Document Title	Specification of Ethernet Switch Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	656

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R23-11

Document Change History

Date	Release	Changed by	Description
2023-11-23	R23-11	AUTOSAR Release Management	• Concept 710 (Deterministic communication with TSN) incorporated • Interaction with the Firewall module added
2022-11-24	R22-11	AUTOSAR Release Management	• Remodel EthSwtPort MAC address and VLAN definition • Implemented MACSec • Implemented Derterministic Communication with TSN
2021-11-25	R21-11	AUTOSAR Release Management	• Added 10BASE-T1S support • Clarified return values and development errors • Removed EthSwtMgmtInfoIndicationTimeout related requirements

▽

△

2020-11-30	R20-11	AUTOSAR Release Management	• Support for Ethernet wake on data line • Modified description of return values in EthSwt_StoreConfiguration, EthSwt_ResetConfiguration • Updated the types for ReTaggingVlanId and DoubleTaggingVlanId • Fix service IDs for EthSwt_DeletePortMirrorConfiguration, EthSwt_PortLinkStateRequest, EthSwt_GetMaxFIFOBufferFillLevel • Editorial changes
2019-11-28	R19-11	AUTOSAR Release Management	• Possibility to explicitly request or release Ethernet link state added • Replace usage of EthTrcv_ModeType with the Eth_ModeType • Support for 2500 MBit/s Ethernet connection • Fix Ethernet Hardware Initialization • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Clarified Port Mirroring concepts. • Introduced timeout for ARL table entries • Added counter synchronization for cascaded switches
2017-12-08	4.3.1	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation

▽

△

2016-11-30	4.3.0	AUTOSAR Release Management	• Restructured VLAN-membership as a port-related configuration parameter • Introduced configuration of rate policers on ingress side • Introduced filter configuration for double tagged frames • Introduced configuration of minimum buffer size for FIFOS • Introduced Types to read HW- statistic by List pointer; reorganized interfaces to read HW-statistics. • Introduced Compensation of Ethernet switch delays for Global Time Synchronization • Add / update elements to describe MAC interface and physical interface • Added testing functionality for diagnostic use cases • Added Possibility to switch off ports and switch instances according to VLAN or PNC. • Introduced interfaces for verification of switch configuration
2015-07-31	4.2.2	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	• Initial Release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

1 Introduction and functional overview

In the AUTOSAR Layered Software Architecture [1], the Ethernet Switch Driver belongs to the Communication Hardware Abstraction.

This indicates the main task of the Ethernet Switch Driver:

Provide to the upper layers (e.g. Ethernet Interface [2]) a hardware independent interface comprising a switch with several ports. This interface shall be uniform for all Ethernet switches. Thus, the upper layers may access the underlying communication technology in a uniform manner.

A single Ethernet Switch Driver module supports only one type of switch hardware. The Ethernet physical layer ports are configured by the Ethernet Transceiver Driver[3]. The Ethernet Switch Driver's prefix generates a unique namespace. The Ethernet Interface can access different Ethernet controller types using different Ethernet Switch Drivers using this prefix. The decision which driver to use to access a particular transceiver is a configuration parameter of the Ethernet Interface.

Figure 1.1 depicts the lower part of the Ethernet stack. Accesses via an SPI- and MII/MDIO-Hardware-Interface for switch specific configuration or functions are directly done via the Ethernet Driver [4] or the SPI driver [5].

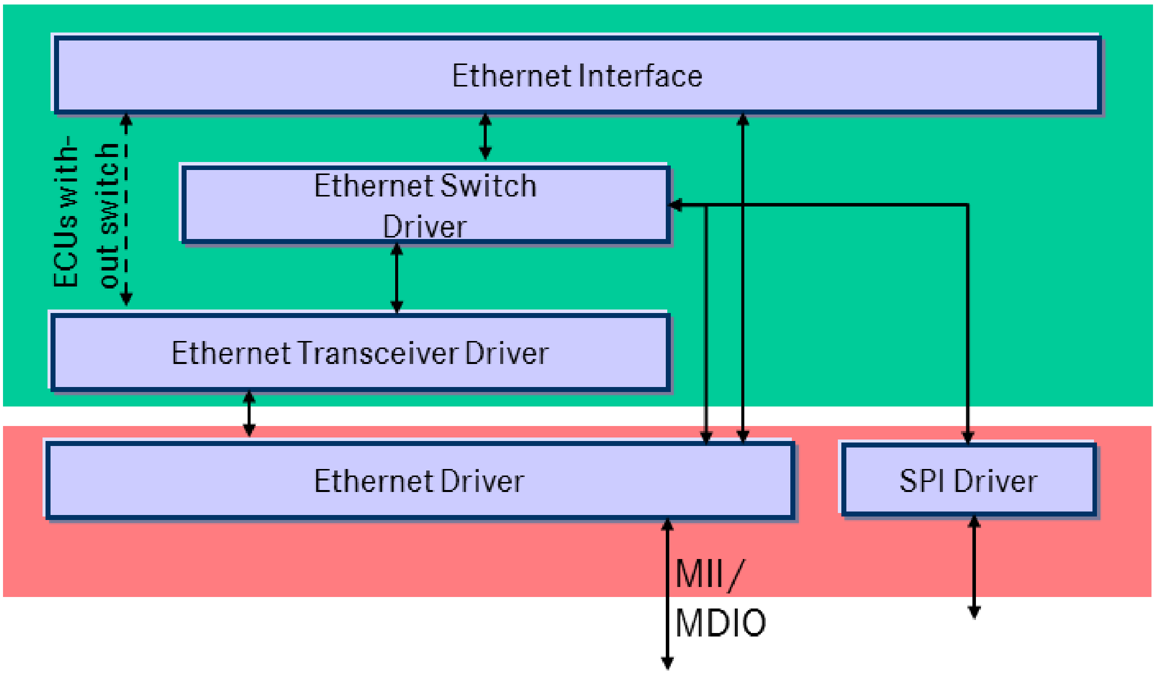


Figure 1.1: Ethernet Switch Driver in layer architecture

2 Acronyms, abbreviations and definitons

The glossary below includes acronyms, abbreviations and definitions relevant to the Ethernet Switch Driver module that are not included in the [6, AUTOSAR glossary].

2.1 Acronyms and abbreviations

Acronym / Abbreviation:	Description:
DEM	Diagnostic Event Manager module
EcuM	ECU State Manager module
Eth	Ethernet Controller Driver (AUTOSAR BSW module)
EthIf	Ethernet Interface (AUTOSAR BSW module)
EthTrcv	Ethernet Transceiver Driver (AUTOSAR BSW module)
MII	Media Independent Interface (standardized interface provided by Ethernet controllers to access Ethernet transceivers)
MDIO	Management Data Input/Output
OA TC10	Open Alliance TC10 specification (see [7])

2.2 Definitions

2.2.1 Ethernet packet

Definition: An "Ethernet packet" is an on-wire format defined by [8, IEEE Std 802.3-2022] which includes the following parts: Preamble (7 bytes), SFD (start frame delimiter, 1 byte), Ethernet frame (up to 2000 bytes))

2.2.2 Ethernet frame

Definition: An "Ethernet frame" is on-wire format defined [8, IEEE Std 802.3-2022] which includes the following parts: MAC destination address field (6 bytes), MAC source address field (6 bytes), Type field (2 bytes), MAC client data field (inlcude optional Q-Tag (4 bytes)) (up to 1982 bytes), optional PAD (padding bytes), FCS (frame check sequence, 4 bytes)

2.2.3 Stream

Definition: A "stream" represent multiple Ethernet frames which are grouped by similar frame attributes (e.g. MAC source address)

2.2.4 Stream identification

Definition: The term "Stream identification" is derived from [9, IEEE Std 802.1Q-2022] and represent the functionality to identify received Ethernet frames based on a particular set of frame attributes. Frames carrying different sets of frame attributes can only be identified with a single stream

3 Related documentation

3.1 Input documents & related standards and norms

[1] Layered Software Architecture
AUTOSAR_CP_EXP_LayeredSoftwareArchitecture

[2] Specification of Ethernet Interface
AUTOSAR_CP_SWS_EthernetInterface

[3] Specification of Ethernet Transceiver Driver
AUTOSAR_CP_SWS_EthernetTransceiverDriver

[4] Specification of Ethernet Driver
AUTOSAR_CP_SWS_EthernetDriver

[5] Specification of SPI Handler/Driver
AUTOSAR_CP_SWS_SPIHandlerDriver

[6] Glossary
AUTOSAR_FO_TR_Glossary

[7] OPEN Sleep/Wake-up Specification for Automotive Ethernet
http://www.opensig.org/Automotive-Ethernet-Specifications/

[8] IEEE 802.3-2022
https://www.ieee802.org/3/

[9] IEEE 802.1Q-2022 - IEEE Standard for Local and Metropolitan Area Network - Bridges and Bridged Networks
https://ieeexplore.ieee.org/

[10] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[11] Requirements on Ethernet Support in AUTOSAR
AUTOSAR_CP_SRS_Ethernet

[12] General Requirements on Basic Software Modules
AUTOSAR_CP_SRS_BSWGeneral

[13] IEEE 802.1Q-2018 - IEEE Standard for Local and Metropolitan Area Network - Bridges and Bridged Networks
https://ieeexplore.ieee.org/

[14] IEEE 802.1CB-2017 - IEEE Standard for Local and Metropolitan Area Network - Frame Replication and Elimination for Reliability
https://ieeexplore.ieee.org/

[15] Specification of Time Synchronization over Ethernet
AUTOSAR_CP_SWS_TimeSyncOverEthernet

[16] Specification of NVRAM Manager
AUTOSAR_CP_SWS_NVRAMManager

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software [10, SWS_BSWGeneral] which is also valid for Ethernet Switch Driver.

Thus, the specifications [SWS_BSWGeneral] [10], SRS_Ethernet [11] shall be considered as additional and required specification for Ethernet Switch Driver.

4 Constraints and assumptions

4.1 Constraints

The following constraints have to be considered:

- The Ethernet switch driver module is only able to handle a single thread of execution. The execution must not be pre-empted by itself.
- The implementation is limited to 10Mbit/s, 100MBit/s and 1000Mbit/s Ethernet bandwidth and to PHYs connected via (gigabit) Media Independent Interface (xMII).
- External MACPHY connected with an Ethernet switch over SPI are not supported.
- The Ethernet switch driver do only support VLAN-aware Ethernet switches
- The Ethernet switch driver support only passive stream identification. Active stream identification is not supported and therefore features like frame replication are not supported
- Stream identification is considered in the out-facing. Thus, stream identification is applied only at ingress side
- The Ethernet switch driver support only a subset of the IEEE specified forwarding process in an Ethernet switch. Table 4.1 give an overview which IEEE specfied forwarding processing steps are covered by AUTOSAR:

IEEE specified bridge execution order	Coverage by AUTOSAR
Default priority assignment ([AC]:13.1)	n/a
Placeholder MacSEC (incl. corresponding filter mechanism)	n/a
support of the EISS ([Q]:6.9.1)	n/a
Frame Type Acceptance filter ([Q]:6.9 f))	covered (see paragraph 7.1.5.2.1)
Ingress VID translation ([Q]:6.9 f))	n/a
Port-based VLAN Classification ([Q]:6.9 d)) XOR Port-and-Protocol-based VLAN classification ([Q]:6.12)	"Port-based VLAN Classification" covered (see paragraph 7.1.5.2.1)
Priority Code Point Decoding ([Q]:6.9.3)	"derivation of priority" covered (see paragraph 7.1.5.2.1)
Priority Regeneration ([Q]:6.9.4)	covered (see paragraph 7.1.5.2.2)
Outfacing Input Stream Identification Function(s) ([CB]:9.1.1.5)	covered (see paragraph 7.1.5.2.1)
Placeholder FRER functionality	n/a

▽

△

Infacing Output Stream Identification Function(s) ([CB]:9.1.1.2)	n/a
Active topology enforcement ([Q]:8.6.1)	n/a
Ingress filtering ([Q]:8.6.2)	covered (see paragraph 7.1.5.2.4)
Frame filtering ([Q]:8.6.3)	covered (see paragraph 7.1.5.2.5)
Egress filtering ([Q]:8.6.4)	covered (see paragraph 7.1.5.2.6)
Stream filtering ([Q]:8.6.5.3) (selection process)	covered (see paragraph 7.1.5.2.7)
Maximum SDU Size Filtering ([Q]:8.6.5.3.1)	covered (see paragraph 7.1.5.2.7)
Stream Gating ([Q]:8.6.5.4)	covered (see paragraph 7.1.5.2.7)
Flow metering ([Q]:8.6.5.5)	covered (see paragraph 7.1.5.2.7)
ATS Eligibility Time Assignment 9([Q]:8.6.5.6)	covered (see paragraph 7.1.5.2.7)
Infacing Input Stream Identification Function(s) ([CB]:9.1.1.4)	n/a
Placeholder FRER functionality	n/a
Outfacing Output Stream Identification Function(s) ([CB]:9.1.1.3)	n/a (note: only passive stream identification is supported)
Queuing frames ([Q]:8.6.6)	covered (see paragraph 7.1.5.2.9)
Shapers and Transmission selection and queuing management	covered (see paragraph 7.1.5.2.9)
Priority Code Point Encoding ([Q]:6.9.3)	covered (see paragraph 7.1.5.2.10)
Egress VID translation ([Q]:6.9 g))	n/a
support of the EISS ([Q]:6.9.2)	covered "VLAN forwarding tagged or untagged" (see paragraph 7.1.5.2.10)
Placeholder MacSEC (incl. corresponding classification mechanism)	n/a

Table 4.1: AUTOSAR coverage of IEEE specified bridge execution order

4.2 Assumptions

The following assumptions have to be considered

- Depending on the Ethernet hardware, it may become necessary that implementations deviate from API specifications in respect to the asynchronous/synchronous behavior.

4.3 Applicability to car domains

The Ethernet BSW stack is intended to be used wherever high data rates are required but no hard real-time is required. Of course, it can also be used for less-demanding use cases, i.e. for low data rates.

5 Dependencies to other modules

This chapter lists the modules interacting with the Ethernet Switch Driver module.

Modules that use the Ethernet Switch Driver module:

- Ethernet Interface (EthIf) calls the Ethernet Switch driver for initializing and accessing the switch device.

Modules used by the Ethernet Switch Driver module:

- Ethernet Controller Driver (Eth) for transceiver access via Media Independent Interface (MII).
- Ethernet Transceiver Driver (EthTrcv) for configuring the PHY ports and controlling/checking the ports.
- The configuration of the Ethernet Switch device can be either via MDIO or SPI. In case of an SPI interface access to SPI module is necessary.

Dependencies to other Modules:

- On certain systems the Ethernet switch might share resources with other components, and may depend on their configuration. If those resources are within the scope of other modules (e.g. PLL configuration, memory mapping, etc.) the Ethernet Switch Driver module does not take care of configuring those components but requires their preceding initialization.

6 Requirements Tracing

The following tables reference the requirements specified in [11] as well as [12] and links to the fulfillment of these. Please note that if column "Satisfied by" is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[FO_RS_Fw_00011]	Hardware-Accelerated Filtering Support	[SWS_EthSwt_91041] [SWS_EthSwt_91042] [SWS_EthSwt_91043]
[SRS_BSW_00003]	All software modules shall provide version and identification information	[SWS_EthSwt_00131]
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_EthSwt_00006] [SWS_EthSwt_00007] [SWS_EthSwt_00008] [SWS_EthSwt_00011]
[SRS_BSW_00161]	The AUTOSAR Basic Software shall provide a microcontroller abstraction layer which provides a standardized interface to higher software layers	[SWS_EthSwt_00099] [SWS_EthSwt_00130]
[SRS_BSW_00162]	The AUTOSAR Basic Software shall provide a hardware abstraction layer	[SWS_EthSwt_00099] [SWS_EthSwt_00130]
[SRS_BSW_00171]	Optional functionality of a Basic-SW component that is not required in the ECU shall be configurable at pre-compile-time	[SWS_EthSwt_00022] [SWS_EthSwt_00029] [SWS_EthSwt_00035] [SWS_EthSwt_00042] [SWS_EthSwt_00049] [SWS_EthSwt_00056] [SWS_EthSwt_00058] [SWS_EthSwt_00090] [SWS_EthSwt_00095] [SWS_EthSwt_00124] [SWS_EthSwt_00129] [SWS_EthSwt_00177] [SWS_EthSwt_00186] [SWS_EthSwt_00191] [SWS_EthSwt_00202] [SWS_EthSwt_00210] [SWS_EthSwt_00215] [SWS_EthSwt_00220] [SWS_EthSwt_00225] [SWS_EthSwt_00229] [SWS_EthSwt_00230] [SWS_EthSwt_00240] [SWS_EthSwt_00243] [SWS_EthSwt_00249] [SWS_EthSwt_00253] [SWS_EthSwt_00257] [SWS_EthSwt_00261] [SWS_EthSwt_00264] [SWS_EthSwt_00268] [SWS_EthSwt_00273] [SWS_EthSwt_00287] [SWS_EthSwt_00291] [SWS_EthSwt_00297] [SWS_EthSwt_00303] [SWS_EthSwt_00308] [SWS_EthSwt_00312] [SWS_EthSwt_00317] [SWS_EthSwt_00322] [SWS_EthSwt_00327] [SWS_EthSwt_00332] [SWS_EthSwt_00338] [SWS_EthSwt_00344] [SWS_EthSwt_00350] [SWS_EthSwt_00362] [SWS_EthSwt_00370] [SWS_EthSwt_00379] [SWS_EthSwt_00403] [SWS_EthSwt_00405] [SWS_EthSwt_00427] [SWS_EthSwt_00432] [SWS_EthSwt_00441] [SWS_EthSwt_00443]
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_EthSwt_00009] [SWS_EthSwt_00154] [SWS_EthSwt_00156] [SWS_EthSwt_00157] [SWS_EthSwt_00180]
[SRS_BSW_00347]	A Naming seperation of different instances of BSW drivers shall be in place	[SWS_EthSwt_00131]
[SRS_BSW_00350]	All AUTOSAR Basic Software Modules shall allow the enabling/ disabling of detection and reporting of development errors.	[SWS_EthSwt_00386] [SWS_EthSwt_00387] [SWS_EthSwt_00389] [SWS_EthSwt_00390] [SWS_EthSwt_00391] [SWS_EthSwt_00392] [SWS_EthSwt_00393]

