType
	Eth_ConfigType
	Eth_CounterType
	Eth_DataType
	Eth_FilterActionType
	Eth_FrameType
	Eth_ModeType
	Eth_RateRatioType
	Eth_RxStatsType
	Eth_RxStatusType
	Eth_TimeIntDiffType
	Eth_TimeStampQualType
	Eth_TimeStampType
	Eth_TxErrorCounterValuesType
Eth_TxStatsType	
Std_Types	Std_ReturnType
	Std_VersionInfoType

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76983: [CRYPTO] incorrect specification of Crypto_JobPrimitiveInfoType

Problem description:

In SWS_Csm_01012 the elements "processingType" and "callbackUpdateNotification" are not correctly specified.

It seems that the data type specification of the one element is swapped with the data type specification of the other element.

Agreed solution:

[SWS_Csm_01012]:

change:

processingType const Crypto_ProcessingType Determines the synchronous or asynchronous behavior.

callbackUpdateNotification const boolean Indicates, whether the callback function

shall be called, if the UPDATE operation has finished.
–Last change on issue 76983 comment 2–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.10 Specification Item SWS_Eth_00039

Trace References:

none

Content:

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS otherwise (if DET is disabled) return E_NOT_OK, otherwise pass the production error ETH_E_ACCESS and return E_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76104: [ETH] Incorrect requirement for Eth_Init return error

Problem description:

The requirement [SWS_Eth_00039]:The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS otherwise (if DET is disabled) return E_NOT_OK, otherwise pass the production error ETH_E_ACCESS and return E_OK.

is not correct because:

- Function Eth_Init is not have return value (according to SWS_Eth_00027)
- The production error ETH_E_ACCESS should be report to DEM and should not depend on DET configuration (according to SWS_Eth_00173)

A proposal for rewording may be:

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS(if DEM is enabled)

–Last change on issue 76104 comment 14–

Agreed solution:

[SWS_Eth_00039]

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS.

[SWS_Eth_00168]

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS and return E_NOT_OK.

–Last change on issue 76104 comment 11–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.11 Specification Item SWS_Eth_00043

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZED UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.12 Specification Item SWS_Eth_00044

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

– SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

- In chapter "7 References":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.13 Specification Item SWS_Eth_00048

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZED_UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

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

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.14 Specification Item SWS_Eth_00049

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

correct description:

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

– SWS_RTE –

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.15 Specification Item SWS_Eth_00050

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlModePtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_XXX_YYYYY]"

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

- "[SWS_XXX_YYYYY]"

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.16 Specification Item SWS_Eth_00054

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance

Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.17 Specification Item SWS_Eth_00055

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

Solution for SWS_RTE:

- SWS_RTE —
 - Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.18 Specification Item SWS_Eth_00056

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter PhysAddrPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]"

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

- "[SWS_xxx_yyyyy]"

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

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

- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
- Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
- Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
-Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.19 Specification Item SWS_Eth_00060

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.20 Specification Item SWS_Eth_00061

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

- In chapter "7 References":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.21 Specification Item SWS_Eth_00066

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyy]"

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.22 Specification Item SWS_Eth_00067

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyyy]

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

- "[SWS_xxx_yyyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyyy]

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

- "[SWS_xxx_yyyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.23 Specification Item SWS_Eth_00068

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter RegValPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_XXX_YYYYY]"

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

- "[SWS_XXX_YYYYY]"

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.24 Specification Item SWS_Eth_00081

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZED_UNINIT and return BUFREQ_E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance

Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.25 Specification Item SWS_Eth_00082

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX and return BUFREQ_E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.26 Specification Item SWS_Eth_00083

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter BufIdxPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER and return BUFREQ_E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]"

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

- "[SWS_xxx_yyyyy]"

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

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

- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
-Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.27 Specification Item SWS_Eth_00084

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter BufPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER and return BUFREQ_E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.28 Specification Item SWS_Eth_00085

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter LenBytePtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER and return BUFREQ_E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

- In chapter "7 References":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.29 Specification Item SWS_Eth_00090

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZED_UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

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

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.30 Specification Item SWS_Eth_00091

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

correct description:

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

– SWS_RTE –

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.31 Specification Item SWS_Eth_00092

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter BufIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_PARAM otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_XXX_YYYYY]"

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

- "[SWS_XXX_YYYYY]"

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.32 Specification Item SWS_Eth_00093

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter PhysAddrPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance

Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.33 Specification Item SWS_Eth_00097

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

Solution for SWS_RTE:

- SWS_RTE —
 - Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.34 Specification Item SWS_Eth_00098

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter `CtrlIdx` for being valid. If the check fails, the function shall raise the **default development** error `ETH_E_INV_CTRL_IDX`.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]"

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

- "[SWS_xxx_yyyyy]"

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

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

- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
- Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
- Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
-Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.35 Specification Item SWS_Eth_00103

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.36 Specification Item SWS_Eth_00104

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_XXX_yyyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

- In chapter "7 References":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.37 Specification Item SWS_Eth_00106

Trace References:

none

Content:

Service name:	Eth_GetVersionInfoEth_GetVersionInfo	
Syntax:	void Eth_GetVersionInfo(Std_VersionInfoType* VersionInfoPtr)	
Service ID[hex]:	0xD	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	VersionInfoPtrEth_GetVersion Info.VersionInfoPtr	Version information of this module
Parameters (inout):	None	
Parameters (out):	None	
Return value:	void	None
Description:	Returns the version information of this module	

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_LdComCbkJCopyTxData_<sn>

BufReq_ReturnType Rte_LdComCbkJCopyTxData_<sn>(const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType Rte_LdComCbkJCopyTxData_<sn>(const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

Specification of Lin

~~~~~

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

## Specification of PduR

~~~~~

* PduR_<User:LoTp>CopyTxData
add const to "RetryInfoType* retry"

Specification of J1939Nm

~~~~~

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

## Specification of SoAd

~~~~~

=> everything already fixed with RfC 65633

Specification of SPIHandlerDriver

~~~~~

==> nothing to change for SWS SPI

## Specification of SynchronizedTimeBaseManager

~~~~~

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

Specification of Tcplp

~~~~~

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

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

=> everything else already fixed with RfC 65633

## Specification of TimeSyncOverFlexRay

~~~~~

"Change SWS_FrTSyn_00064: parameter versioninfo of type Std_VersionInfoType* is marked wrongly as IN. Change to OUT"

Specification of EFX

~~~~~

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

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

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

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

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

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

~ [SWS\_Efx\_00307] Efx\_RampGetSwitchPos: Include constant for pointer

Input-parameter as like below.

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

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

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

### Specification of MFL

~~~~~

~ [SWS_Mfl_00192] Mfl_Debounce_u8_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl_Debounce_u8_u8(boolean X, Mfl_DebounceState_Type* State, const Mfl_DebounceParam_Type* Param, float32 dT)

~ [SWS_Mfl_00266] Mfl_DebounceInit: The parameter Mfl_DebounceState_Type* State should be Out instead of In parameter as like below.

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

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

~ [SWS_Mfl_00246] Mfl_HystDeltaRight_f32_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl_HystDeltaRight_f32_u8(float32 X, float32 Delta, float32 Rsp, const uint8* State)

~ [SWS_Mfl_00285] Mfl_MedianSort_f32_f32: The parameter Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS_Mfl_00037] Mfl_PT1SetState: The parameter State_cpst should be Out instead of In parameter as like below.

Parameters (in): X1_f32 Initial value for input state

Y1_f32 Initial value for output state

Parameters (out): State_cpst Pointer to internal state structure

~ [SWS_Mfl_00225] Mfl_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Mfl_RampCheckActivity(const Mfl_StateRamp_Type* State_cpst)

~ [SWS_Mfl_00223] Mfl_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Mfl_RampGetSwitchPos(const Mfl_StateRamp_Type* State_cpst)
–Last change on issue 68035 comment 135–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.38 Specification Item SWS_Eth_00129

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the controller mode for being active (ETH_MODE_ACTIVE). If the check fails, the function shall raise the **default development** error ETH_E_INV_MODE otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyyy]

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

- "[SWS_xxx_yyyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyyy]