▽

△

Requirement	Description	Satisfied by
[SRS_BSW_00369]	All AUTOSAR Basic Software Modules shall not return specific development error codes via the API	[SWS_EthSwt_00009] [SWS_EthSwt_00128] [SWS_EthSwt_00154] [SWS_EthSwt_00156] [SWS_EthSwt_00157] [SWS_EthSwt_00164] [SWS_EthSwt_00180]
[SRS_BSW_00375]	Basic Software Modules shall report wake-up reasons	[SWS_EthSwt_00098]
[SRS_BSW_00385]	List possible error notifications	[SWS_EthSwt_00001] [SWS_EthSwt_00113] [SWS_EthSwt_00395]
[SRS_BSW_00386]	The BSW shall specify the configuration and conditions for detecting an error	[SWS_EthSwt_00016] [SWS_EthSwt_00164]
[SRS_BSW_00395]	The Basic Software Module specifications shall list all configuration parameter dependencies	[SWS_EthSwt_00165]
[SRS_BSW_00406]	A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called	[SWS_EthSwt_00123]
[SRS_BSW_00413]	An index-based accessing of the instances of BSW modules shall be done	[SWS_EthSwt_00120] [SWS_EthSwt_00154] [SWS_EthSwt_00156] [SWS_EthSwt_00157] [SWS_EthSwt_00180]
[SRS_BSW_00433]	Main processing functions are only allowed to be called from task bodies provided by the BSW Scheduler	[SWS_EthSwt_00114] [SWS_EthSwt_00115]
[SRS_Eth_00087]	Semi-Static Auto-Configuration	[SWS_EthSwt_00031] [SWS_EthSwt_00032] [SWS_EthSwt_00060] [SWS_EthSwt_00061] [SWS_EthSwt_00086] [SWS_EthSwt_00087] [SWS_EthSwt_00091] [SWS_EthSwt_00092] [SWS_EthSwt_00098] [SWS_EthSwt_00111] [SWS_EthSwt_00117] [SWS_EthSwt_00118] [SWS_EthSwt_00125] [SWS_EthSwt_00126] [SWS_EthSwt_00127] [SWS_EthSwt_00182] [SWS_EthSwt_00183] [SWS_EthSwt_00187] [SWS_EthSwt_00188] [SWS_EthSwt_00193] [SWS_EthSwt_00194] [SWS_EthSwt_00196] [SWS_EthSwt_00197] [SWS_EthSwt_00203] [SWS_EthSwt_00204] [SWS_EthSwt_00226] [SWS_EthSwt_00227] [SWS_EthSwt_00228] [SWS_EthSwt_00235]
[SRS_Eth_00107]	The Ethernet Transceiver Driver shall support access to the wake up reason.	[SWS_EthSwt_00442] [SWS_EthSwt_91040]
[SRS_Eth_00114]	Ethernet Switch Filtering and Policing	[SWS_EthSwt_00134] [SWS_EthSwt_00172] [SWS_EthSwt_00173] [SWS_EthSwt_00233] [SWS_EthSwt_00491] [SWS_EthSwt_00492] [SWS_EthSwt_00493] [SWS_EthSwt_00494] [SWS_EthSwt_00601] [SWS_EthSwt_00602] [SWS_EthSwt_00604] [SWS_EthSwt_00605] [SWS_EthSwt_00606] [SWS_EthSwt_00607] [SWS_EthSwt_00608] [SWS_EthSwt_00609] [SWS_EthSwt_CONSTR_00489] [SWS_EthSwt_CONSTR_00602] [SWS_EthSwt_CONSTR_00603]

▽

△

Requirement	Description	Satisfied by
[SRS_Eth_00118]	Transparent interface to underlying EthTrcv module(s)	[SWS_EthSwt_00018] [SWS_EthSwt_00019] [SWS_EthSwt_00023] [SWS_EthSwt_00025] [SWS_EthSwt_00026] [SWS_EthSwt_00038] [SWS_EthSwt_00044] [SWS_EthSwt_00045] [SWS_EthSwt_00051] [SWS_EthSwt_00052] [SWS_EthSwt_00098] [SWS_EthSwt_00154] [SWS_EthSwt_00156] [SWS_EthSwt_00157] [SWS_EthSwt_00164] [SWS_EthSwt_00217] [SWS_EthSwt_00222] [SWS_EthSwt_00398] [SWS_EthSwt_00440] [SWS_EthSwt_91003]
[SRS_Eth_00119]	Access to hardware status of ports	[SWS_EthSwt_00037] [SWS_EthSwt_00038] [SWS_EthSwt_00098] [SWS_EthSwt_00117] [SWS_EthSwt_00118] [SWS_EthSwt_00154] [SWS_EthSwt_00203] [SWS_EthSwt_00204] [SWS_EthSwt_00430] [SWS_EthSwt_00431]
[SRS_Eth_00120]	Hardware access via MII and/or SPI	[SWS_EthSwt_00098] [SWS_EthSwt_00206] [SWS_EthSwt_00207] [SWS_EthSwt_00211] [SWS_EthSwt_00212] [SWS_EthSwt_00216] [SWS_EthSwt_00217] [SWS_EthSwt_00221] [SWS_EthSwt_00222]
[SRS_Eth_00121]	Configuration of forwarding rules	[SWS_EthSwt_00132] [SWS_EthSwt_00133] [SWS_EthSwt_00134] [SWS_EthSwt_00135] [SWS_EthSwt_00172] [SWS_EthSwt_00173] [SWS_EthSwt_00178] [SWS_EthSwt_00179] [SWS_EthSwt_00234] [SWS_EthSwt_00455] [SWS_EthSwt_00460] [SWS_EthSwt_00461] [SWS_EthSwt_00462] [SWS_EthSwt_00463] [SWS_EthSwt_00611] [SWS_EthSwt_00612] [SWS_EthSwt_00613] [SWS_EthSwt_CONSTR_00452] [SWS_EthSwt_CONSTR_00453] [SWS_EthSwt_CONSTR_00454] [SWS_EthSwt_CONSTR_00457] [SWS_EthSwt_CONSTR_00495]
[SRS_Eth_00122]	Persistent storage of configurations	[SWS_EthSwt_00086] [SWS_EthSwt_00087] [SWS_EthSwt_00091] [SWS_EthSwt_00092] [SWS_EthSwt_00098] [SWS_EthSwt_00125] [SWS_EthSwt_00126] [SWS_EthSwt_00127] [SWS_EthSwt_00182] [SWS_EthSwt_00183] [SWS_EthSwt_00192] [SWS_EthSwt_00193] [SWS_EthSwt_00194] [SWS_EthSwt_00196]
[SRS_Eth_00123]	Testing and diagnostic of switch ports	[SWS_EthSwt_00293] [SWS_EthSwt_00299] [SWS_EthSwt_00305] [SWS_EthSwt_00309] [SWS_EthSwt_00313] [SWS_EthSwt_00318] [SWS_EthSwt_00323] [SWS_EthSwt_00328] [SWS_EthSwt_00334] [SWS_EthSwt_00340] [SWS_EthSwt_00346] [SWS_EthSwt_00416] [SWS_EthSwt_00417] [SWS_EthSwt_00418] [SWS_EthSwt_00419] [SWS_EthSwt_00420] [SWS_EthSwt_00421] [SWS_EthSwt_00422] [SWS_EthSwt_00424] [SWS_EthSwt_00425] [SWS_EthSwt_00426] [SWS_EthSwt_91014] [SWS_EthSwt_91015] [SWS_EthSwt_91016] [SWS_EthSwt_91017] [SWS_EthSwt_91018] [SWS_EthSwt_91019] [SWS_EthSwt_91020] [SWS_EthSwt_91021] [SWS_EthSwt_91022] [SWS_EthSwt_91023] [SWS_EthSwt_91024] [SWS_EthSwt_91025] [SWS_EthSwt_91029] [SWS_EthSwt_91030] [SWS_EthSwt_91031] [SWS_EthSwt_91032]

▽

△

Requirement	Description	Satisfied by
[SRS_Eth_00125]	The Ethernet Switch Driver shall support switch frame management	[SWS_EthSwt_00098] [SWS_EthSwt_00240] [SWS_EthSwt_00241] [SWS_EthSwt_00242] [SWS_EthSwt_00243] [SWS_EthSwt_00245] [SWS_EthSwt_00378] [SWS_EthSwt_91002] [SWS_EthSwt_91004] [SWS_EthSwt_91005] [SWS_EthSwt_91006] [SWS_EthSwt_91007] [SWS_EthSwt_91008] [SWS_EthSwt_91009] [SWS_EthSwt_91010] [SWS_EthSwt_91028]
[SRS_Eth_00126]	Independent reset of host ECU and switch hardware	[SWS_EthSwt_00292] [SWS_EthSwt_91012] [SWS_EthSwt_91013]
[SRS_Eth_00128]	The Ethernet Switch Driver shall provide statistic counter values per port	[SWS_EthSwt_00106] [SWS_EthSwt_00198] [SWS_EthSwt_00199] [SWS_EthSwt_00231] [SWS_EthSwt_00372] [SWS_EthSwt_00373] [SWS_EthSwt_91000] [SWS_EthSwt_91001]
[SRS_Eth_00178]	Ethernet Switch Stream Identification	[SWS_EthSwt_00465] [SWS_EthSwt_00467] [SWS_EthSwt_00469] [SWS_EthSwt_00471] [SWS_EthSwt_00472] [SWS_EthSwt_00475] [SWS_EthSwt_00476] [SWS_EthSwt_00477] [SWS_EthSwt_00478] [SWS_EthSwt_00479] [SWS_EthSwt_00480] [SWS_EthSwt_00481] [SWS_EthSwt_00482] [SWS_EthSwt_00483] [SWS_EthSwt_00484] [SWS_EthSwt_00486] [SWS_EthSwt_00487] [SWS_EthSwt_00610] [SWS_EthSwt_CONSTR_00464] [SWS_EthSwt_CONSTR_00468] [SWS_EthSwt_CONSTR_00470] [SWS_EthSwt_CONSTR_00485]
[SRS_Eth_00179]	Ethernet Switch Transmission Selection Algorithm	[SWS_EthSwt_00613]
[SRS_Eth_00180]	Ethernet Switch port scheduling of egress queues	[SWS_EthSwt_00613] [SWS_EthSwt_00614]

Table 6.1: RequirementsTracing

7 Functional specification

7.1 Ethernet BSW stack

As part of the AUTOSAR Layered Software Architecture according to Figure 7.1, the Ethernet BSW modules also form a layered software stack.

Figure 7.1 depicts the basic Ethernet BSW stack. The EthIf module accesses several switches using one or more Ethernet Switch Driver modules. The role of the Ethernet transceiver driver is to configure and control the physical layer ports (PHY) integrated into or connected to a switch. Whereas, the role of the Ethernet switch driver is the configuration and control of the switch. In case the Ethernet interface wants to access a PHY, it has to use the APIs of the switch driver which forward the API call to the addressed transceiver driver.

By separating the transceiver driver from the switch driver, different hardware architectures will be supported. In HW-Variant 1, the PHYs are separate devices from different vendors. They are connected via MII and MDIO to a switch which is integrated into a microcontroller. In HW-Variant 2, the switch has integrated PHYs. In HW-Variant 3, the microcontroller can control the switch via MDIO or SPI and the switch has three external PHYs which can be controlled via MDIO. In this case, different Ethernet transceiver drivers might occur.

Please note that the functional behavior of the ingress and egress port of a switch is implemented in hardware in the switch devices (see [13]). Thus, the configuration from chapter 10 in some parts has to be written to the switch device.

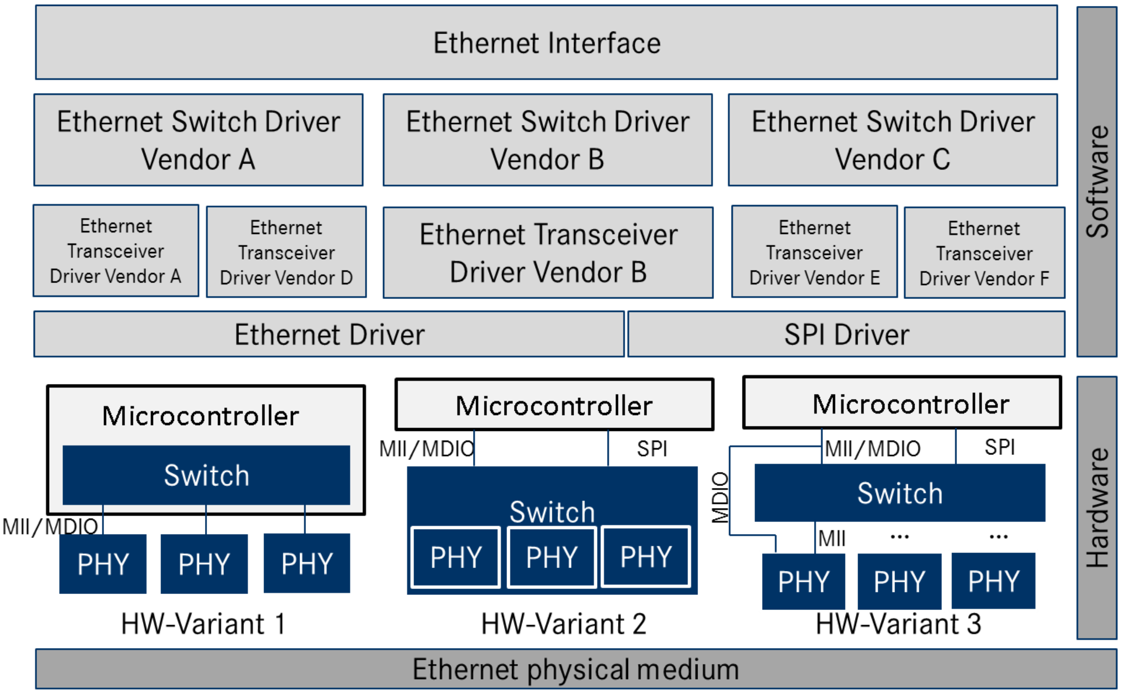

Ethernet Interface
Ethernet Switch Driver Vendor A
Ethernet Switch Driver Vendor B
Ethernet Switch Driver Vendor C
Ethernet Transceiver Driver Vendor A
Ethernet Transceiver Driver Vendor D
Ethernet Transceiver Driver Vendor B
Ethernet Transceiver Driver Vendor E
Ethernet Transceiver Driver Vendor F
Ethernet Driver
SPI Driver
Software
Microcontroller
Switch
MII/MDIO
PHY
PHY
PHY
HW-Variant 1
Microcontroller
MII/MDIO
SPI
Switch
PHY
PHY
PHY
HW-Variant 2
Microcontroller
MDIO
MII/MDIO
SPI
Switch
MII
...
...
PHY
PHY
PHY
HW-Variant 3
Hardware
Ethernet physical medium

Figure 7.1: Basic Structure of the Ethernet BSW stack.(Note: The different hardware variants are alternative setups)

7.1.1 Indexing scheme

Users of the Ethernet Switch Driver identify switch resources using an indexing scheme as depicted in Figure 7.2.

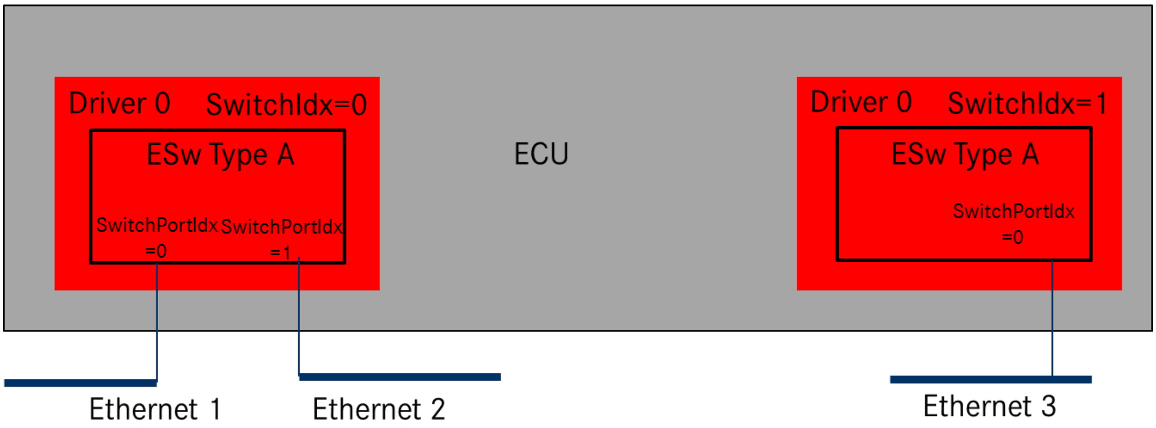


Figure 7.2: Ethernet Switch Driver indexing scheme

[SWS_EthSwt_00099] ⌈The Ethernet Switch Driver shall use a zero-based index to abstract the access for upper software layers.⌋*(SRS_BSW_00161, SRS_BSW_00162)*

[SWS_EthSwt_00130] ⌈The `SwitchPortIdx` is an index for a port at the switch.⌋ *(SRS_BSW_00161, SRS_BSW_00162)*

[SWS_EthSwt_00120] ⌈The parameter `EthSwtIdx` within the configuration shall correspond to the argument used in the API.⌋*(SRS_BSW_00413)*

[SWS_EthSwt_00180] ⌈The parameter `EthSwtIndex` shall be used to distinguish different instances of a switch driver module in case the API `Det_ReportError (uint16 ModuleId, uint8 InstanceId, uint8 ApiId, uint8 ErrorId)` is called.⌋*(SRS_BSW_00413, SRS_BSW_00323, SRS_BSW_00369)*

[SWS_EthSwt_00131] ⌈In case different Switch devices are used in one ECU, the function names of the different Ethernet Switch drivers must be modified such that no two functions with the same names are generated. It is the responsibility of the user to take care that no two functions with the same names are configured. The names may be extended with a vendor ID or a type ID.⌋*(SRS_BSW_00003, SRS_BSW_00347)*

7.1.2 Ethernet Switch Port Mirroring

Ethernet switch port mirroring use the common established functionality of the Ethernet switch hardware to mirror traffic of one or more Ethernet switch ports (mirrored port) to a another Ethernet switch port (capture port). The mirroring configuration is given by the port mirror configuration (see [SWS_EthSwt_91017]). The port mirror configuration is set up per Ethernet switch. The configuration is stored persistently by the Ethernet switch driver. Therefore a shadow buffer is used to store the port mirror configuration during runtime and stored persistently according to the NvM storing strategy (e.g. store the shadow buffer persistently upon ECU shutdown). The port mirror configuration could be activated and de-activated, respectively, explicitly via dedicated APIs. The port mirroring is controlled by a dedicated diagnostic CDD with receive diagnostic request and forward them to the Ethernet switch driver.

[SWS_EthSwt_00416] ⌈The port mirror configuration (see [SWS_EthSwt_91017]) shall be written to a shadow buffer of the Ethernet switch driver per Ethernet Switch by calling `EthSwt_WritePortMirrorConfiguration`.⌋*(SRS_Eth_00123)*

Note: One port mirror configuration is maintained per Ethernet switch.

[SWS_EthSwt_00417] ⌈The port mirror configuration shall be enabled and disabled, respectively, per Ethernet Switch by calling `EthSwt_SetPortMirrorState`. The current state of the stored port mirror configuration shall be stored persistently, to outlast an ECU reset and to restore the port mirroring activities after an ECU reset.⌋ *(SRS_Eth_00123)*

[SWS_EthSwt_00418] ⌈The stored port mirror configuration shall be marked as "to be deleted" by calling `EthSwt_DeletePortMirrorConfiguration`, if the port mirroring of the given Ethernet switch index is disabled (see [SWS_EthSwt_91022]. Otherwise the request to delete the port mirror configuration shall be rejected.⌋*(SRS_Eth_00123)*

Note: The shadow buffer is stored persistently according to the NvM storing strategy, e.g. store the shadow buffer persistently upon ECU shutdown.

[SWS_EthSwt_00419] ⌈The current port mirroring state shall be returned by calling `EthSwt_GetPortMirrorState`.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00420] ⌈The port mirror configuration per Ethernet switch shall be returned by calling `EthSwt_ReadPortMirrorConfiguration`.⌋*(SRS_Eth_00123)*

7.1.3 State Handling

[SWS_EthSwt_00435] ⌈All functions apart from `EthSwt_SetSwitchPortMode`, `EthSwt_GetSwitchPortMode`, `EthSwt_StartSwitchPortAutoNegotiation`, `EthSwt_GetLinkState`, `EthSwt_GetBaudRate`, `EthSwt_GetDuplexMode`, `EthSwt_ReadTrcvRegister`, `EthSwt_WriteTrcvRegister`, `EthSwt_Init`, `EthSwt_MainFunction` and `EthSwt_BackgroundTask` may only be called in state `ETHSWT_STATE_ACTIVE`.

If a function which can only run (succeed with `E_OK`) in the states `ETHSWT_STATE_PORTINIT_COMPLETED` and `ETHSWT_STATE_ACTIVE` is called before state `ETHSWT_STATE_PORTINIT_COMPLETED` is reached, the Ethernet switch driver shall raise the runtime error `ETHSWT_INIT_NOT_COMPLETED`.⌋*()*

[SWS_EthSwt_00436] ⌈`ETHSWT_STATE_PORTINIT_COMPLETED` shall be reached as soon as the port initialization has finished.⌋*()*

Note: ETHSWT_STATE_PORTINIT_COMPLETED can be reached either by the function EthSwt_Init or by a background task (see [SWS_EthSwt_91104]).

[SWS_EthSwt_00437] ⌈ETHSWT_STATE_ACTIVE shall be reached, when the Ethernet switch initialization has finished.⌋*()*

Note: The initialization of the Ethernet switch takes longer than the initialization of the Ethernet switch ports.

7.1.4 Handling of cable diagnostic

Cable diagnostic measurement is triggered by calling EthSwt_RunPortCableDiagnostic. The current state of the cable diagnostic measurement is polled by calling EthSwt_GetPortCableDiagnosticsResult. If EthSwt_GetPortCableDiagnosticsResult return with other value then ETHTRCV_CABLEDIAG_PENDING, then the cable diagnostic has finished.

Its up to the caller to re-trigger cable diagnostic again, if the measurement failed by returning ETHTRCV_CABLEDIAG_ERROR.

[SWS_EthSwt_00428] ⌈The cable diagnostic APIs (EthSwt_RunPortCableDiagnostic, EthSwt_GetPortCableDiagnosticsResult) shall only be called for Ethernet switch ports of a Ethernet switch, where the Ethernet switch ports reference an Ethernet transceiver.⌋*()*

Note: The upper layer is a CDD that triggers the cable diagnostic measurement and maintains the cable diagnostic result. The EthSwt forwards the API calls to the EthTrcv (see [SWS_EthSwt_00429] and [SWS_EthSwt_00346]).

7.1.5 Functional Description

7.1.5.1 Learning Phase at Start-up

[SWS_EthSwt_00226] ⌈The switch driver shall support a learning phase which can be divided into several sequential steps.⌋*(SRS_Eth_00087)*

Note: After assembly and initial power-up of the network, three learning phases follow which include MAC-Learning and IP-Address Assignment. Afterwards the learned parameters are stored to one or several non-volatile memories to make them available for subsequent start-ups. This process is shown in Figure 7.3. As an example for triggering this process, the DCM receives a diagnostic request via a bus system or a broadcast message in the Ethernet network. This diagnostic request can be forwarded to an SWC which triggers the auto-configuration process. However, the trigger is not part of this specification.

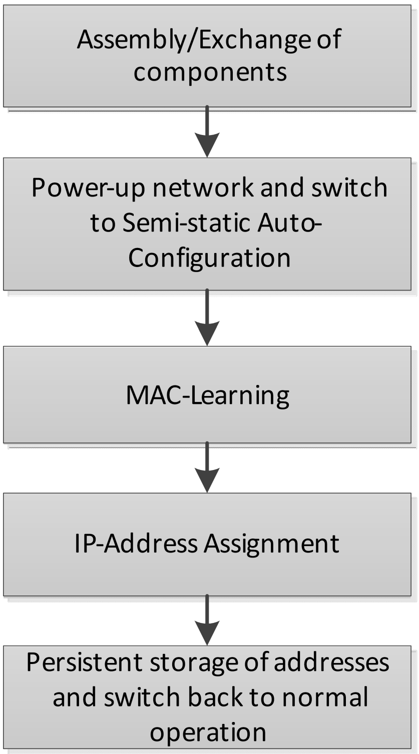
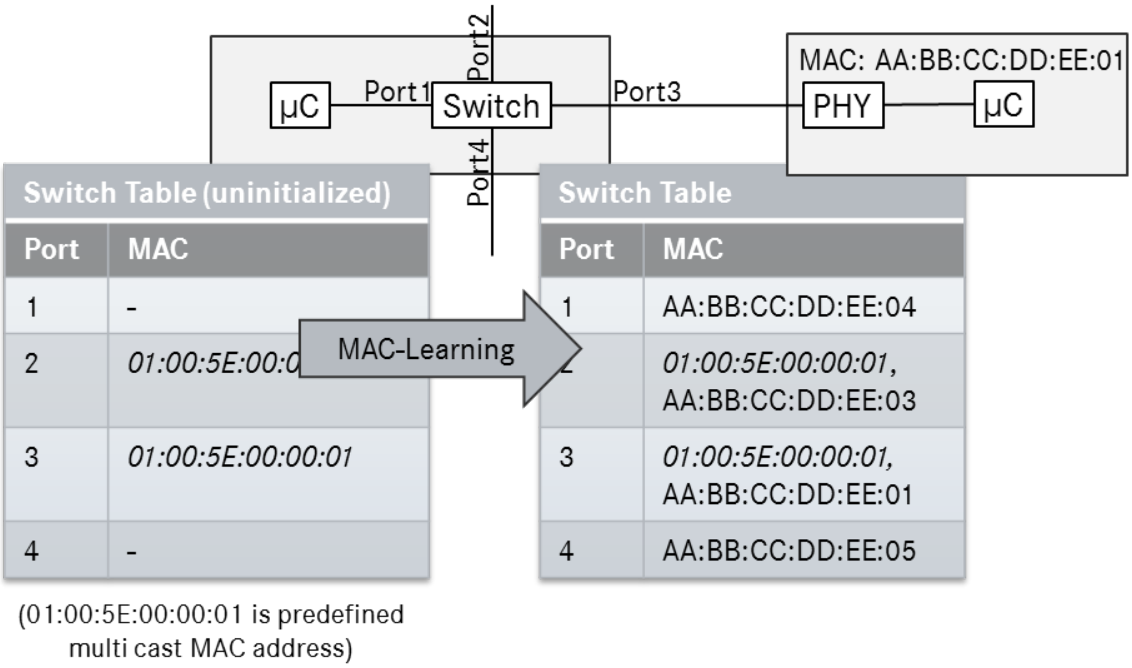

Assembly/Exchange of components
Power-up network and switch to Semi-static Auto-Configuration
MAC-Learning
IP-Address Assignment
Persistent storage of addresses and switch back to normal operation

Figure 7.3: Learning Process

MAC-Learning (Optional Step): In this phase, messages need to be sent through the network and the switch will learn new MAC addresses (cf. Figure 7.4). These MAC-addresses will be stored in addition to predefined addresses, e.g. multicast MAC addresses which are configured during the vehicle network design. If static learning is executed, i.e. MAC address will be persistently stored, it might be possible to add dynamically learned entries in the tables.

If software MAC learning is supported by switch hardware and the switch hardware expects an external microcontroller (see Variant 2 and 3 in Figure 7.1), packets with unknown MAC Source Address will be routed to this microcontroller. The MAC learning is done by integration code. It is intentionally not defined where this algorithm is located within the AUTOSAR stack as this might need a very time-optimized solution.



Switch Table (uninitialized)

Port	MAC
1	-
2	*01:00:5E:00:0*
3	*01:00:5E:00:00:01*
4	-

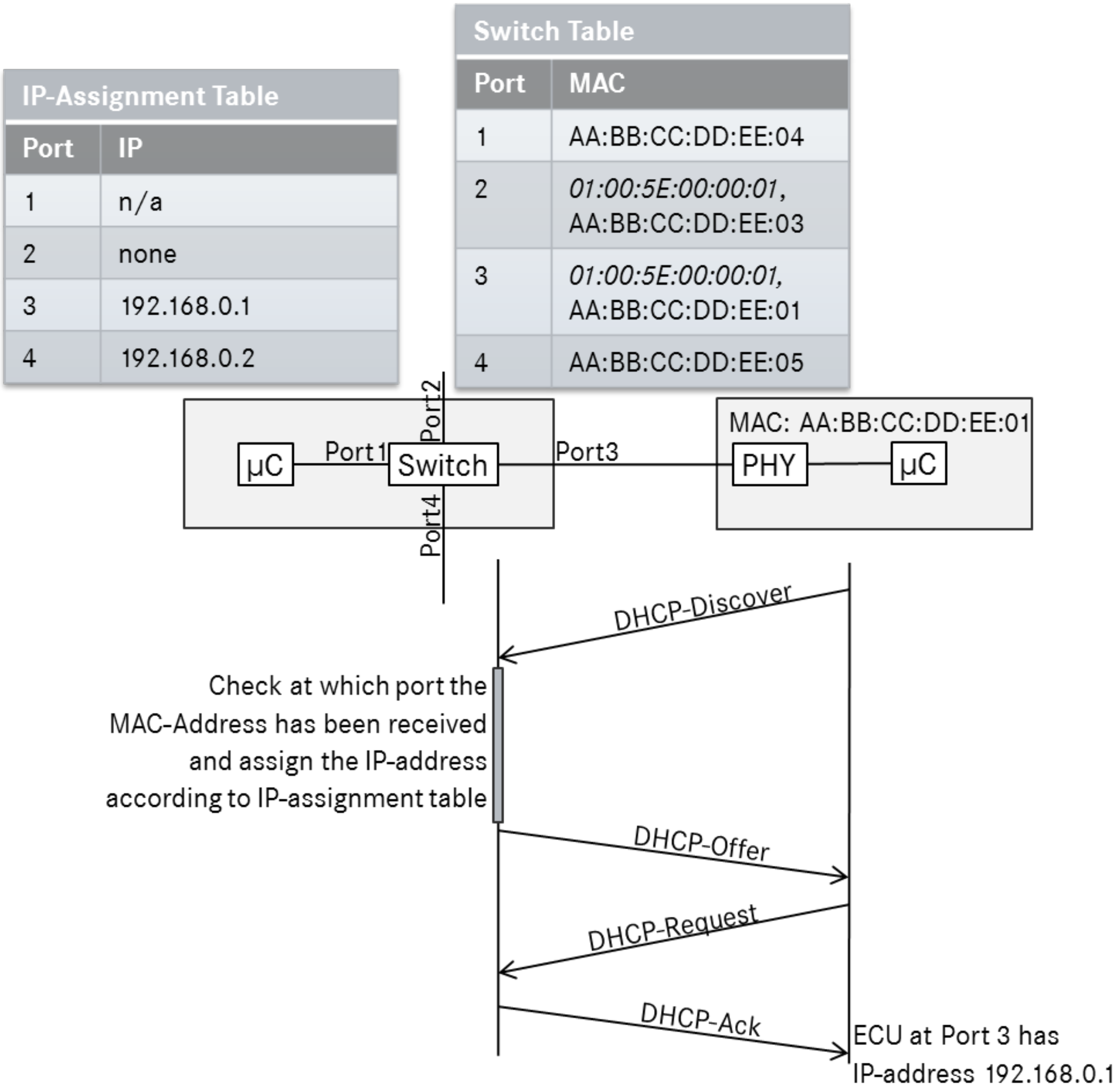
Switch Table

Port	MAC
1	AA:BB:CC:DD:EE:04
2	*01:00:5E:00:00:01,* AA:BB:CC:DD:EE:03
3	*01:00:5E:00:00:01,* AA:BB:CC:DD:EE:01
4	AA:BB:CC:DD:EE:05

(01:00:5E:00:00:01 is predefined
multi cast MAC address)

Figure 7.4: MAC-learning within the switch

IP-Address Assignment: In this phase, ECUs without a predefined IP-address will start to acquire an IP-address via DHCP (cf. Figure 7.5). Thus, these ECUs will run a DHCP-client while the ECU with the switch will run a DHCP server. In order to be able to assign always the same IP-address to a certain node, the DHCP server needs the information at which port the MAC address has been received. This port information can be interpreted as a "domain name" in the internet which is resolved to an IP address using a domain name server (DNS). With this port information the DHCP-server will assign the IP-address according to the IP-Assignment Table to the node. As mentioned above, this allows the assignment of MAC addresses by the Tier 1 and assignment of IP addresses by the OEM. With this mechanism it is also possible to assign different IP addresses to several VLANs at the same port. For this purpose, the IP-Assignment Table needs to be extended with a VLAN-column. Please note that the MAC-Learning-Phase can be combined with this phase.



IP-Assignment Table

Port	IP
1	n/a
2	none
3	192.168.0.1
4	192.168.0.2

Switch Table

Port	MAC
1	AA:BB:CC:DD:EE:04
2	*01:00:5E:00:00:01*, AA:BB:CC:DD:EE:03
3	*01:00:5E:00:00:01*, AA:BB:CC:DD:EE:01
4	AA:BB:CC:DD:EE:05

µC
Port1
Switch
Port2
Port3
Port4
MAC: AA:BB:CC:DD:EE:01
PHY
µC
DHCP-Discover
Check at which port the MAC-Address has been received and assign the IP-address according to IP-assignment table
DHCP-Offer
DHCP-Request
DHCP-Ack
ECU at Port 3 has IP-address 192.168.0.1

Figure 7.5: IP-address assignment via DHCP

[SWS_EthSwt_00087] ⌈The function EthSwt_StoreConfiguration shall request to store the configuration of the learned MAC/Port tables of a switch in a persistent manner. This can be done in two ways: 1.) Reading out the parameters and storing them in the NV-RAM of the host CPU using the NV-RAM manager. 2.) Advising the switch to store the configuration data in its local NV-RAM.

In both alternatives <EthSwtPersistentConfigurationResultCallback> shall be invoked if EthSwtPersistentConfigurationResultCallback is configured. In case of storage to switch local NV-RAM, JobResult shall be set to NVM_REQ_OK to indicate success or to NVM_REQ_NOT_OK to indicate failure.⌋*(SRS_Eth_00087, SRS_Eth_00122)*

[SWS_EthSwt_00092] ⌈The function EthSwt_ResetConfiguration shall request to reset the configuration of the learned MAC/Port tables of a switch in a persistent manner. This can be done in two ways: 1.) Overwriting the learned parameters in the

NV-RAM of the host CPU with preconfigured default values. 2.) Advising the switch to reset the learned configuration data in its local NV-RAM.

In both alternatives <EthSwtPersistentConfigurationResultCallback> shall be invoked if EthSwtPersistentConfigurationResultCallback is configured. In case of storage to switch local NV-RAM, JobResult shall be set to NVM_REQ_OK to indicate success or to NVM_REQ_NOT_OK to indicate failure.⌋*(SRS_Eth_00122, SRS_Eth_00087)*

[SWS_EthSwt_00061] ⌈The function EthSwt_GetPortMacAddr shall return the port index over which the given MAC-address is reachable within the indexed switch. If for the PortIdxPtr the maximal possible value (255) is returned the given MAC address cannot be reached via a port of this switch. If multiple ports were found the API returns E_NOT_OK.⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00163] ⌈The Ethernet Switch driver shall support an API which allows to reset learned parameters like address resolution tables by using the API EthSwt_ResetConfiguration.⌋*()*

[SWS_EthSwt_00407] ⌈Unused ARL table entries shall be removed from the ARL table after the timeout configured via EthSwtArlTableEntryTimeout, if this parameters is present in the configuration.⌋*()*

7.1.5.2 Frame forwarding process

As shown in Figure 7.6, the Ethernet switch consists of a certain number of Ethernet switch ports. A single physical Ethernet port is logically divided in an ingress port and an egress port. A frame is received by an Ethernet switch port in the role of an ingress port. This frame is processed within the Ethernet switch and most likely forwarded to one or more Ethernet switch ports in the role of an egress port. This process is called the "frame forwarding process". A frame forwarding process considers among others the follwoing points:

- An Ethernet frame is typically not forwarded to the Ethernet switch port where it has been received.
- A unicast Ethernet frame could be forwarded to exactly one egress port. (Please note, for some reasons (e.g. mirroring or unkown unicast Ethernet frame) a unicast Ethernet frame may forwarded to multiple egress ports)
- A multicast Ethernet frame (e.g. SOME/IP-SD offer frame) could be forwarded to one or more egress ports.
- A broadcast Ethernet frame (e.g. ARP frame) is forwarded to all egress ports except the Ethernet switch port from where the frame has been received.

Please note: The route of the frame within an Ethernet switch from an ingress port to one or multipe egress ports is called "internal frame route".