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

- "[SWS_xxx_yyyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development

error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

- In chapter "7 References":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.39 Specification Item SWS_Eth_00132

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the controller mode for being active (ETH_MODE_ACTIVE). If the check fails, the function shall raise the **default development** error ETH_E_INV_MODE.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

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

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.40 Specification Item SWS_Eth_00134

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the controller mode for being active (ETH_MODE_ACTIVE). If the check fails, the function shall raise the **default development** error ETH_E_INV_MODE.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

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

- "[SWS_xxx_yyyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyyy]

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

- "[SWS_xxx_yyyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.41 Specification Item SWS_Eth_00136

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter VersionInfoPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.42 Specification Item SWS_Eth_00140

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_XXX_YYYYY]"

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

- "[SWS_XXX_YYYYY]"

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.43 Specification Item SWS_Eth_00141

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance

Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.44 Specification Item SWS_Eth_00142

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter PhysAddrPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

Solution for SWS_RTE:

- SWS_RTE —
 - Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

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

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.45 Specification Item SWS_Eth_00148

Trace References:

none

Content:

Eth.h shall include Eth_GeneralTypes.h for the **include inclusion** of general Eth type declarations . **listed in SWS_Eth_00026.**

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74677: [Eth] Eth_GeneralTypes.h must not include Eth_Cfg.h

Problem description:

According to Figure 5.1 "Figure 5.1 Ethernet Driver file structure" Eth_GeneralTypes.h shall include Eth_Cfg.h.

This is incorrect for the following reasons:

- * This way Eth_GeneralTypes.h will be dependent on the configuration of the Ethernet driver(s). - This must not be the case!
- * There may be multiple Ethernet drivers within one ECU project. Each of these Ethernet drivers will come with its own Eth_<vi>_<ai>_Cfg.h file. Thus Eth_GeneralTypes.h would need to include each of them.

Furtheron SWS_Eth_00026 states that Eth_GeneralTypes will provide Eth_ConfigType which is incorrect, since Eth_ConfigType is again Ethernet driver specific (might be different for different Ethernet drivers) and thus will probably be provided by Eth_Types.h.

Last but not least SWS_Eth_00149 states that "The types specified in SWS_EthernetDriver shall be declared in Eth_GeneralTypes.h.". Here Eth_ConfigType shall be excluded since this shall not be provided by Eth_GeneralTypes.h.

Agreed solution:

Modify Figure 5.1 "Figure 5.1 Ethernet Driver file structure" in a way that

- * Eth_Types.h directly includes Eth_Cfg.h
- * Eth_GeneralTypes.h no longer includes Eth_Cfg.h

Remove Eth_ConfigType from the table of SWS item SWS_Eth_00026.

Rephrase SWS_Eth_00148 from "Eth.h shall include Eth_GeneralTypes.h for the include of general Eth type declarations." to "Eth.h shall include Eth_GeneralTypes.h for the include of general Eth type declarations listed in SWS_Eth_00026."

Remove SWS_Eth_00149

–Last change on issue 74677 comment 7–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.46 Specification Item SWS_Eth_00149**Trace References:**

none

Content:

The types specified in SWS_EthernetDriver shall be declared in Eth_GeneralTypes.h.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74677: [Eth] Eth_GeneralTypes.h must not include Eth_Cfg.h

Problem description:

According to Figure 5.1 "Figure 5.1 Ethernet Driver file structure" Eth_GeneralTypes.h shall include Eth_Cfg.h.

This is incorrect for the following reasons:

- * This way Eth_GeneralTypes.h will be dependent on the configuration of the Ethernet driver(s). - This must not be the case!
- * There may be multiple Ethernet drivers within one ECU project. Each of these Ethernet drivers will come with its own Eth_<vi>_<ai>_Cfg.h file. Thus Eth_GeneralTypes.h would need to include each of them.

Furtheron SWS_Eth_00026 states that Eth_GeneralTypes will provide Eth_ConfigType which is incorrect, since Eth_ConfigType is again Ethernet driver specific (might be different for different Ethernet drivers) and thus will probably be provided by Eth_Types.h.

Last but not least SWS_Eth_00149 states that "The types specified in SWS_EthernetDriver shall be declared in Eth_GeneralTypes.h.". Here Eth_ConfigType shall be excluded since this shall not be provided by Eth_GeneralTypes.h.