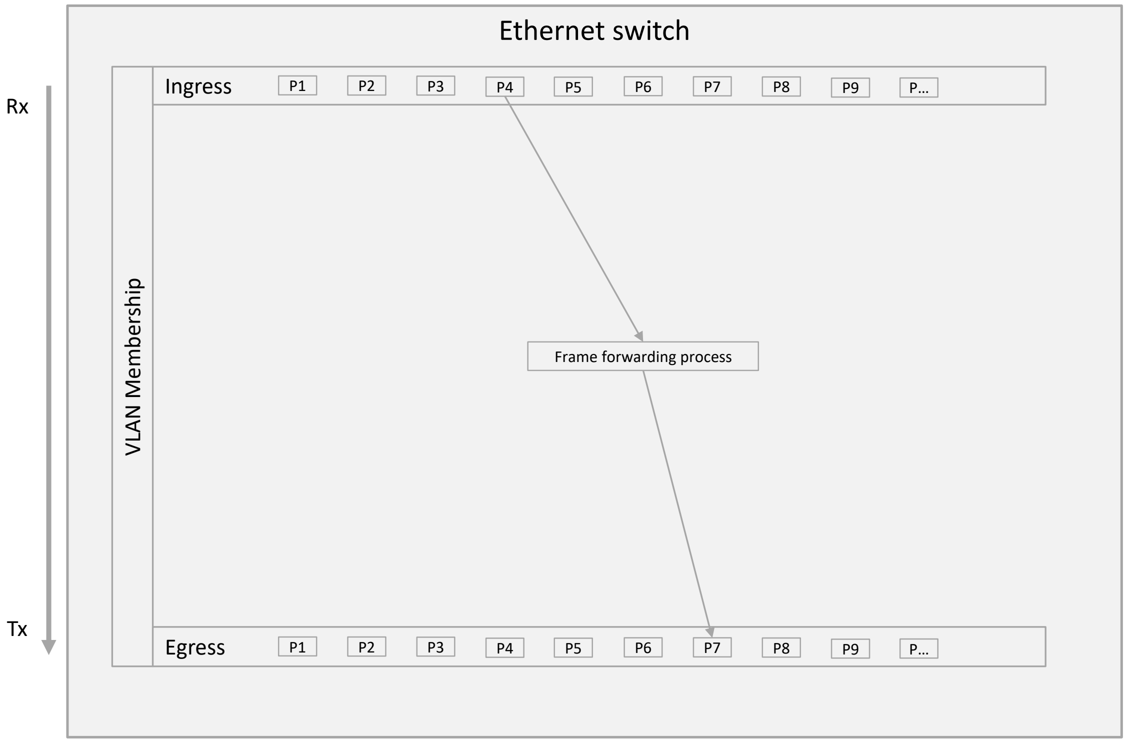


Figure 7.6: Simplified frame forwarding process within an Ethernt switch

The forwarding process consists of multiple frame processing stages. Some frame processing stages are always performed (e.g. check of VLAN membership) and some frame processing stages are performed if they are configured (e.g. flow metering). A frame processing stage may qualify a received frame as invalid. Such a frame is discarded and therefore not forwarded to the subsequential frame processing stage. [9, IEEE Std 802.1Q] specifies the frame forwarding process and particular frame processing stages. Figure 7.7 shows an overview of the processing stages which are supported by AUTOSAR (please refer to section 4.1)

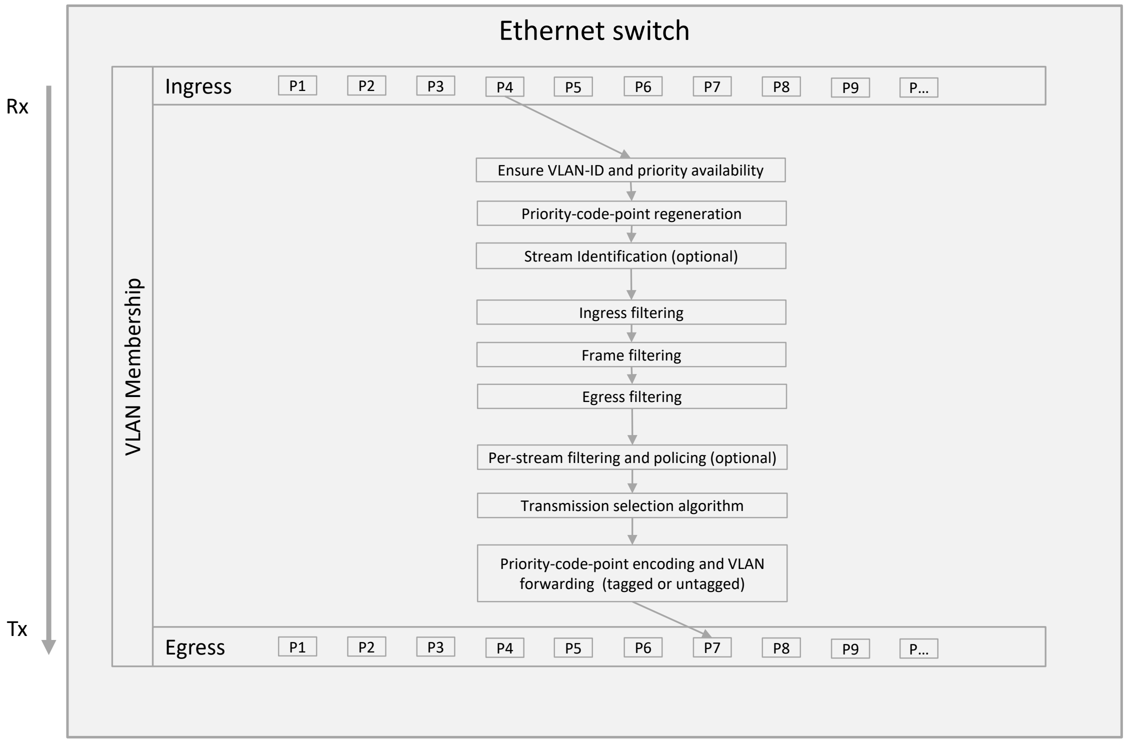

Ethernet switch
Rx
Ingress
P1
P2
P3
P4
P5
P6
P7
P8
P9
P...
VLAN Membership
Ensure VLAN-ID and priority availability
Priority-code-point regeneration
Stream Identification (optional)
Ingress filtering
Frame filtering
Egress filtering
Per-stream filtering and policing (optional)
Transmission selection algorithm
Priority-code-point encoding and VLAN forwarding (tagged or untagged)
Tx
Egress
P1
P2
P3
P4
P5
P6
P7
P8
P9
P...

Figure 7.7: Overview of frame processing stages within an Ethernet switch supported by AUTOSAR

Most likely Ethernet frames are not modified in an Ethernet switch. Important information which impact the Ethernet frame attributes (e.g. VLAN-ID, priority) are kept in a separate memory section for each Ethernet frame while traveling through the processing stages. In this specification such a section is called "Ethernet frame meta information". If an Ethernet frame arrive at an ingress port, a Ethernet frame meta information is assigned to this Ethernet frame. The Ethernet frame meta information contain important state values (e.g. stream handle id, outbound priority value). Available entries in the Ethernet frame meta information could be updated (e.g. destination port vector, priority). The Ethernet frame meta information is available along the internal frame route. At the very last processing stage all relevant entries of the Ethernet frame meta information which impact the Ethernet frame attributes are written to the Ethernet frame.

The following chapters describe the behaviour of the supported processing stages.

7.1.5.2.1 Ensure VLAN-ID and priority availability

AUTOSAR Ethernet switches are exclusively VLAN-aware (refer to chapter section 4.1). If an Ethernet frame is received, then an AUTOSAR Ethernet switch ensures the availability of a VLAN-ID and a priority for this Ethernet frame before forwarding to the next processing stage. Therefore the so-called "frame type acceptance filter", "port-based VLAN classification" and "priority-code-point decoding" is performed by an

Ethernet switch. The following chapters describe how to ensure VLAN-ID and priority availability.

7.1.5.2.1.1 Handling of untagged Ethernet frames

Ethernet frames carrying a TPID set to 0x8100 are considered as tagged Ethernet frames, i.e. they carry a VLAN-tag. For Ethernet frames which are received without an VLAN-tag, a specific Ethernet switch handling could be configured via the AUTOSAR Ethernet switch driver.

There are two ways to handle untagged Ethernet frames at ingress side:

- Drop all untagged Ethernet frames at ingress side of the Ethernet port where the Ethernet frame was received
- Tag all untagged Ethernet frames at ingress side with a default VLAN and default VLAN priority.

Note: The handling of untagged Ethernet frames by the Ethernet switch is expected to be performed before all other modifications of the VLAN (e.g. VLAN modification). This applies also for the VLAN priority handling, which is expected to be performed before a Traffic Class assignement (see subsequential chapters) is done.

Basically, an Ethernet switch tag all Ethernet frames internally for its internal processing with hardware specific default value for a VLAN-tag. This hardware specific default value can be overwritten via `EthSwtPortIngressDefaultVlan` and `EthSwtPortIngressDefaultPriority` per Ethernet switch port

[SWS_EthSwt_CONSTR_00452]{DRAFT} ⌈If `EthSwtPortIngressDefaultVlan` and `EthSwtPortIngressDefaultPriority` for a particular Ethernet port is available, then the default VLAN and the default priority of this hardware Ethernet switch port shall be configured with the available values.⌋*(SRS_Eth_00121)*

Implementation Hint: A VLAN-tag consist of a VLAN-ID and the VLAN priority.

[SWS_EthSwt_CONSTR_00453]{DRAFT} ⌈A configuration of an ingress port shall be rejected as invalid, where `EthSwtPortIngressDropUntagged` is set to `TRUE`, and the parameters `EthSwtPortIngressDefaultVlan` and `EthSwtPortIngressDefaultPriority` are configured (multiplicity of both parameters are `0`).⌋*(SRS_Eth_00121)*

[SWS_EthSwt_00611]{DRAFT} ⌈If parameter `EthSwtPortIngressDropUntagged` of an specific ingress port is set to `TRUE` and a Ethernet frame without a VLAN-tag (untagged Ethernet frame) is received at this specific ingress port, then this Ethernet frame shall be dropped.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_CONSTR_00454]{DRAFT} ⌈A configuration of an specific ingress port shall either have both parameters `EthSwtPortIngressDefaultVlan` and `EthSwtPortIngressDefaultPriority` configured or none of them.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_00612]{DRAFT} ⌈If parameter EthSwtPortIngressDropUntagged of an specific ingress port is set to FALSE, the parameters EthSwtPortIngressDefaultVlan and EthSwtPortIngressDefaultPriority are configured and a Ethernet frame without a VLAN-tag (untagged Ethernet frame) is received at this specific ingress port, then the default vlan and default priority shall be assigend to this Ethernet frame and handled for further processing by the Ethernet switch.⌋*(SRS_Eth_00121)*

Note: If a Ethernet frame shall be sent without a VLAN-tag (untagged Ethernet frame) of a specific VLAN-ID and on a particular egress port, then EthSwtVlanForwardingType of this VLAN-ID at this Ethernet port need to be set to ETHSWT_SENT_UNTAGGED (see subparagraph 7.1.5.2.4.1)

7.1.5.2.1.2 Handling of double tagged Ethernet frames

AUTOSAR support to configure the handling for so-called "double tagged" Ethernet frames per Ethernet switch. Double tagged Ethernet frames contain two VLAN-tags. The first tag is called "S-TAG" (service provider tag) and the second tag is called "C-TAG" (customer tag). Per default the forwarding of double tagged frames is supported. In the forwarding process the S-TAG is considered. For some use cases it is necessary to avoid handdling of such Ethernet frames. Therefore a boolean parameter EthSwtDropDoubleTagged is available. The Ethernet Switch Driver supports a configuration of dropping double tagged frames via the configuration parameter EthSwtDropDoubleTagged, if the Ethernet switch hardware supports dropping of double tagged Ethernet frames.

[SWS_EthSwt_00233] ⌈If parameter EthSwtDropDoubleTagged is set to TRUE, double tagged Ethernet frames shall be dropped independent on which Ethernet switch port this Ethernet frame has been received.⌋*(SRS_Eth_00114)*

Note: Dropping of double tagged Ethernet frames depend on the configuration of the TPID for the outer VLAN-tag.

7.1.5.2.2 Priority-Code-Point-Regeneration

If an Ethernet frame pass the processing stage to ensure VLAN-ID and priority availability, then the co-called "priority regeneration" is performed. This processing step is mandatory and will always be executed. The PCP-field (priority code point) within an VLAN-tag of an received Ethernet frame can be modified at an ingress port of an Ethernet switch. For this purpose a so-called priority regeneration table has to be defined:

Priority Regeneration Table								
Ingress PCP	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Regenerated PCP	0	1	2	3	4	5	6	7

Table 7.1: In this table, the "Ingress PCP" is mapped to the "Regenerated PCP".

[SWS_EthSwt_00178]{OBSOLETE} ⌈Replaced by [SWS_EthSwt_00614]. The switch configuration shall support the configuration how the PCP field of incoming Ethernet frames will be modified before they are forwarded to the egress port, i.e. a priority regeneration table can be configured (Please refer to EthSwtPortPriorityRegeneration,EthSwtPortPriorityRegenerationIngressPCP and EthSwtPortPriorityRegenerationRegeneratedPriority.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_00614]{DRAFT} ⌈If an Ethernet frame is forwarded within an Ethernet switch, then the Ethernet switch shall perform a PCP regeneration for the PCP of this Ethernet frame by considering the configured priority regeneration table (see EthSwtPortPriorityRegeneration,EthSwtPortPriorityRegenerationIngressPCP and EthSwtPortPriorityRegenerationRegeneratedPriority) available at the EthSwtPortIngress, where this Ethernet frame was received.⌋*(SRS_Eth_00180)*

Please note: If no modification is required, than the PCP ingress and PCP regenerated should have the same value.

7.1.5.2.3 Stream identification

If an Ethernet frame pass the "priority-code-point regeneration" than a so-called "stream identification" could be performed by an Ethernet switch, if this processing stage is configured. Otherwise the Ethernet switch forward the Ethernet frame to the next processing stage "ingress filtering".

[14, IEEE Std 802.1CB] defines stream identification. A stream identification is the mandatory pre-condition to perform "per-stream filtering and policing" in a later processing stage. The stream identification function is used to identify a Ethernet frame according particular Ethernet frame attributes. If the Ethernet frame match, then a so-called "stream handle id" is assigned to the Ethernet frame. Therefore the stream handle id is added to Ethernet frame meta information. If the Ethernet frame reaches the processing stage "per-stream filtering and policing", then this stream handle id is used to find a corresponding EthSwtStreamFilterEntry to perform e.g. a flow metering.

If the Ethernet switch HW supports this feature, then it can be configured by using the sub container EthSwtStreamIdentificationTable. EthSwtStreamIdentificationTable represents a table, where multipe EthSwtStreamIdentificationEntrys form an ordered list. Each entry represents an stream identification definition. The stream identification definition applies to streams within an Ethernet switch. Ethernet frames (i.e. streams) are received by an ingress port. A stream identification could be defined in dependency of ingress ports or independent of ingress ports. Therefore a stream identification could reference 0...n ingress ports (EthSwtPortIngress).

[SWS_EthSwt_CONSTR_00464]{DRAFT} ⌈If an EthSwtStreamIdentificationEntry is configured, then it shall be possible that this EthSwtStreamIdentificationEntry could reference none, one or multiple ingress via EthSwtStreamIdentificationIngressPortRef.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00465]{DRAFT} ⌈If an EthSwtStreamIdentificationEntry references one or multiple ingress ports, then the stream identification shall be processed for streams received via any of the referencing ingress ports.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00467]{DRAFT} ⌈If an EthSwtStreamIdentificationEntry references no ingress ports, then the stream identification shall be processed for all streams received via any ingress port.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_CONSTR_00468]{DRAFT} ⌈If a configured EthSwtStreamIdentificationEntry references an EthSwtPortIngress, then this stream identification definition shall reference the same EthSwtPortIngress exclusively one time.⌋*(SRS_Eth_00178)*

The configuration of an EthSwtStreamIdentificationEntry which references ingress ports define the dependency between an EthSwtStreamIdentificationEntry and the stream route of a received Ethernet frame (i.e. stream) within an Ethernet switch. Therefore this stream route is called the "internal stream route".

[SWS_EthSwt_00469]{DRAFT} ⌈If an Ethernet frame (i.e stream) has been received, then the stream identification shall be processed by configured EthSwtStreamIdentificationEntry where the internal stream route match.⌋*(SRS_Eth_00178)*

Multiple EthSwtStreamIdentificationEntrys are configured as an ordered list of an EthSwtStreamIdentificationTable. The position within the ordered list is defined with the configured value of EthSwtStreamIdentificationPosition. The list is processed in ascending order by the Ethernet switch. As soon as the first EthSwtStreamIdentificationEntry matches, the Ethernet switch will threat this Ethernet frame according to the configuration that is associated with this EthSwtStreamIdentificationEntry. Subsequential EthSwtStreamIdentificationEntrys of the odered list will not be applied. In case a received Ethernet frame does not match any of the EthSwtStreamIdentificationEntrys, the Ethernet frame will be forwarded to the next frame processing stage.

[SWS_EthSwt_CONSTR_00470]{DRAFT} ⌈Every EthSwtStreamIdentificationEntry shall have a unique position value configured via EthSwtStreamIdentificationPosition. The value shall start with 0 and continue in ascending order with no gaps for each subsequential EthSwtStreamIdentificationEntry.⌋*(SRS_Eth_00178)*

Note: The position value forms a ordered list of EthSwtStreamIdentificationEntrys

[SWS_EthSwt_00471]{DRAFT} ⌈If a Ethernet frame (i.e stream) has been received, then the Ethernet switch shall check for a matching EthSwtStreamIdentificationEntry in ascending order according the EthSwtStreamIdentificationPosition, starting with EthSwtStreamIdentificationPosition configured with value 0.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00472]{DRAFT} ⌈If a received Ethernet frame (i.e stream) does not match any EthSwtStreamIdentificationEntrys, the Ethernet frame shall be forwarded to the next frame processing stage without applying any further stream identification handling.⌋*(SRS_Eth_00178)*

An EthSwtStreamIdentificationEntry consist of the EthSwtStreamFilterRule (multiplicity 1) a EthSwtStreamHandleAssignment (multiplicity 1) and additionally of the optional element EthSwtStreamFilterAction.

The elements of an EthSwtStreamIdentificationEntry define the filter rules and filter actions. The order to perform the stream identification (apply filter rules, perform filter actions and further stream handling) is statically defined.

[SWS_EthSwt_00475]{DRAFT} ⌈If a Ethernet frame (i.e. stream) has been received and the internal stream route match to a configured EthSwtStreamIdentificationEntry, then this stream identification shall be processed in the following order:

1. Apply the EthSwtStreamFilterRule
2. If the EthSwtStreamFilterRule identifies a match, the configured stream handle id (see EthSwtStreamHandleAssignment) shall be added to the Ethernet frame meta information and, if EthSwtStreamFilterAction is configured, consider the filter action to be performed

⌋*(SRS_Eth_00178)*

Note:

- It is implementation specific in which processing stage a configured EthSwtStreamFilterAction is performed. For example, if the filter action EthSwtStreamFilterActionDropFrame set to TRUE and a stream is identified, then it makes sense to immediatly drop the Ethernet frame and abort the forwarding process. But if the filter action is configured to EthSwtStreamFilterActionDestinationPortModification, then the action should be considered after egress filtering is finalized.
- If a filter rule is empty (no filter rule primitves configured (see subparagraph 7.1.5.2.3.1)), then the Ethernet frame (i.e. stream) pass this filter per default. Thus, the configured stream handle id (see EthSwtStreamHandleAssignment) is added to the Ethernet frame meta information. The stream processing proceed with the "ingress filtering"

An EthSwtStreamIdentificationEntry is considered as an empty stream identification definition, where none of the optional elements are defined. Thus, incoming Ethernet frames which match the internal stream route of an empty stream identification definition, always idenfied as match of this EthSwtStreamIdentificationEntry per default.

[SWS_EthSwt_00476]{DRAFT} ⌈A configured EthSwtStreamIdentificationEntry where no optional elements are configured, shall be considered as empty stream identification, where incoming Ethernet frames always pass.⌋*(SRS_Eth_00178)*

7.1.5.2.3.1 Stream identification and filter rules

An EthSwtStreamIdentificationEntry has always a filter rule configured (EthSwtStreamFilterRule). EthSwtStreamFilterRule defines which parts of a received Ethernet frame are considered for the filtering (e.g. MAC source address, IP destination address a.s.o). The EthSwtStreamFilterRule could contain multiple filter rules. One particular filter rule (e.g. EthSwtStreamFilterMACDestAddress) is called a "filter rule primitive".

[SWS_EthSwt_00477]{DRAFT} ⌈If an EthSwtStreamFilterRule have multiple filter rule primitives configured (e.g. EthSwtStreamFilterMACSrcAddress and EthSwtStreamFilterVlanId), then the filter rule primitives shall be considered as AND-linked filter rules.⌋*(SRS_Eth_00178)*

Note: A received Ethernet frame (i.e. stream) matches the filter, if all configured filter rule primitives are matches. E.g. if EthSwtStreamFilterMACSrcAddress and EthSwtStreamFilterVlanId is configured, then a stream matches, if the source MAC address AND the VLAN-ID match the defined values. A stream where for example only the EthSwtStreamFilterMACSrcAddress matches is considered as NOT matching Ethernet frame.

[SWS_EthSwt_00478]{DRAFT} ⌈If a Ethernet frame (i.e. stream) has been received, the internal stream route for this Ethernet frame matches to an configured EthSwtStreamIdentificationEntry, the EthSwtStreamFilterRule of this EthSwtStreamIdentificationEntry have filter rule primitives configured (e.g. EthSwtStreamFilterMACSrcAddress and EthSwtStreamFilterVlanId) and the Ethernet frame matches all configured filter rule primitives, then this Ethernet frame shall be qualified as matching stream.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00479]{DRAFT} ⌈If an Ethernet frame (i.e. stream) has been received, the internal stream route for this Ethernet frame matches to an configured EthSwtStreamIdentificationEntry and a particular filter rule primitive in the EthSwtStreamFilterRule of this entry is not configured, then this filter rule primitive shall be considered as matching filter rule primitive.⌋*(SRS_Eth_00178)*

Note: Not configured filter rule primitives within an existing EthSwtStreamFilterRule are called "wildcard filter rule primitives". In order to qualify an Ethernet frame (i.e. stream) as matching stream, an Ethernet frame must match all filter rule primitives, as all filter rule primitives are AND-linked. Therefore, an empty EthSwtStreamFilterRule, i.e. without any filter rule primitives configured, will match every received Ethernet frame where the internal frame route of Ethernet frame match to configuration of the according EthSwtStreamIdentificationEntry.

7.1.5.2.3.2 Stream identification and filter action

An EthSwtStreamIdentificationEntry could define a filter action (EthSwtStreamFilterAction). The filter action describe the expected behaviour, if a matching stream has been detected. A filter action always refer to the filter rule of the same EthSwtStreamIdentificationEntry.

[SWS_EthSwt_00480]{DRAFT} ⌈If a Ethernet frame (i.e. stream) has been qualified as matching stream and an EthSwtStreamFilterAction is configured, then this filter action shall be applied on this Ethernet frame.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00481]{DRAFT} ⌈If a filter action is applied on a Ethernet frame (i.e. stream) and the corresponding EthSwtStreamFilterAction has EthSwtStreamFilterActionDropFrame set to TRUE, then this Ethernet frame shall be dropped.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00482]{DRAFT} ⌈If a filter action is applied on a Ethernet frame (i.e. stream) and the corresponding EthSwtStreamFilterAction EthSwtStreamFilterActionBlockSource set to TRUE, then this Ethernet frame and all sub sequential receptions of Ethernet frames with the same source MAC address shall be blocked.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00483]{DRAFT} ⌈If a filter action is applied on a Ethernet frame (i.e. stream) and the corresponding EthSwtStreamFilterAction has an EthSwtStreamFilterActionVlanModification configured, then the VLAN-ID of this Ethernet frame shall be modified with the configured VLAN-ID given by EthSwtStreamFilterActionVlanModificationVlanId.⌋*(SRS_Eth_00178)*

An Ethernet switch determine the egress destination of an Ethernet frame within the forwarding process. An egress destination for an Ethernet frame could include one or multiple destination ports (egress ports). This egress destination could be modified if EthSwtStreamFilterActionDestinationPortModification is configured. The egress destination which is used for the modification is configured as reference to the according egress port(s).

[SWS_EthSwt_00484]{DRAFT} ⌈If a filter action EthSwtStreamFilterActionDestinationPortModification is configured, then the egress destination which is used for the modification shall be determined according the configured references to egress ports via EthSwtStreamFilterActionDestinationPortModificationEgressPortRef.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_CONSTR_00485]{DRAFT} ⌈If a filter action EthSwtStreamFilterActionDestinationPortModification is configured, then this EthSwtStreamFilterActionDestinationPortModification shall reference the same EthSwtPortEgress via EthSwtStreamFilterActionDestinationPortModificationEgressPortRef exclusively one time.⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00486]{DRAFT} ⌈If a filter action is applied on a Ethernet frame (i.e. stream), the corresponding EthSwtStreamFilterAction has an EthSwtStreamFilterActionDestinationPortModification configured and the

EthSwtStreamFilterActionDestinationPortModificationType is set to ETHSWT_STREAM_EGRESS_DESTINATION_OVERWRITE, then the egress destination of this Ethernet frame shall be overwritten with the configured egress destination (see EthSwtStreamFilterActionDestinationPortModificationEgressPortRef).⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00487]{DRAFT} ⌈If a filter action is applied on a Ethernet frame (i.e. stream), the corresponding EthSwtStreamFilterAction has an EthSwtStreamFilterActionDestinationPortModification configured and the EthSwtStreamFilterActionDestinationPortModificationType is set to ETHSWT_STREAM_EGRESS_DESTINATION_EXTEND, then the egress destination of this Ethernet frame shall be extended with the configured egress destination (see EthSwtStreamFilterActionDestinationPortModificationEgressPortRef).⌋*(SRS_Eth_00178)*

[SWS_EthSwt_00610]{DRAFT} ⌈If a filter action is applied on a Ethernet frame (i.e. stream), the corresponding EthSwtStreamFilterAction has an EthSwtStreamFilterActionDestinationPortModification configured and the EthSwtStreamFilterActionDestinationPortModificationType is set to ETHSWT_STREAM_EGRESS_DESTINATION_LIMIT, then the egress destination of this Ethernet frame shall be extended with the configured egress destination (see EthSwtStreamFilterActionDestinationPortModificationEgressPortRef, but limited to those referenced egress ports where this Ethernet frame is allowed to be transmitted according to the egress port state (e.g. VLAN membership, assigned MAC address)).⌋*(SRS_Eth_00178)*

Note: use case for [SWS_EthSwt_00610] is to limit/restrict the egress ports on which packet are allowed to egress the Ethernet switch.

7.1.5.2.4 Ingress filtering

If an Ethernet frame pass the "priority-code-point regeneration" and the optional "stream identification" frame processing stage, then a so-called ingress filtering is performed by an Ethernet switch. The following sub chapters describe the details of the processing.

7.1.5.2.4.1 Vlan-Membership

For each Ethernet switch port a VLAN membership could be defined. An Ethernet switch port could be member of 0..* VLANs. The VLAN membership impacts the frame processing. A VLAN Membership describes ingress and egress behavior in terms of filtering, tagging or untagging.

Vlan-Membership - ingress implications

If an Ethernet frame has been received, then the Ethernet switch inspect the frame regarding a VLAN-tag. If an VLAN-tag within the received Ethernet frame exist, the Ethernet switch evaluates the received VLAN identifier (VLAN-ID). If the Ethernet switch port from where the Ethernet frame has been received is member of the VLAN which is associated with the received VLAN-ID, then the frame processing will continue, otherwise the frame is discarded and no further frame processing will be performed. Thus, all supported VLAN-IDs are configured in `EthSwtVlanMembership`.

[SWS_EthSwt_00601]{DRAFT} ⌈If an Ethernet switch port, from where the Ethernet frame has been received, is member of the VLAN according to the configuration of `EthSwtVlanMembership` and the VLAN-ID of the received Ethernet match to the configured VLAN membership of this Ethernet switch port, then the frame processing shall continue, otherwise the Ethernet frame shall be discarded and frame processing aborted.⌋*(SRS_Eth_00114)*

Vlan-Membership - egress implications

Please note: VLAN-membership egress handling is performed in processing stage "egress filtering", but it is described in this section, since this gives an overview of the VLAN-membership handling.

If a received Ethernet frame with an particular VLAN-ID passed all processing stages, the Ethernet switch has to add the frame to an egress queue according to the internal frame route. Thereby, the VLAN membership defines with `EthSwtVlanForwardingType`, if an Ethernet frame with a particular VLAN-ID shall be sent on the affected port with a VLAN-tag (`ETHSWT_SENT_TAGGED`, or if this Ethernet frame shall be sent on the affected port without the VLAN-tag (`ETHSWT_SENT_UNTAGGED`), or if this Ethernet frame shall not be sent on the affected port (`ETHSWT_NOT_SENT`).

For each VLAN-ID a table is necessary which stores at which egress port an Ethernet frame with the corresponding VLAN-ID is sent tagged, sent untagged or not sent. For an 8-port switch, this table could look like the following example where T stands for tagging, U for untagging, N for not sent and "-" not member of this VLAN:

VLAN Forwarding Table								
VLAN-ID	**Port Number**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	T	T	-	U	-	-	-	T
2	T	U	-	T	-	-	-	N
. . .								
4094								

Examples of communication scenarios:

- Incoming Ethernet frames which contain a VLAN-ID of e.g. 1 can be forwarded to the ports 1, 2, 4, and 8. At ports 1, 2, and 8 these Ethernet frames will be transmitted with the VLAN-tag and at port 4 the VLAN-tag will be removed. Ethernet frames which contain a VLAN-ID e.g. 1 received on ports 3,5,6 and 7 will be discarded.
- If a broadcast message with e.g. VLAN-ID 2 will be received at port 2, it will be forwarded to port 1,4 and 8. At ports 1 and 4 these Ethernet frames will be transmitted with the VLAN-tag and on port 8 it will not be send, since the forwarding type is configured with N (ETHSWT_NOT_SENT). The other ports 3, 5, 6, and 7 are not in the same VLAN. Thus, the Ethernet frame will not be forwarded to these egress ports.
- If a broadcast message with e.g. VLAN-ID 2 will be received at port 8, it will be forwarded to port 1,2 and 4. At ports 1 and 4 these Ethernet frames will be transmitted with the VLAN-tag and on port 2 the VLAN-tag will be removed. The other ports 3, 5, 6, and 7 are not in the same VLAN. Thus, the Ethernet frame will not be forwarded to these egress ports.

The table considers only messages, which contain a VLAN-ID within the Ethernet switch.

[SWS_EthSwt_00134]{OBSOLETE} ⌈Replaced by [SWS_EthSwt_00450]. The switch configuration shall support the configuration how packets will be forwarded with respect to configured VLANs by using the configuration parameters of the subcontainer EthSwtVlanMembership.⌋*(SRS_Eth_00121, SRS_Eth_00114)*

[SWS_EthSwt_00450] ⌈If an Ethernet frame has been received and the Ethernet frame passes the Egress filtering, then the Ethernet frame shall be forwarded to the Egress port according the EthSwtVlanForwardingType configuration:

- If egress port is configured to ETHSWT_SENT_TAGGED, then the Ethernet frame shall be transmitted with a VLAN tag,
- else If egress port is configured to ETHSWT_SENT_UNTAGGED, then the Ethernet frame shall be transmitted without a VLAN tag,
- else if egress port is configured to ETHSWT_NOT_SENT, then the Ethernet frame shall be dropped

⌋*()*

Note: VLAN-Memberships of a port are modeled with the container EthSwtVlanMembership where the associated ports are referenced via EthSwtVlanMembershipPortRef and the according EthSwtVlanForwardingType is configured.

7.1.5.2.4.2 VLAN-modification at ingress side

It is possible to define a port-based modification of the VLAN-ID or an insertion of a VLAN-ID into a received Ethernet frame. (Please note, as described in subsubsection 7.1.5.2, the Ethernet frame itself will not be modified, but the change is stored in the Ethernet frame meta information to be considered in the subsequential forwarding process) This is specified with another table, e.g.:

Ingress VLAN Modification/Insertion Table								
Port Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
VLAN-ID	2	-	-	6	-	-	-	-

In this example, all incoming Ethernet frames at port 1 will get the VLAN-ID 2 no matter if they already had one before. At port 4, all incoming Ethernet frames will get 6 as their VLAN-ID. At the remaining ports, no VLAN-IDs will be inserted and an existing VLAN-ID in the Ethernet frame will remain without modification.

[SWS_EthSwt_00135]{OBSOLETE} ⌈Replaced by [SWS_EthSwt_00451]. The switch configuration shall support the configuration how VLANs will be inserted into Ethernet frames or existing VLANs will be modified by the configuration `EthSwtPortIngressVlanModification`.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_00451] ⌈If an Ethernet frame has been received, the `EthSwtPortIngressVlanModification` is configured and the VLAN-ID of the received Ethernet frame match to the configured VLAN-ID of `EthSwtPortIngressVlanModification`, then the Ethernet switch shall consider the configured VLAN-ID for this Ethernet frame.⌋*()*

7.1.5.2.4.3 Priority handling

A VLAN-tag of an Ethernet frame consist of a VLAN-ID and the VLAN priority. The VLAN priority within a VLAN-tag is called the PCP-field (priority code point). The PCP defines the priority with which this Ethernet frame shall be handled in an Ethernet network. The PCP is a 3bit value and defines the lowest priority with 0 and highest priority with 7. The prioritisation of Ethernet traffic supports the quality of service technique on a switched Ethernet network. From the Ethernet switch perspective the priority received with an Ethernet frame could be re-defined for the internal frame processing. AUTOSAR supports the following methodes to re-define the internal priority of a received Ethernet frame for the internal frame processing:

- priority regeneration based on ingress PCP (regenerated priority)
- definition of an internal priority value (IPV) which could be configured if stream filtering is used (see paragraph 7.1.5.2.7)

The methods to re-define the (internal) priority of a received Ethernet frame could co-exit. It is possible to have an IPV configured besided the table for priority regeneration. The usage of a re-defined priority differs partly:

- regenerated priority and internal priority values are used for the traffic class assignment of an Ethernet frame
- regenerated priority is used for the outbound priority assignment

Internal priority value

An internal priority value could be defined by configuring an EthSwtPSFP (per-stream filtering and policing) in combination with an EthSwtStreamFilterTable, where an EthSwtStreamFilterEntry references an EthSwtStreamGateEntry which has an EthSwtStreamGateIPV. As described before, the internal priority value is used to assign a traffic class to an Ethernet frame. Please refer to paragraph 7.1.5.2.7 for further description regarding the configuration.

7.1.5.2.4.4 Priority to traffic class assignment

The Ethernet switch performs a so-called "priority to traffic class assignment" for each received Ethernet frame that arrives at the ingress filtering processing stage. The selected priority is used to assign a particular Ethernet frame to a dedicated traffic class (please note: the usage of a traffic class is described in paragraph 7.1.5.2.9). The priority selection of a particular Ethernet frame for the internal Ethernet frame processing is depicted in Figure 7.8.

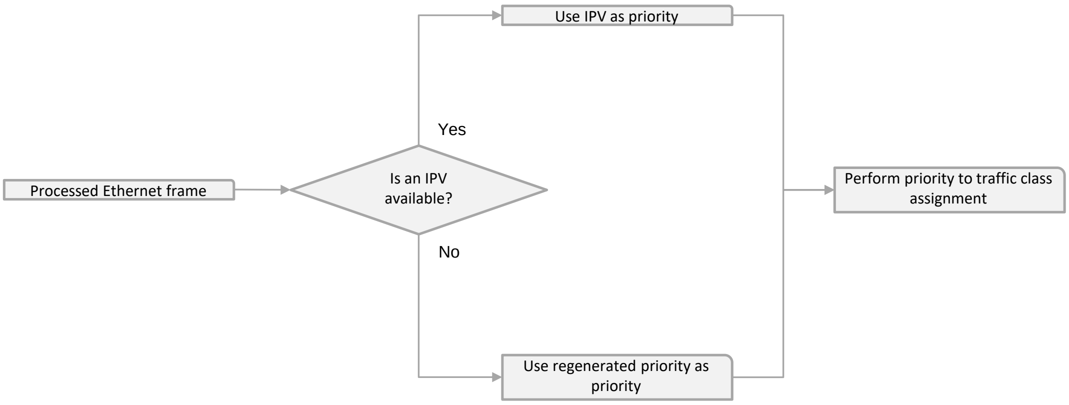

Use IPV as priority
Yes
Processed Ethernet frame
Is an IPV available?
Perform priority to traffic class assignment
No
Use regenerated priority as priority

Figure 7.8: Priority Selection process to select the priority for the internal Ethernet frame procession

The Ethernet switch has to check if the internal frame route of a received Ethernet frame is processed by a per-stream filtering to determine the stream gate and the availability of the according "internal priority value" (IPV, see EthSwtStreamGateIPV). If an IPV is available, then the Ethernet switch selects the IPV as priority for the traffic class assignment, otherwise the regenerated priority value (see EthSwtPortPriorityRegeneration).

[SWS_EthSwt_00455]{DRAFT} ⌈If an Ethernet switch has to select a priority value for a received Ethernet frame, the Ethernet switch detects that the received Ethernet frame is handled by an EthSwtStreamFilterEntry and the EthSwtStreamFilterEntry references a stream gate (see EthSwtStreamGateEntry), then the configured IPV (EthSwtStreamGateIPV) shall be used as priority to map this Ethernet frame to a traffic class. Otherwise the Ethernet switch shall use the regenerated priority value (see EthSwtPortPriorityRegeneration).⌋*(SRS_Eth_00121)*

Note: see paragraph 7.1.5.2.7 for further details about per-stream filtering and policing.

For the mapping of an Ethernet frame to a certain traffic class, a table is necessary where the priorities are mapped to traffic classes. Eight traffic classes are specified, similarly to the eight specified VLAN priorities. Most likely eight priorities are mapped to the eight traffic classes. It is also supported to map multiple priorities to one traffic class. The table for the priority based mapping can be defined at the ingress port. Table 7.2 shows an example for priority to traffic class mapping.

Priority	Traffic Class
Prio 0	7
Prio 1	6
Prio 2-7	5
-	4
-	3
-	2
-	1
-	0

Table 7.2: In this table, "Priorities" are mapped to a particular "Traffic class".

[SWS_EthSwt_00456] ⌈If EthSwtPortTrafficClassAssignment is configured and a Ethernet frame is received at an ingress port, then this Ethernet frame shall be assigned to the traffic class which corresponds to the priority configured at EthSwtPriorityTrafficClassAssignment.⌋*()*

As alternative to configure a table for priority to traffic class assignment, it is possible to configure an ingress port to traffic class assignment. This is used to simplify the configuration. This configuration will result also in a table for priority to traffic class assignment. All incoming traffic of the according ingress port is mapped to the according traffic class. Table 7.3 shows an example for ingress port to traffic class mapping.