Agreed solution:

Modify Figure 5.1 "Figure 5.1 Ethernet Driver file structure" in a way that

- * Eth_Types.h directly includes Eth_Cfg.h

* Eth_GeneralTypes.h no longer includes Eth_Cfg.h

Remove Eth_ConfigType from the table of SWS item SWS_Eth_00026.

Rephrase SWS_Eth_00148 from "Eth.h shall include Eth_GeneralTypes.h for the include of general Eth type declarations." to "Eth.h shall include Eth_GeneralTypes.h for the include of general Eth type declarations listed in SWS_Eth_00026."

Remove SWS_Eth_00149

–Last change on issue 74677 comment 7–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.47 Specification Item SWS_Eth_00164

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.48 Specification Item SWS_Eth_00165

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]"

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

- "[SWS_xxx_yyyyy]"

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.49 Specification Item SWS_Eth_00166

Trace References:

none

Content:

If **default development** error detection is enabled the function shall check the parameter PhysAddrPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

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

- "[SWS_xxx_yyyyy]

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

- "[SWS_xxx_yyyyy]

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

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

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

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

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance

Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

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

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

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

SWS_MFXLibrary:

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

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.50 Specification Item SWS_Eth_00168

Trace References:

none

Content:

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS and return E_NOT_OK, **otherwise pass the production error ETH_E_ACCESS and return E_OK.**

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76104: [ETH] Incorrect requirement for Eth_Init return error

Problem description:

The requirement [SWS_Eth_00039]:The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS otherwise (if DET is disabled) return E_NOT_OK, otherwise pass the production error ETH_E_ACCESS and return E_OK.

is not correct because:

- Function Eth_Init is not have return value (according to SWS_Eth_00027)
- The production error ETH_E_ACCESS should be report to DEM and should not depend on DET configuration (according to SWS_Eth_00173)

A proposal for rewording may be:

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS(if DEM is enabled)

–Last change on issue 76104 comment 14–

Agreed solution:

[SWS_Eth_00039]

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS.

[SWS_Eth_00168]

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETH_E_ACCESS and return E_NOT_OK.

–Last change on issue 76104 comment 11–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.51 Specification Item SWS_Eth_00171

Trace References:

none

Content:

Service name:	Eth_MainFunctionEth_MainFunction
Syntax:	void Eth_MainFunction(void)
Service ID[hex]:	0x0a 0x20
Description:	The function checks for controller errors and lost frames. Used for polling state changes. Calls EthIf_CtrlModeIndication when the controller mode changed.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76202: [ETH] Service ID for Eth_MainFunction API conflicts with Eth_Transmit API

Problem description:

Service ID for Eth_MainFunction API is mentioned as 0x0a , but the same is used for Eth_Transmit API .New service ID should be provided for Eth_MainFunction API

Agreed solution:

SWS_Eth_00171

Service ID[hex]: 0x20

–Last change on issue 76202 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.52 Specification Item SWS_Eth_00182

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

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

Problem description:

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

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

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

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

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

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

Agreed solution:

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

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.53 Specification Item SWS_Eth_00183

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]"

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.54 Specification Item SWS_Eth_00184

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter timeQualPtr and timeStampPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"
- "[SWS_xxx_yyyyyy]
- In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"
- "[SWS_xxx_yyyyyy]
- If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"
- "In case default errors are enabled,..."
- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyyy]
- In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"
- "[SWS_xxx_yyyyyy]
- If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"
- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance

Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.55 Specification Item SWS_Eth_00187

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_ **NOT_INITIALIZEDUNINIT**.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —
 - Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
 - In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
- | | | |
|---|----|----------------------------------|
| Rename "AUTOSAR_SWS_DevelopmentErrorTracer" | to | "AUTOSAR_SWS_DefaultErrorTracer" |
|---|----|----------------------------------|

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.56 Specification Item SWS_Eth_00188

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter `CtrlIdx` for being valid. If the check fails, the function shall raise the **default development** error `ETH_E_INV_CTRL_IDX`.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters `*DevErrorDetect` are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]"