Ingress port	Traffic Class
e.g. Port2, Port3, Port4	7
e.g. Port1	6
-	5
-	4
-	3
-	2
-	1
-	0

Table 7.3: In this table, "Ingress ports" are mapped to a particular "Traffic Class"

[SWS_EthSwt_00179] ⌈If an untagged Ethernet frame is received and EthSwtPortIngressDropUntagged is set to False for an ingress port, the EthSwtPriorityTrafficClassAssignment container shall be used to set the default traffic class for the default VLAN configured at EthSwtPortIngressDefaultVlan.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_CONSTR_00457]{DRAFT} ⌈If the same EthSwtPortIngress need to configure a priority to traffic class assignment, then exclusively either EthSwtPortTrafficClassAssignment or EthSwtPortPriorityTrafficClassAssignment shall be configured.⌋*(SRS_Eth_00121)*

7.1.5.2.4.5 Outbound priority assignment

It is possible to define a particular VLAN priority at the ingress port for outgoing Ethernet frames. This VLAN priority is called outbound VLAN priority. An outbound VLAN priority could be assigned to a regenerated priority. Therefore a mapping between the regenerated priority (see EthSwtPortOutboundVlanPriorityAssignmentRegeneratedPriority) to an outbound VLAN priority (see EthSwtPortOutboundVlanPriorityAssignmentOutboundVlanPriority) could be configured. The outbound priority is configured at an ingress port.

[SWS_EthSwt_00459] ⌈If EthSwtPortOutboundVlanPriorityAssignment is configured at an ingress port and Ethernet frames arrive at this ingress port, then an outbound VLAN priority according to the regenerated VLAN priority shall be assigned according to the EthSwtPortOutboundVlanPriorityAssignment, EthSwtPriorityRegeneratedIngressVlanPriority, and EthSwtPriorityOutboundVlanPriority configuration.⌋*()*

The outbound VLAN priority is used within the VLAN-tag as VLAN priority for an Ethernet frame which is transmitted on the network. The outbound VLAN priority is added to the Ethernet frame meta information. Therefore, the outbound VLAN priority does not impact the internal used priority of a particular Ethernet frame and vice versa. As soon as an Ethernet frame is transmitted with a VLAN-tag and this Ethernet frame has an assigned outbound VLAN priority, then this outbound VLAN priority is used as VLAN priority in the VLAN-tag.

[SWS_EthSwt_00460]{DRAFT} ⌈If an Ethernet frame has to be transmitted on a particular egress port, this Ethernet frame is transmitted with a VLAN-tag and an outbound VLAN priority is assgined to this Ethernet frame, then this Ethernet frame shall be transmitted with the VLAN priority set to the assigned outbound VLAN priority.⌋ *(SRS_Eth_00121)*

7.1.5.2.5 Frame filtering

If an Ethernet frame passes the "ingress filtering", then a so-called "frame filtering" is performed by an Ethernet switch. This processing stage has the focus on the destination MAC address of the received Ethernet frame. The Ethernet switch check if this destination MAC address is available in the internal address resolution table (ARL). Please refer to chapter subsubsection 7.1.5.1 for further information regarding the process to setup the ARL table.

[SWS_EthSwt_00461]{DRAFT} ⌈If the destination MAC address of a received Ethernet frame is qualified as a MAC unicast/multicast address and this MAC address is available in the ARL table, then this Ethernet frame shall be forwarded for further processing. Otherwise this Ethernet frame shall be dropped and the forwarding process shall be aborted.⌋*(SRS_Eth_00121)*

Note: If the destination MAC address of a received Ethernet frame is qualified as a MAC broadcast address, then this Ethernet frame is forwarded for further processing. No check in ARL table is performed.

If the Ethernet frame is forwarded, then the Ethernet frame is assigned to the corresponding egress port(s) according the ARL table entries. If the frame is forwarded and MAC address learning is enabled, the source MAC address is added into the ARL table.

7.1.5.2.6 Egress filtering

If an Ethernet frame pass the "frame filtering" than a so-called "egress filtering" is performed by an Ethernet switch. This processing stage has the focus on the VLAN membership. The previous processing stage "frame filtering" assigned this Ethernet to one or multiple egress ports. The egress filtering inspect the VLAN membership of the egress ports where the received Ethernet frame has been assigned to and the VLAN-ID of the received Ethernet frame. The egress filtering process keep the Ethernet frame assignment to those egress ports where the VLAN-ID of the received Ethernet frame and the VLAN membership of the egress port match. Otherwise the assignment of the Ethernet frame to a egress port is removed.

[SWS_EthSwt_00462]{DRAFT} ⌈If the VLAN membership of a egress port match to the VLAN-ID of a received Ethernet frame and this Ethernet frame has been assigned to this egress port, then the assignment of this Ethernet frame to this egress port shall be kept. Otherwise the assignment of this Ethernet frame to the affected egress port shall be removed.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_00463]{DRAFT} ⌈After the egress filtering has been finalized for an received Ethernet frame and the Ethernet frame is still assigned to a least one egress port, then this Ethernet frame shall be forwarded for further processing. Otherwise this Ethernet frame shall be dropped and the forwarding process shall be aborted.⌋*(SRS_Eth_00121)*

7.1.5.2.7 Per-stream filtering and policing

If an Ethernet frame pass the "egress filtering" than a so-called "per stream filtering and policing" could be performed by an Ethernet switch, if this processing stage is configured. Otherwise the Ethernet switch forward the Ethernet frame to the next procession stage "transmission selection algorithm".

[9, IEEE Std 802.1Q] defines per-stream filtering and policing. Per stream filtering and policing could be configured with `EthSwtPSFP`. The neighboring configuration container `EthSwtAtsInstanceTable` represent a table of so-called "aynchronous traffic shapers", which could be used by `EthSwtPSFP`. Both `EthSwtPSFP` and `EthSwtAtsInstanceTable` reside below the superordinated `EthSwtPSCM` (per stream classification and metering).

The EthSwtPSFP container include the following tables:

- EthSwtFilterMaxSduSizeTable, if configured at least one EthSwtFilterMaxSduSizeEntry exists
- EthSwtFlowMeteringTable, if configured at least one EthSwtFlowMeteringEntry exists
- EthSwtStreamFilterTable, if configured at least one EthSwtStreamFilterEntry exists
- EthSwtStreamGateTable, if configured at least one EthSwtStreamGateEntry exists

If the Ethernet switch HW supports this feature, then it can be configured by using the tables of EthSwtPSFP and EthSwtAtsInstanceTable.

The EthSwtStreamFilterTable represents the core table, because an entry of the EthSwtStreamFilterTable could reference one entry from the EthSwtFilterMaxSduSizeTable, EthSwtFlowMeteringTable and EthSwtStreamGateTable.

The EthSwtStreamFilterTable could have multiple EthSwtStreamFilterEntrys, where each entry represents a stream filter. EthSwtStreamFilterEntrys are configured as an ordered list. The position within the ordered list is defined with the configured value of EthSwtStreamFilterEntryPosition. The list is processed in ascending order by the Ethernet switch. As soon as the first EthSwtStreamFilterEntry matches, the Ethernet switch will threat this Ethernet frame according to the configuration that is associated with this EthSwtStreamFilterEntry. Subsequential EthSwtStreamFilterEntrys of the odered list will not be applied. In case a received Ethernet frame does not match any of the EthSwtStreamFilterEntrys, the Ethernet frame will be forwarded to the next frame processing stage.

[SWS_EthSwt_CONSTR_00602]{DRAFT} ⌈Every EthSwtStreamFilterEntry shall have a unique position value configured via EthSwtStreamFilterEntryPosition. The value shall start with 0 and continue in ascending order with no gaps for each subsequential EthSwtStreamFilterEntry.⌋*(SRS_Eth_00114)*

Note: The position value forms a ordered list of EthSwtStreamFilterEntrys

[SWS_EthSwt_00602]{DRAFT} ⌈If a Ethernet frame (i.e stream) arrives at processing stage per-stream filtering and policing, then the Ethernet switch shall check for a matching EthSwtStreamFilterEntry in ascending order according the EthSwtStreamFilterEntryPosition, starting with EthSwtStreamFilterEntryPosition configured with value 0.⌋*(SRS_Eth_00114)*

[SWS_EthSwt_00604]{DRAFT} ⌈If an arrived Ethernet frame (i.e stream) does not match any EthSwtStreamFilterEntrys, the Ethernet frame shall be forwarded to the next frame processing stage without applying any further stream filter handlings.⌋*(SRS_Eth_00114)*

An EthSwtStreamFilterEntry consist of EthSwtAssignedStreamHandle , EthSwtStreamFilterPriority, EthSwtFilterMaxSduSizeRef and additionally the optional references to the neighboring tables:EthSwtAssignedStreamHandle, EthSwtFlowMeteringEntryRef and EthSwtStreamGateEntryRef

A match of Ethernet frame to an stream filter is identified by considering EthSwtAssignedStreamHandle and EthSwtStreamFilterPriority

[SWS_EthSwt_00605]{DRAFT} ⌈If an Ethernet frame (i.e. stream) arrives at processing stage per-stream-filtering-and-policing, and this Ethernet Frame carries an Ethernet frame meta information which contains a stream handle id (see EthSwtStreamHandleAssignment) assigned by the stream-identification processing stage, then the Ethernet switch shall scan the EthSwtStreamFilterTable (with respect to [SWS_EthSwt_00602]) for an EthSwtStreamFilterEntry where its EthSwtAssignedStreamHandle and EthSwtStreamFilterPriority match to the stream handle id and priority of the arrived Ethernet frame (see EthSwtStreamHandleAssignment and EthSwtPortPriorityRegenerationRegeneratedPriority). Ohterwise the per-stream-filtering-and-policing process shall be aborted for this Ethernet frame and the Ethernet frame shall be forwarded to the next processing stage.⌋ *(SRS_Eth_00114)*

If an Ethernet switch detect an Ethernet frame which match to an EthSwtStreamFilterEntry, then the size of the Ethernet frame will be evaluated by considering the maximal acceptable Ethernet frame size referenced by the EthSwtStreamFilterEntry via EthSwtFilterMaxSduSizeRef.

[SWS_EthSwt_CONSTR_00603]{DRAFT} ⌈All EthSwtStreamFilterEntrys shall have a reference to a value of max-sdu-size via EthSwtFilterMaxSduSizeRef configured.⌋*(SRS_Eth_00114)*

Note: The definition of EthSwtFilterMaxSduSizeEntry includes the size of Preamble, SFD and minimum IPG (see subsection 10.1.13)

Implementation hint: An Ethernet switch hardware does not need (and probably does not) consider the length of Preamble, SFD and minimum IPG in its native filtering mechanism. In general, these three elements can be considered as known constants in an engineered Ethernet network by the Ethernet switch engine, such that an Ethernet switch can easily be configured to behave according to the definition of EthSwtFilterMaxSduSizeEntry

[SWS_EthSwt_00606]{DRAFT} ⌈If an Ethernet switch detect a match of an Ethernet frame to an EthSwtStreamFilterEntry according to [SWS_EthSwt_00605], then the Ethernet switch shall evaluate, if the frame size of this Ethernet frame (i.e. stream) exceeds the value of the EthSwtFilterMaxSduSizeEntry referenced by the EthSwtStreamFilterEntry via EthSwtFilterMaxSduSizeRef:

- If the frame size of the Ethernet frame exceeds the referenced max-sdu-size and then the per-stream-filtering-and-policing process shall be aborted for this Ethernet frame and the Ethernet frame shall be dropped.

- If the frame size of the Ethernet frame is equal or smaller than the referenced max-sdu-size, then the per-stream-filtering-and-policing processing shall continue.

⌋*(SRS_Eth_00114)*

[SWS_EthSwt_00607]{DRAFT} ⌈If the evaluation of the Ethernet frame size result to continue with the per-stream-filtering-and-policing processing according to [SWS_EthSwt_00606] and the EthSwtStreamFilterEntry reference a EthSwtStreamGateEntry, then the Ethernet switch shall assign the configured internal priority value (see EthSwtStreamGateIPV) by updating the Ethernet frame meta information of this Ethernet frame⌋*(SRS_Eth_00114)*

Note: The internal priority value is used for the traffic class assignment (see subparagraph 7.1.5.2.4.4)The internal prioity value is modeled as an 32bit value, but only the least 3 signification bits are considered. Thus, the internal priority value directly matches to the VLAN priority of an received Ethernet frame and an traffic class.

The state of a gate could be open or close. If a gate is open, then Ethernet frames could pass through for further processing. Otherwise a gate is closed and Ethernet frames are not permitted to pass through. Please note, AUTOSAR supports only open gates.

[SWS_EthSwt_CONSTR_00489]{DRAFT} ⌈If a EthSwtPSFP is configured, then the corresponding Ethernet switch hardware shall be configured such that Ethernet frames (i.e. streams) could always pass through (open gate).⌋*(SRS_Eth_00114)*

[SWS_EthSwt_00608]{DRAFT} ⌈If the evaluation of the Ethernet frame size result to continue with the per-stream-filtering-and-policing processing according to [SWS_EthSwt_00606], then the Ethernet switch shall perform the actions in dependency of the affected EthSwtStreamFilterEntry configuration in the following order:

1. If the EthSwtStreamFilterEntry reference a EthSwtStreamGateEntry, then the Ethernet switch shall assign the conifgured internal priority value (see EthSwtStreamGateIPV) to the Ethernet frame by updating its Ethernet frame meta information.
2. If the EthSwtStreamFilterEntry reference a EthSwtFlowMeteringEntry, then the Ethernet switch shall apply the flow metering configuration on the Ethernet frame.
3. If the EthSwtStreamFilterEntry reference a EthSwtAtsInstanceEntry, then the Ethernet switch shall apply the asynchronous traffic shaper configuration on the Ethernet frame.

⌋*(SRS_Eth_00114)*

A EthSwtPSFP has the possibility to define a flow metering by configuring a (EthSwtFlowMeteringTable). The table contain one or multiple EthSwtFlowMeteringEntrys. Each EthSwtFlowMeteringEntry represents a configuration of one flow metering. One EthSwtStreamFilterEntry could reference excatly one EthSwtFlowMeteringEntry. The configuration of the flow metering support to limit the rate of Ethernet frames (i.e. streams).

[SWS_EthSwt_00491]{DRAFT} ⌈If an Ethernet frame match to EthSwtStreamIdentificationEntry, this Ethernet frame pass the filtering and a EthSwtFlowMeteringEntry is available, then this Ethernet frame shall be handled by this EthSwtFlowMeteringEntry.⌋*(SRS_Eth_00114)*

[SWS_EthSwt_00492]{DRAFT} ⌈A configured EthSwtFlowMeteringEntry shall perform the metering according to the configuration: EthSwtFlowMeteringColorMode, EthSwtFlowMeteringCIR, EthSwtFlowMeteringCBS, EthSwtFlowMeteringEIR, EthSwtFlowMeteringEBS and EthSwtFlowMeterCF⌋*(SRS_Eth_00114)*

A EthSwtPSCM has the possibility to define asynchronous traffic shaping by configuring a (EthSwtAtsInstanceTable). The table contain one or multiple EthSwtAtsInstanceEntrys. Each EthSwtAtsInstanceEntry represents a configuration of one asynchronous traffic shaper. One EthSwtStreamFilterEntry could reference excatly one EthSwtAtsInstanceEntry. The configuration of an asynchronous traffic shapping support to shape Ethernet traffic according a so-called elgibility time .

[SWS_EthSwt_00493]{DRAFT} ⌈If asynchronous traffic shaping is configured EthSwtAtsInstanceEntry and applied on an Ethernet frame, then a elegibility time shall be assigned to this Ethernet frame by updating its Ethernet frame meta information.⌋*(SRS_Eth_00114)*

[SWS_EthSwt_00494]{DRAFT} ⌈A configured EthSwtAtsInstanceEntry shall perform the scheduling according to the following configuration: EthSwtPortATSCommitedBurstSize , EthSwtPortATSCommitedInformationRate and EthSwtAtsGroupMaximumResidenceTime.⌋*(SRS_Eth_00114)*

Note: EthSwtAtsGroupMaximumResidenceTime is available by the referenced EthSwtAtsGroupInstanceEntry (referenced via EthSwtPortATSSchedulerGroupRef) which is part of the EthSwtAtsGroupInstanceTable.

An EthSwtAtsGroupInstanceEntry represents one so-called "ATS Scheduler Group". All ATS instances (EthSwtAtsInstanceEntrys) which belonging to the same ATS Scheduler Group (referencing the same EthSwtAtsGroupInstanceEntry) use the same EthSwtAtsGroupMaximumResidenceTime. For an ATS Scheduler Group the eligibility assignment algorithm ensures, that Ethernet frames which have been received in a specific order will also be transmitted in that same order if they have been processed by any ATS instance belonging to that ATS scheduler group.

[SWS_EthSwt_00609]{DRAFT} ⌈If multiple EthSwtAtsInstanceEntrys reference the same EthSwtAtsGroupInstanceEntry and Ethernet frames are processed by those EthSwtAtsInstanceEntry, then the Ethernet switch elegibility assignment algorithm shall ensure, that processed Ethernet frames are transmitted in the same order as they have been arrived at the Ethernet switch⌋*(SRS_Eth_00114)*

7.1.5.2.8 Interaction with the Firewall module

The network packet inspection with per-stream filtering can be supported by the AUTOSAR firewall module to perform more advanced inspection techniques like stateful packet inspection and deep packet inspection. This chapter describes the functionality required by the Firewall module in the EthSwtDrv to perform efficient network packet inspection and filtering supported by per-stream filtering. More details about the interaction between the firewall module and per-stream filtering can be found in CP_SWS_Firewall.

Extraction of StreamHandleIdentifier from network packet

When a network packet is passed to the firewall for inspection, it has already passed the inspection by per-stream filtering. The firewall needs to know the per-stream filtering filter rule that allows the network packet to pass. This filter rule is modeled by the StreamHandleIdentifier in the EthSwtDrv. The value of the StreamHandleIdentifier can be added by the switch core to the network packet by modifying the network packet header and adding metadata to the network packet. This metadata is however not standardized and depends on the switch vendor.

[SWS_EthSwt_00500]{DRAFT} ⌈When EthSwt_ExtractStreamHandleIdx is called, the EthSwtDrv shall extract the StreamHandleIdentifier from the passed network packet, write the value to the StreamHandleIdxPtr and return E_OK.⌋*()*

Read out of StreamHandleIdentifier counting statistics

Many switches support counting statistics of the per-stream filtering filter rules, i.e., they count how often the filter rules provided matches to network packets. The firewall module requires this statistics information to raise Security Events (SEvs) to the IdsM. The counting statistics is typically implemented in terms of buckets, where a bucket counts the number of matches for multiple filter rules (see Figure 7.9)

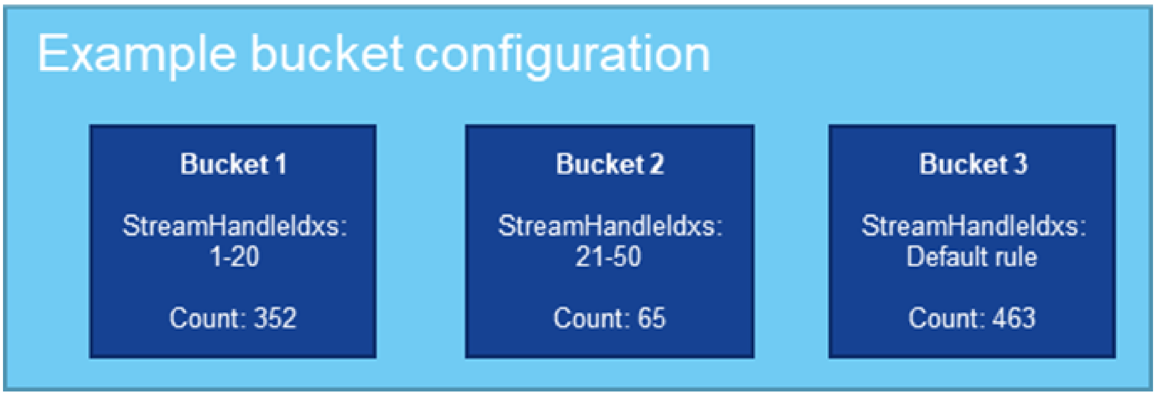


Figure 7.9: Example configuration of buckets counting matches of per-stream filtering filter rules

[SWS_EthSwt_00501]{DRAFT} ⌈When EthSwt_GetStreamHandleIdxStatistics is called, the EthSwtDrv shall read the count values for the buckets from the switch and return them to the EthIf by calling EthIf_StreamHandleIdxStatistics, where NumberOfBuckets is set to the number of buckets configured in the switch and the StreamHandleIdxStatisticsPtr points to the concatenated count values per bucket. Each count value has a length of 4 Byte.⌋*()*

Depending on the switch type, the bucket values are either reset upon read-out or keeping their count values. The EthSwtDrv shall provide a uniform handling of the count values independent of the switch type to ensure correct handling of the count values in the firewall module

[SWS_EthSwt_00502]{DRAFT} ⌈The EthSwtDrv shall provide monotonically increasing count values via EthIf_StreamHandleIdxStatistics starting with 0 upon boot up. If the count values are reset by the switch, the EthSwtDrv shall buffer the last read out count values and add them to the count values that were retrieved from the switch.⌋*()*

Runtime management of per-stream filtering filter rules

The firewall module can be switched by the BswM into different states, i.e., it can be switched to enable a different set of firewall rules to be active. This use-case also extends to the per-stream filtering filter rules, which should follow the state of the firewall module to ensure coherent packet filtering.

[SWS_EthSwt_00503]{DRAFT} ⌈When EthSwt_SetStreamHandleIdxConfiguration is called, the EthSwtDrv shall set the filter rule of the StreamHandleIdentifier identified by the StreamHandleIdx to

- Active, if StreamHandleIdxActivityStatus is set to TRUE
- Passive, if StreamHandleIdxActivityStatus is set to FALSE

⌋*()*

[SWS_EthSwt_00504]{DRAFT} ⌈After successfully setting the filter rule active/passive according to [SWS_EthSwt_00503], the EthSwtDrv shall call EthIf_StreamHandleIdxConfiguration for this StreamHandleIdx with StreamHandleIdxActivityStatus set to the current activity status of this filter rule.⌋*()*

7.1.5.2.9 Transmission selection algorithm

If an Ethernet frame pass all previous processing stages of the forwarding process, then this Ethernet frame has been assigned to a traffic class at ingress side by the processing stage "ingress filtering". Additionally, this Ethernet frame has also been assigned to one or multiple egress ports by processing the stages "frame filtering" and "egress filtering". As result, the forwarding process is adding the Ethernet frame to the queue according the traffic class assignment of each egress port where this Ethernet frame has been assigned. Figure 7.10 shows the linkage between the priority of an received Ethernet frame and the according egress queue via the traffic class assignment.

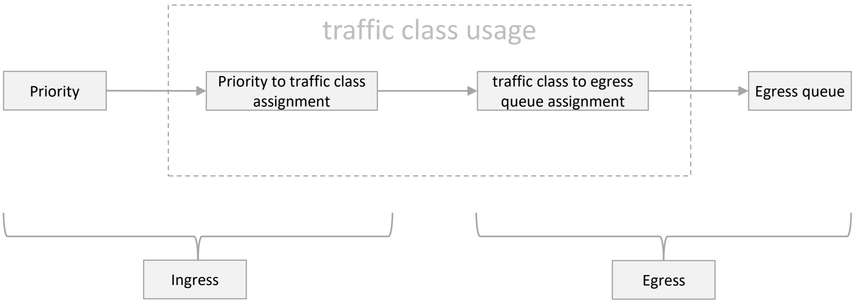

traffic class usage
Priority
Priority to traffic class assignment
traffic class to egress queue assignment
Egress queue
Ingress
Egress

Figure 7.10: Traffic class assignment as linkage between the priority at ingress side and a queue at egress side

Each egress port supports up to 8 queues. Each traffic class is mapped to excatly one queue. Figure 7.11 shows the linkage between the priority of an received Ethernet frame and the according egress queue via the traffic class assignment.

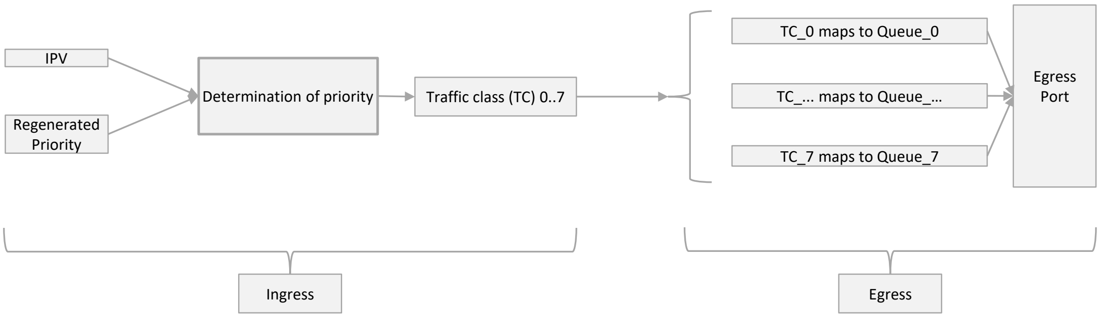


Figure 7.11: Mapping between the priority of an received Ethernet frame and the egress port queues

May be a Ethernet switch hardware cannot provide 8 queues per egress port. In such a case only a subset of traffic classes should be used. Starting from traffic class 0 in consecutive and ascending order. Table 7.4 shows an example of traffic class to egress port queue mapping, if only 4 queues are provided by an Ethernet switch hardware.

Traffic Class	Queue (if 4 Queues available)
0	0
1	1
2	2
3	3

Table 7.4: In this table,"Queue" to "Traffic class" at an egress port is depict

[SWS_EthSwt_00133]{OBSOLETE} ⌈Replaced by [SWS_EthSwt_CONSTR_00495]. The Ethernet switch configuration shall support to configure the linkage between the priority of an received Ethernet frame and the according queue of an egress port via the traffic class assignment. Therefore the priority to traffic class assignment at an ingress port (exclusively either via `EthSwtPortTrafficClassAssignment` or `EthSwtPortPriorityTrafficClassAssignment`) and the traffic class to a queue assignment at the egress port (via `EthSwtPortQueueTrafficClassAssignment`) shall be configured.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_CONSTR_00495]{DRAFT} ⌈The Ethernet switch configuration shall support to configure the linkage between the priority of an received Ethernet frame and the according queue of an egress port via the traffic class assignment. Therefore the priority to traffic class assignment at an ingress port (exclusively either via EthSwtPortTrafficClassAssignment or EthSwtPortPriorityTrafficClassAssignment) and the traffic class to a queue assignment at the egress port (via EthSwtPortQueueTrafficClassAssignment) shall be configured.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_00234] ⌈The Parameter EthSwtPortQueueMinimumLength shall define the minimum length for one queue of an dedicated egress port.⌋*(SRS_Eth_00121)*

Note: The actual queue length can be longer. The decision on the queue length is most likely to be taken by the Ethernet switch hardware or fixed by the Ethernet switch design. The definition of the minimum queue length in the configuration is supposed to guarantee that some priorities have enough egress buffer.

Each egress queue (see EthSwtPortQueue) has to configure the algorithm to select the Ethernet frames for transmission. Therefore each egress queue has an mandatory sub containter EthSwtPortEgressQueueTransmissionSelection. EthSwtPortEgressQueueTransmissionSelection defines the selection algorithm via EthSwtPortEgressQueueTransmissionSelectionAlgorithm (e.g. credit based shaper, asynchronous traffic shaper ... a.s.o.). Each EthSwtPortQueue is connected to an port scheduler. The port scheduler has to schedule all connected egress queues. Each port scheduler has an mandatory sub container EthSwtPortEgressScheduler which defines the scheduler algorithm via EthSwtPortSchedulerAlgorithm (e.g. strict priority, weighted round robin ... a.s.o.). Multiple egress schedulers at the same egress port could be configured and connected in an cascaded manner. Thus, the output of an egress scheduler is used as an input for the sub sequential egress port scheduler. Figure 7.12 shows examples for an egress port structure.

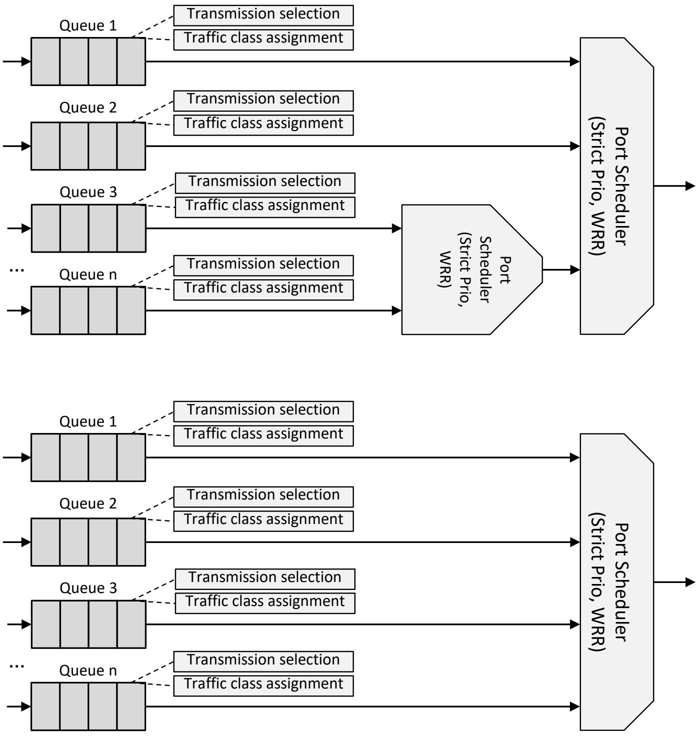


Figure 7.12: Examples for an egress port structure

The port scheduler algorithm schedule its input (either an egress queue or an egress port scheduler) by considering the according properties (e.g. traffic class assignment). Once the port scheduler algorithm has decided which of its input should be handled, the port scheduler select an Ethernet frame from the according egress queue based on the configured transmission selection algorithm:

- If the transmission selection alogrithm is configured as credit based shaper, then the according egress queue is handled as FIFO. The egress queue has an budget of credits, which is increased in the idle phase and decreased for each transmission of Ethernet frame from this egress queue.
- If the transmission selection alogrithm is configured as asynchronous traffic shaper, then the according egress queue is handled as queue. Each Ethernet frame of the queue has an assigned eligibility time. According the eligibility time a Ethernet frame is selected from this egress queue. The Ethernet frames are not handled according the arrival in this egress queue, but according the assigned eligibility time which has been added at the ingress side
- If the transmission selection alogrithm is configured as unshaped, then the according egress queue is handled as FIFO
- If the transmission selection alogrithm is configured as enhanced traffic shaping, then the according egress queue is handled is handled as queue

Note: The parameterization of the egress port influences the latency of Ethernet frames within the network.

The configuration of the egress port schedulers is done with the container EthSwtPortEgressScheduler and its sub-container EthSwtPortEgressSchedulerPredecessor with multiplicity 1 to *. Egress port scheduler connect its predecessors with the predecessor references EthSwtPortEgressPredecessorRef. An egress port scheduler could either have an further egress port scheduler or a egress port queue as predecessor.

[SWS_EthSwt_00132]{OBSOLETE} ⌈Replaced by[SWS_EthSwt_00613]. The configuration of the Ethernet switch driver shall support different egress port structures by the configuration EthSwtPortEgressScheduler.⌋*(SRS_Eth_00121)*

[SWS_EthSwt_00613]{DRAFT} ⌈If an Ethernet frame is added to an EthSwtPortQueue, then the Ethernet switch shall handle this Ethernet frame according the configured transmission selection algorithm (EthSwtPortEgressQueueTransmissionSelection) of this EthSwtPortQueue and with respect to the configured egress port structure (EthSwtPortEgressScheduler, EthSwtPortEgressSchedulerPredecessor) of the corresponding egress port (EthSwtPortEgress)⌋*(SRS_Eth_00121, SRS_Eth_00179, SRS_Eth_00180)*

7.1.5.2.10 Transmission on the network

An Ethernet frame which is selected by the very last EthSwtPortEgressScheduler, will be transfered from the egress queue to the Ethernet network. As preparation for the transmission a so-called "Priority Code Point Encoding" and "VLAN forwarding" is performed. This is the last procession stage in the frame forwarding process. This processing stage ensure that all information of the Ethernet frame meta information are written in the Ethernet frame before the frame is forwarded to an PHY. The Ethernet frame meta information contain the latest state of the Ethernet frame from the path through the Ethernet switch. An Example for the content of an Ethernet frame meta information:

- EthSwtPortOutboundVlanPriorityAssignmentOutboundVlanPriority: the Ethernet frame will be transmitted with this priority in the VLAN-tag
- EthSwtVlanForwardingType set to ETHSWT_SENT_TAGGED: the Etherent frame will be transmitted with an VLAN-tag

7.1.5.3 Switch Management support

Switch Management enables the possibility to control an Ethernet frame regarding a Switch-Port specific ingress and egress handling as well as providing a Switch-Port specific timestamp. This functionality is essential for other BSW modules, in particular for EthTSyn, which requires Port specific information associated to a time synchronization or path-delay measurement frame.

For an introduction of the basic HW architecture and interaction, please refer to [4, SWS_EthernetDriver].

[SWS_EthSwt_00240] ⌈The Switch driver shall offer Switch management APIs

- EthSwt_EthRxProcessFrame
- EthSwt_EthRxFinishedIndication
- EthSwt_EthTxAdaptBufferLength
- EthSwt_EthTxPrepareFrame
- EthSwt_SetMgmtInfo
- EthSwt_EthTxProcessFrame and
- EthSwt_EthTxFinishedIndication

if EthSwtManagementSupportApi is set to TRUE.⌋*(SRS_BSW_00171, SRS_Eth_00125)*

Note: Switch management APIs support the EthIf to gather / modify Switch-Port specific communication attributes.

[SWS_EthSwt_00241] ⌈The Switch Driver management APIs

- EthSwt_EthRxProcessFrame
- EthSwt_EthRxFinishedIndication
- EthSwt_EthTxAdaptBufferLength
- EthSwt_EthTxPrepareFrame
- EthSwt_SetMgmtInfo
- EthSwt_EthTxProcessFrame and
- EthSwt_EthTxFinishedIndication

shall support the Ethernet Driver to gather the Switch specific management information out of an Ethernet frame for reception or to prepare an Ethernet frame for management mode conformant frame transmission, e.g. the egress route of a frame.⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00242] ⌈The Switch Driver management APIs EthSwt_EthTxProcessFrame and EthSwt_EthTxFinishedIndication shall return immediately, if EthSwt_SetMgmtInfo has not been called before a call of EthSwt_EthTxProcessFrame.⌋*(SRS_Eth_00125)*

7.1.5.4 Global Time support

For more details regarding time measurement with Switches, please refer to [15, SWS_TimeSyncOverEthernet].

[SWS_EthSwt_00243] ⌈The Switch driver shall access the port specific hardware time stamps if EthSwtPortTimeStampSupport of the port is set to TRUE.⌋*(SRS_BSW_00171, SRS_Eth_00125)*

[SWS_EthSwt_00378] ⌈If EthSwt_PortEnableTimeStamp is called for a PortIdx, the switch driver shall enable the time-stamping for this port if EthSwtPortTimeStampSupport is set to TRUE for this port.⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00245] ⌈The Switch driver shall inform the EthIf about the availability of port specific ingress and egress timestamps using the APIs EthIf_SwitchIngressTimeStampIndication and EthIf_SwitchEgressTimeStampIndication , if EthSwtGlobalTimeSupportApi is set to TRUE.⌋*(SRS_Eth_00125)*

Note: Global Time support typically requires the activation of the Switch management support functionality within the Switch device.

7.1.5.5 Counter synchronization of Ethernet switches which are connected via uplink ports

Some Ethernet Switches provide the possibility to synchronize their internal clock. For Ethernet switches which are connected via uplink ports it is not necessary to measure the delay between the connected uplink ports, if the clock synchronization clock is activated (EthSwtClockSynchronizationSupport set to TRUE).

[SWS_EthSwt_00408] ⌈The Switch driver shall enable clock synchronization with another Ethernet switch to which it is connected via uplink port, if EthSwtClockSynchronizationSupport is set to TRUE.⌋*()*

[SWS_EthSwt_CONSTR_00409] ⌈The port specific timestamping (EthSwtPortTimeStampSupport) can be set to TRUE, if clock synchronization for connected Ethernet switches is deactivated (EthSwtClockSynchronizationSupport set to FALSE).⌋ *()*

[SWS_EthSwt_CONSTR_00410] ⌈The port specific timestamping (EthSwtPortTimeStampSupport) can be set to TRUE, if EthSwtClockSynchronizationSupport is activated and EthSwtPortRole is not ETHSWT_UP_LINK_PORT. EthSwtPorts with EthSwtPortRole ETHSWT_UP_LINK_PORT are connected to another Ethernet switch and not considered for the time delay compensation, if EthSwtClockSynchronizationSupport is activated.⌋*()*

7.1.5.6 Verification of Configuration

There are some situations where the Host controller needs to verify the Switch configuration.

[SWS_EthSwt_00292] ⌈If the parameter EthSwtVerifyConfigApi is set to TRUE the function EthSwt_VerifyConfig shall be used to verify switch configuration.⌋ *(SRS_Eth_00126)*

Implementation hint: As Switch configuration is highly HW-Architecture dependent the steps inside the function are implementation specific.

In some use cases, it is necessary to stop frame forwarding during the verification using the optional function EthSwt_SetForwardingMode

The function EthSwt_VerifyConfig could for example do the following steps:

- Stop frame forwarding by calling EthSwt_SetForwardingMode(FALSE).
- Verify the switch configuration
- In case the switch configuration is valid then frame forwarding shall be enabled by calling EthSwt_SetForwardingMode(TRUE)(if disabled in step 1).
- In case the switch configuration is not valid then the switch shall be reset and reconfigured.

Note: Please note that a reset of the Host Controller does not necessarily need a reset of the connected Switch HW. This needs to be evaluated individually very carefully as a reset raises the risk of uncontrolled communication during reset phase of the host controller.

Note: The Verification of the Switch Configuration as described above is just an example how and when this Verification may be done. It is very dependent on the used switch HW as well as the individual HW-Architecture and even Power supply and Reset strategy of the Switch of the ECU how the Configuration is verified or even how it can be verified. The only thing what this Module specifies is the interface to the upper layer to apply some verification on the switch configuration.