If default error detection is enabled: the function shall check that the service `xxx_Init` was previously called. If the check fails, the function shall raise the default error `XXX_E_NOT_INITIALIZED` otherwise (if DET is disabled) return `E_NOT_OK`. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_XXX_YYYYY]"

In case development error detection is enabled for the XXXX module: The XXXX module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_XXX_YYYYY]"

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
- Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
- Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
- Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
-Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.57 Specification Item SWS_Eth_00191

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.58 Specification Item SWS_Eth_00192

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_XXX_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

– SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.59 Specification Item SWS_Eth_00193

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter `timeQualPtr` and `timeStampPtr` for being valid. If the check fails, the function shall raise the **default development** error `ETH_E_PARAM_POINTER`.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters `*DevErrorDetect` are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.60 Specification Item SWS_Eth_00195

Trace References:

none

Content:

Service name:	Eth_GetIngressTimeStampEth_GetIngressTimeStamp	
Syntax:	<pre>void Eth_GetIngressTimeStamp(uint8 CtrlIdx, const Eth_DataType* DataPtr, Eth_TimeStampQualType* timeQualPtr, Eth_TimeStampType* timeStampPtr)</pre>	
Service ID[hex]:	0x19	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	CtrlIdxEth_GetIngressTimeStamp.CtrlIdx	Index of the addresses ETH controller.
	DataPtrEth_GetIngressTimeStamp.DataPtr	Pointer to the message buffer, where Application expects ingress time stamping
Parameters (inout):	None	
Parameters (out):	timeQualPtrEth_GetIngressTimeStamp.timeQualPtr	quality of HW time stamp, e.g. based on current drift
	timeStampPtrEth_GetIngressTimeStamp.timeStampPtr	current time stamp
Return value:	None	
Description:	Reads back the ingress time stamp on a dedicated message object. It must be called within the RxIndication() function.	

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 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 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* Authenticated, uint8* AuthenticationReqData, uint8* AuthenticationResData)

To:

Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed, const uint8* ConfirmationReqData, uint8* ConfirmationResData)

Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authenticated, const uint8* AuthenticationReqData, uint8* AuthenticationResData)

Specification of E2ELibrary

~~~~~

no change required

#### Specification of Eth

~~~~~

no change required

Specification of EthIf

~~~~~

no change required

#### Specification of EthSwitchDriver

~~~~~

no change required

Specification of ICUDriver

~~~~~

SWS\_Icu\_00201: Icu\_StartTimestamp

Parameter (IN): Icu\_ValueType\* BufferPtr shall be changed to Parameters (out) type

#### Specification of LdCom

~~~~~

[SWS_LDCOM_00027]: LdCom_CopyTxData

BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to

BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

[SWS_LDCOM_00036]: Rte_LdComCbkJCopyTxData_<sn>

BufReq_ReturnType Rte_LdComCbkJCopyTxData_<sn>(const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to

BufReq_ReturnType Rte_LdComCbkJCopyTxData_<sn>(const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

Specification of Lin

~~~~~

PduInfoPtr needs to be const in Std\_ReturnType Lin\_SendFrame( uint8 Channel, const Lin\_PduType\* PduInfoPtr )



### Specification of PduR

~~~~~

* PduR_<User:LoTp>CopyTxData
add const to "RetryInfoType* retry"

Specification of J1939Nm

~~~~~

Change parameter 'name' of User\_AddressClaimedIndication to type 'const uint8''

### Specification of SoAd

~~~~~

=> everything already fixed with RfC 65633

Specification of SPIHandlerDriver

~~~~~

==> nothing to change for SWS SPI

### Specification of SynchronizedTimeBaseManager

~~~~~

"StbM not affected. All issues listed in the WP-A attachment have been already implemented by IT 69124 in context of RfC 65633"

Specification of Tcplp

~~~~~

~[SWS\_TCPIP\_00040] Tcplp\_DhcpReadOption: change DataPtr from (IN) to (OUT)

~[SWS\_TCPIP\_00189] Tcplp\_DhcpV6ReadOption: change DataPtr from (IN) to (OUT)

=> everything else already fixed with RfC 65633

### Specification of TimeSyncOverFlexRay

~~~~~

"Change SWS_FrTSyn_00064: parameter versioninfo of type Std_VersionInfoType* is marked wrongly as IN. Change to OUT"

Specification of EFX

~~~~~

~ [SWS\_Efx\_00355] Efx\_Debounce\_u8\_u8: Include constant for pointer Input-parameter as like below.

uint8 Efx\_Debounce\_u8\_u8( boolean X, Efx\_DebounceState\_Type \* State, const Efx\_DebounceParam\_Type \* Param, sint32 dT )

~ [SWS\_Efx\_00376] Efx\_MedianSort: The parameter <InType>\* Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS\_Efx\_00309] Efx\_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Efx\_RampCheckActivity(const Efx\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Efx\_00307] Efx\_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Efx\_RampGetSwitchPos(const Efx\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Efx\_00193] Efx\_Array\_Average: Include constant for pointer Input-parameter as like below.

<OutType> Efx\_Array\_Average\_<InTypeMn>\_<OutTypeMn>( const <InType>\* Array, uint16 Count)

## Specification of MFL

~~~~~

~ [SWS_Mfl_00192] Mfl_Debounce_u8_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl_Debounce_u8_u8(boolean X, Mfl_DebounceState_Type* State, const Mfl_DebounceParam_Type* Param, float32 dT)

~ [SWS_Mfl_00266] Mfl_DebounceInit: The parameter Mfl_DebounceState_Type* State should be Out instead of In parameter as like below.

Parameters (in): X Initial value for the input state

Parameters (out): State Pointer to structure for debouncing state variables

~ [SWS_Mfl_00246] Mfl_HystDeltaRight_f32_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl_HystDeltaRight_f32_u8(float32 X, float32 Delta, float32 Rsp, const uint8* State)

~ [SWS_Mfl_00285] Mfl_MedianSort_f32_f32: The parameter Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS_Mfl_00037] Mfl_PT1SetState: The parameter State_cpst should be Out instead of In parameter as like below.

Parameters (in): X1_f32 Initial value for input state

Y1_f32 Initial value for output state

Parameters (out): State_cpst Pointer to internal state structure

~ [SWS_Mfl_00225] Mfl_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Mfl_RampCheckActivity(const Mfl_StateRamp_Type* State_cpst)

~ [SWS_Mfl_00223] Mfl_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Mfl_RampGetSwitchPos(const Mfl_StateRamp_Type* State_cpst)

–Last change on issue 68035 comment 135–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.61 Specification Item SWS_Eth_00196

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!))

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.62 Specification Item SWS_Eth_00197

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.63 Specification Item SWS_Eth_00198

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter DataPtr, timeQualPtr and timeStampPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.64 Specification Item SWS_Eth_00228

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZED_UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_XXX_YYYYY]"

In case default error detection is enabled for the XXXX module: The XXXX module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_XXX_YYYYY]"

If default error detection is enabled: the function shall check that the service XXX_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.65 Specification Item SWS_Eth_00229

Trace References:

none

Content:

If dev **default development** elopment error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **de-fault development** error ETH_E_INV_CTRL_IDX otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error

tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"
- "[SWS_xxx_yyyyyy]"
- In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"
- "[SWS_xxx_yyyyyy]"
- If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"
- "In case default errors are enabled,..."
- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyyy]"
- In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"
- "[SWS_xxx_yyyyyy]"
- If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"
- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

TOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.66 Specification Item SWS_Eth_00230

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CounterPtr for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return

E_NOT_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —
 - Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
 - In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
- | | | |
|---|----|----------------------------------|
| Rename "AUTOSAR_SWS_DevelopmentErrorTracer" | to | "AUTOSAR_SWS_DefaultErrorTracer" |
|---|----|----------------------------------|

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.67 Specification Item SWS_Eth_00235

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZED UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]"

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error

XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."
- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents":
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents":
Rename "AU-

TOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.68 Specification Item SWS_Eth_00236

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET

does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"
- "[SWS_xxx_yyyyy]
- In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"
- "[SWS_xxx_yyyyy]
- If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"
- "In case default errors are enabled,..."
- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyy]
- In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"
- "[SWS_xxx_yyyyy]
- If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"
- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

errors"

- Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
 - In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
- | | | |
|---|----|----------------------------------|
| Rename "AUTOSAR_SWS_DevelopmentErrorTracer" | to | "AUTOSAR_SWS_DefaultErrorTracer" |
|---|----|----------------------------------|

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
- | | | |
|---|----|----------------------------------|
| Rename "AUTOSAR_SWS_DevelopmentErrorTracer" | to | "AUTOSAR_SWS_DefaultErrorTracer" |
|---|----|----------------------------------|

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.69 Specification Item SWS_Eth_00237

Trace References:

none

Content:

If **default development** error detection is enabled: the function shall check the parameter RxStats for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately re-named "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx

module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"

- In chapter "7 References":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.70 Specification Item SWS_Eth_00248

Trace References:

SRS_BSW_00101, SRS_BSW_00416

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZED UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"
- "[SWS_XXX_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module

shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.71 Specification Item SWS_Eth_00249

Trace References:

SRS_BSW_00323, SRS_BSW_00369

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the

correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

– SWS_RTE –

- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611] "If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":

Rename "Development Error Tracer" to "Default Error Tracer"

Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.72 Specification Item SWS_Eth_00250

Trace References:

SRS_BSW_00323, SRS_BSW_00369

Content:

If **default development** error detection is enabled: the function shall check the parameter TxStats for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_XXX_YYYYY]"

In case default error detection is enabled for the XXXX module: The XXXX module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_XXX_YYYYY]"

If default error detection is enabled: the function shall check that the service XXX_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyy]"

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"
- "[SWS_xxx_yyyyy]"

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"
- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1

1.73 Specification Item SWS_Eth_00252

Trace References:

SRS_BSW_00101, SRS_BSW_00416

Content:

If **default development** error detection is enabled: the function shall check that the service Eth_Init was previously called. If the check fails, the function shall raise the **default development** error ETH_E_NOT_INITIALIZED_UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error

tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."

- "module raises the development error XXX_E_TRANSITION"

- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors

- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"

- Remove [SWS_Rte_07676]

- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."

- Change [SWS_Rte_06631]

[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

TOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.74 Specification Item SWS_Eth_00253

Trace References:

SRS_BSW_00323, SRS_BSW_00369

Content:

If **default development** error detection is enabled: the function shall check the parameter CtrlIdx for being valid. If the check fails, the function shall raise the **default development** error ETH_E_INV_CTRL_IDX otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "development error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the development error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "development error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyyy]

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."

- "module raises the Default error XXX_E_TRANSITION"

- "The DET provides services to store default errors"

...

The correct text would be:

- sub chapter is called "7.x Development errors"

- "[SWS_xxx_yyyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return

E_NOT_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

- SWS_RTE —
 - Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
 - In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
- | | | |
|---|----|----------------------------------|
| Rename "AUTOSAR_SWS_DevelopmentErrorTracer" | to | "AUTOSAR_SWS_DefaultErrorTracer" |
|---|----|----------------------------------|

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
- Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.75 Specification Item SWS_Eth_00254

Trace References:

SRS_BSW_00323, SRS_BSW_00369

Content:

If **default development** error detection is enabled: the function shall check the parameter TxStats for being valid. If the check fails, the function shall raise the **default development** error ETH_E_PARAM_POINTER otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

Problem description:

The DET was renamed from development error tracer to default error tracer.

This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

"default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:

"Switches the Default Error Tracer (Det) detection and notification..."

The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

Agreed solution:

Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

Blueprint/Example:

- sub chapter is now called "7.x Default errors"

- "[SWS_xxx_yyyyy]"

In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"

- "[SWS_xxx_yyyyy]"

If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."
- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
- ...

The correct text would be:

- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyy]

In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"

- "[SWS_xxx_yyyyy]

If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:

— SWS_RTE —

- Change 4.8 Default errors to 4.8 Development errors
 - Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
 - Remove [SWS_Rte_07676]
 - Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
 - Change [SWS_Rte_06631]
- [SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:

- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:

- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:

- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:

- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:

- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:

- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:

- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
-Last change on issue 73570 comment 47-

BW-C-Level:

Application	Specification	Bus
1	1	1