7.1.5.7 Testing and Diagnostic of Switch Ports

If configured, the Ethernet Switch Driver provides following interfaces to apply Testing and diagnostic functionalities

- EthSwt_GetPortSignalQuality
- EthSwt_GetPortIdentifier
- EthSwt_GetSwitchIdentifier
- EthSwt_WritePortMirrorConfiguration

- EthSwt_ReadPortMirrorConfiguration
- EthSwt_GetPortMirrorState
- EthSwt_SetPortMirrorState
- EthSwt_SetPortTestMode
- EthSwt_SetPortLoopbackMode
- EthSwt_SetPortTxMode
- EthSwt_GetPortCableDiagnosticsResult
- EthSwt_GetCfgDataRaw
- EthSwt_GetCfgDataInfo

The Availabiliy of these functions is strongly depending on the possibilities of the used Transceiver-(Phy)-HW.

7.1.5.8 Low Power Mode Support

[SWS_EthSwt_00376] ⌈If EthSwtLowPowerModeSupport is set to TRUE and at least one EthSwtPort of a Ethernet switch is enabled and the corresponding Ethernet switch HW is in an inactive or low power mode the Ethernet switch HW shall be set to an active mode in which forwarding of Ethernet frames is possible.⌋*()*

[SWS_EthSwt_00377] ⌈If EthSwtLowPowerModeSupport is set to TRUE and no EthSwtPort for a certain Ethernet switch is enabled, the corresponding Ethernet switch HW shall be set to an inactive or low power mode.⌋*()*

7.2 Error Classifications

Section 7.2 "Error Handling" of the document [10, SWS_BSW General] describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below

7.2.1 Development Errors

[SWS_EthSwt_00001] Definiton of development errors in module EthSwt ⌈

Type of error	*Related error code*	*Error value*
Invalid switch index	ETHSWT_E_INV_SWITCH_IDX	0x01
EthSwt module was not initialized	ETHSWT_E_UNINIT	0x02
Invalid pointer in parameter list	ETHSWT_E_PARAM_POINTER	0x03
Invalid API which is not available by another module	ETHSWT_E_INV_API	0x05
Invalid switch port index	ETHSWT_E_INV_SWITCHPORT_IDX	0x06
Invalid Controller Index	ETHSWT_E_INV_CTRL_IDX	0x07
Invalid input parameter	ETHSWT_E_INV_PARAM	0x08
Invalid configuration	ETHSWT_E_INIT_FAILED	0x09

⌋*(SRS_BSW_00385)*

[SWS_EthSwt_00009] ⌈If development error detection is enabled, the function EthSwt_Init shall check the parameter CfgPtr for being valid. If the check fails, EthSwt_Init shall raise the development error ETHSWT_E_INIT_FAILED.⌋*(SRS_BSW_00323, SRS_BSW_00369)*

Note: Please note that in case of variant pre-compile NULL_PTR is allowed.

[SWS_EthSwt_00164] ⌈The switch driver shall check whether the lower layer driver, i.e. the EthTrcv provides the APIs which can be called by an upper layer module (EthIf) of the switch driver and will be forwarded to the lower layer. In case of missing APIs, the switch driver shall raise the development error ETHSWT_E_INV_API if APIs are missing in the lower layer module.⌋*(SRS_BSW_00369, SRS_BSW_00386, SRS_Eth_00118)*

Note: This check will be performed upon calling a certain API. For this check the input parameter SwitchPortIdx and a configuration table which needs to be derived from the configuration of the Ethernet transceiver drivers which are attached to the Ethernet switch driver are necessary. This functionality is necessary if development error tracing is activated. This check is necessary because an Ethernet switch driver API can be called by an upper layer module with the argument SwitchPortIdx. This value of this SwitchPortIdx can be in a valid range, but some Ethernet transceiver driver which are used by the switch driver support the API and some do not support this API. In order to resolve this conflict, this check has been implemented.

[SWS_EthSwt_00156] ⌈The function EthSwt_SetSwitchPortMode shall check whether the EthTrcv_SetTransceiverMode API of the indexed transceiver driver is available by checking whether for this SwitchPortIdx the corresponding EthTrcv API is available. If this is not the case, the function shall raise the development error ETHSWT_E_INV_API.⌋*(SRS_BSW_00413, SRS_BSW_00323, SRS_BSW_00369, SRS_Eth_00118)*

[SWS_EthSwt_00157] ⌈The function EthSwt_GetSwitchPortMode shall check whether the EthTrcv_GetTransceiverMode API of the indexed transceiver driver is available by checking whether for this SwitchPortIdx the corresponding EthTrcv

API is available. If this is not the case, the function shall raise the development error ETHSWT_E_INV_API.⌋*(SRS_BSW_00413, SRS_BSW_00323, SRS_BSW_00369, SRS_Eth_00118)*

[SWS_EthSwt_00386] ⌈If development error detection is activated by `EthSwtDevErrorDetect`, all functions except `EthSwt_Init` shall check that the service `EthSwt_Init` was previously called. If the check fails, the function shall raise the development error ETHSWT_E_UNINIT.⌋*(SRS_BSW_00350)*

[SWS_EthSwt_00387] ⌈If development error detection is activated by `EthSwtDevErrorDetect`, all functions with input parameter SwitchIdx shall check the parameter for being valid. If the check fails, the functions shall raise the development error ETHSWT_E_INV_SWITCH_IDX.⌋*(SRS_BSW_00350)*

[SWS_EthSwt_00389] ⌈If development error detection is enabled, all functions with input parameter `SwitchPortIdx` or `PortIdx` shall check the parameter for being valid. If the check fails, the functions shall raise the development error ETHSWT_E_INV_SWITCH_IDX.⌋*(SRS_BSW_00350)*

[SWS_EthSwt_00390] ⌈If development error detection is enabled, all functions with input parameter CtrlIdx shall check the parameter for being valid. If the check fails, the functions shall raise the development error ETHSWT_E_INV_CTRL_IDX.⌋*(SRS_BSW_00350)*

[SWS_EthSwt_00391] ⌈If development error detection is enabled, all functions with input parameter `BufIdx` shall check the parameter for being valid. If the check fails, the functions shall raise the development error ETHSWT_E_INV_PARAM.⌋*(SRS_BSW_00350)*

[SWS_EthSwt_00392] ⌈If development error detection is enabled, all functions with inout or output pointer parameter shall check the parameter for being valid. If the check fails, the functions shall raise the development error ETHSWT_E_PARAM_POINTER.⌋*(SRS_BSW_00350)*

[SWS_EthSwt_00393] ⌈If development error tracing is activated by `EthSwtDevErrorDetect`, the functions which call an Ethernet Transceiver API and do not obtain the functionality directly from the switch port interface shall check whether the API of the indexed transceiver driver is available. If this is not the case, the functions shall raise the development error ETHSWT_E_INV_API.⌋*(SRS_BSW_00350)*

[SWS_EthSwt_00154] ⌈If development error detection is activated by `EthSwtDevErrorDetect`, the function `EthSwt_GetLinkState` shall check whether the `EthTrcv_GetLinkState` API of the indexed transceiver driver is available by checking whether for this `SwitchPortIdx` the corresponding `EthTrcv` API is available. If this is not the case, the function shall raise the development error ETHSWT_E_INV_API.⌋*(SRS_Eth_00118, SRS_Eth_00119, SRS_BSW_00413, SRS_BSW_00323, SRS_BSW_00369)*

7.2.2 Runtime Errors

[SWS_EthSwt_00434] Definiton of runtime errors in module EthSwt ⌈

Type of error	*Related error code*	*Error value*
Initialization of ports is not finished	ETHSWT_INIT_NOT_COMPLETED	0x01

⌋*()*

7.2.3 Transient Faults

There are no transient faults.

7.2.4 Production Errors

There are no production errors.

7.2.5 Extended Production Errors

[SWS_EthSwt_00113] ⌈

Error Name:	ETHSWT_E_ACCESS	
Short Description:	Ethernet Switch Access Failure	
Long Description:	This production error shall be issued when the switch is not accessible.	
Recommended DTC:	N/A	
Detection Criteria:	Fail	When access to the Ethernet Switch fails the module shall report the extended production error with event status DEM_EVENT_STATUS_PREFAILED to DEM.
	Pass	When access to the Ethernet Switch succeeds the module shall report the extended production error with event status DEM_EVENT_STATUS_PREPASSED to DEM.
Secondary Parameters:	N/A	
Time Required:	N/A	
Monitor Frequency	N/A	
MIL illumination:	N/A	

Table 7.5: ETHSWT_E_ACCESS

⌋*(SRS_BSW_00385)*

[SWS_EthSwt_00395] ⌈

Error Name:	ETHSWT_E_SYNCPORT2PHY	
Short Description:	Ethernet switch port and the referenced Ethernet transceiver are in contradicting modes.	
Long Description:	While getting the Ethernet switch port mode, the Ethernet switch driver detected an inconsistent state between Ethernet switch port and the referenced Ethernet transceiver Mode.	
Recommended DTC:	N/A	
Detection Criteria:	Fail	When getting the Ethernet switch port mode together with the Ethernet transceiver mode and the mode of the two referenced modules was found inconsistent the module shall report the extended production error with event status DEM_EVENT_STATUS_PREFAILED to DEM.
	Pass	When getting the Ethernet switch port mode together with the Ethernet transceiver mode and the mode of the two referenced modules was found consistent the module shall report the extended production error with event status DEM_EVENT_STATUS_PREPASSED to DEM.
Secondary Parameters:	N/A	
Time Required:	N/A	
Monitor Frequency	N/A	
MIL illumination:	N/A	

Table 7.6: ETHSWT_E_SYNCPORT2PHY

⌋*(SRS_BSW_00385)*

8 API specification

8.1 Imported types

This chapter lists all types included from the following files:

[SWS_EthSwt_00002] Definition of imported datatypes of module EthSwt ⌈

Module	***Header File***	***Imported Type***
ComStack_Types	ComStackTypes.h	TimeStampType (draft)
Dem	Rte_Dem_Type.h	Dem_EventIdType
	Rte_Dem_Type.h	Dem_EventStatusType
Eth	Eth_GeneralTypes.h	Eth_BufIdxType
	Eth_GeneralTypes.h	Eth_CounterType
	Eth_GeneralTypes.h	Eth_DataType
	Eth_GeneralTypes.h	Eth_MacVlanType
	Eth_GeneralTypes.h	Eth_ModeType
	Eth_GeneralTypes.h	Eth_RxStatsType
	Eth_GeneralTypes.h	Eth_TxErrorCounterValuesType
	Eth_GeneralTypes.h	Eth_TxStatsType
EthTrcv	Eth_GeneralTypes.h	EthTrcv_BaudRateType
	Eth_GeneralTypes.h	EthTrcv_CableDiagResultType
	Eth_GeneralTypes.h	EthTrcv_DuplexModeType
	Eth_GeneralTypes.h	EthTrcv_LinkStateType
	Eth_GeneralTypes.h	EthTrcv_PhyLoopbackModeType
	Eth_GeneralTypes.h	EthTrcv_PhyTestModeType
	Eth_GeneralTypes.h	EthTrcv_PhyTxModeType
	Eth_GeneralTypes.h	EthTrcv_WakeupReasonType
Mka	Mka.h	Mka_ConfidentialityOffsetType (DRAFT)
	Mka.h	Mka_MacSecConfigType (DRAFT)
	Mka.h	Mka_SakKeyPtrType (DRAFT)
	Mka.h	Mka_Stats_Rx_ScType (DRAFT)
	Mka.h	Mka_Stats_Rx_SecYType (DRAFT)
	Mka.h	Mka_Stats_SecYType (DRAFT)
	Mka.h	Mka_Stats_Tx_ScType (DRAFT)
	Mka.h	Mka_Stats_Tx_SecYType (DRAFT)
	Mka.h	Mka_ValidateFramesType (DRAFT)
NvM	Rte_NvM_Type.h	NvM_BlockIdType
	Rte_NvM_Type.h	NvM_BlockRequestType
	Rte_NvM_Type.h	NvM_RequestResultType
Spi	Spi.h	Spi_AsyncModeType
	Spi.h	Spi_ChannelType
	Spi.h	Spi_DataBufferType
	Spi.h	Spi_NumberOfDataType

▽

△

Module	*Header File*	*Imported Type*
	Spi.h	Spi_SequenceType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

⌋()

8.2 Type definitions

8.2.1 EthSwt_StateType

[SWS_EthSwt_00123] Definition of datatype EthSwt_StateType ⌈

Name	EthSwt_StateType		
Kind	Enumeration		
Range	ETHSWT_STATE_UNINIT	0x00	Switch is not yet configured
	ETHSWT_STATE_INIT	0x01	Switch driver is initialized
	ETHSWT_STATE_ PORTINIT_COMPLETED	0x02	Port initialization is completed
	ETHSWT_STATE_ACTIVE	0x03	Switch is active
Description	Status supervision used for Development Error Detection. The state shall be available for debugging.		
Available via	Eth_GeneralTypes.h		

⌋(*SRS_BSW_00406*)

8.2.2 EthSwt_ConfigType

[SWS_EthSwt_00165] Definition of datatype EthSwt_ConfigType ⌈

Name	EthSwt_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	–
Description	Implementation specific structure of the post build configuration.	
Available via	EthSwt.h	

⌋(*SRS_BSW_00395*)

8.2.3 EthSwt_MacLearningType

[SWS_EthSwt_00227] Definition of datatype EthSwt_MacLearningType ⌈

Name	EthSwt_MacLearningType		
Kind	Enumeration		
Range	ETHSWT_MACLEARNING_HWDISABLED	–	If hardware learning disabled, the switch must not learn new MAC addresses
	ETHSWT_MACLEARNING_HWENABLED	–	If hardware learning enabled, the switch learns new MAC addresses
	ETHSWT_MACLEARNING_SWENABLED	–	If software learning enabled, the hardware learning is disabled and the switch forwards packets with an unknown source address to a host CPU
Description	The interpretation of this value		
Available via	Eth_GeneralTypes.h		

⌋*(SRS_Eth_00087)*

8.2.4 EthSwt_MgmtInfoType

[SWS_EthSwt_91002] Definition of datatype EthSwt_MgmtInfoType ⌈

Name	EthSwt_MgmtInfoType	
Kind	Structure	
Elements	SwitchIdx	
	Type	uint8
	Comment	Switch index
	SwitchPortIdx	
	Type	uint8
	Comment	Port index of the switch
Description	Type for holding the management information received/transmitted on Switches (ports).	
Available via	Eth_GeneralTypes.h	

⌋*(SRS_Eth_00125)*

8.2.5 EthSwt_PortMirrorCfgType

[SWS_EthSwt_91017] Definition of datatype EthSwt_PortMirrorCfgType ⌈

Name	EthSwt_PortMirrorCfgType	
Kind	Structure	
Elements	srcMacAddrFilter	
	Type	Array of uint8
	Size	6

▽

△

	Comment	Specifies the source MAC address [0..255,0..255,0..255,0..255,0..255,0..255] that should be mirrored. If set to 0,0,0,0,0,0, no source MAC address filtering shall take place.
	dstMacAddrFilter	
	Type	Array of uint8
	Size	6
	Comment	Specifies the destination MAC address [0..255,0..255,0..255,0..255,0..255,0..255] that should be mirrored. If set to 0,0,0,0,0,0, no destination MAC address filtering shall take place.
	VlanIdFilter	
	Type	uint16
	Comment	Specifies the VLAN address 0..4094 that should be mirrored. If set to 65535, no VLAN filtering shall take place.
	MirroringPacketDivider	
	Type	uint8
	Comment	Divider if only a subset of received frames should be mirrored. E.g. MirroringPacketDivider = 2 means every second frames is mirrored
	MirroringMode	
	Type	uint8
	Comment	specifies the mode how the mirrored traffic should be tagged : 0x00 == No VLAN retagging; 0x01 == VLAN retagging; 0x02 == VLAN Double tagging
	TrafficDirectionIngressBitMask	
	Type	uint32
	Comment	Specifies the bit mask of Ethernet switch ingress port traffic direction to be mirrored. The bit mask is calculated depending of the values of EthSwtPortIdx. (e.g. set EthSwtPortIdx == 2 => TrafficDirectionIngressBitMask = 0b0000 0000 0000 0000 0000 0000 0000 0100). 0b0 == enable ingress port mirroring 0b1 == disable ingress port mirroring Example: TrafficDirectionIngressBitMask = 0b0000 0000 0000 0000 0000 0000 0000 0100 => Ingress traffic mirroring is enabled of Ethernet switch port with EthSwtPortIdx=2
	TrafficDirectionEgressBitMask	
	Type	uint32
	Comment	Specifies the bit mask of Ethernet switch egress port traffic direction to be mirrored. The bit mask is calculated depending of the values of EthSwtPortIdx. (e.g. set EthSwtPortIdx == 2 => TrafficDirectionEgressBitMask = 0b0000 0000 0000 0000 0000 0000 0000 0100). 0b0 == enable egress port mirroring 0b1 == disable egress port mirroring Example: TrafficDirectionEgressBitMask = 0b0000 0000 0000 0000 0000 0000 0000 0001 => Egress traffic mirroring is enabled of Ethernet switch port with EthSwtPortIdx=0
	CapturePortIdx	
	Type	uint8
	Comment	Specifies the Ethernet switch port which capture the mirrored traffic
	ReTaggingVlanId	
	Type	uint16
	Comment	Specifies the VLAN address 0..4094 which shall be used for re-tagging if MirroringMode is set to 0x01 (VLAN re-tagging). If the value is set to 65535, the value shall be ignored, because the VLAN address for re-tagging is provided by the Ethernet switch configuration
	DoubleTaggingVlanId	

▽

△

	Type	uint16
	Comment	Specifies the VLAN address 0..4094 which shall be used for double-tagging if MirroringMode is set to 0x02 (VLAN double tagging). If the value is set to 65535, the value shall be ignored, because the VLAN address for double tagging is provided by the Ethernet switch configuration
Description	The EthSwt_PortMirrorCfgType specify the port mirror configuration which is set up per Ethernet switch. The configuration is written to the Ethernet switch driver by calling EthSwt_WritePortMirror Configuration. One port mirror configuration is maintained per Ethernet Switch.	
Available via	Eth_GeneralTypes.h	

⌋*(SRS_Eth_00123)*

8.2.6 EthSwt_PortMirrorStateType

[SWS_EthSwt_91020] Definition of datatype EthSwt_PortMirrorStateType ⌈

Name	EthSwt_PortMirrorStateType		
Kind	Enumeration		
Range	PORT_MIRRORING_ DISABLED	0x00	port mirroring disabled
	PORT_MIRRORING_ ENABLED	0x01	port mirroring enabled
Description	Type to request or obtain the port mirroring state (enable/disable) for a particular port mirror configuration per Ethernet switch.		
Available via	Eth_GeneralTypes.h		

⌋*(SRS_Eth_00123)*

8.2.7 EthSwt_ReturnType

[SWS_EthSwt_91033] Definition of Std_ReturnType-extension for module EthSwt ⌈

Range	ETHSWT_PORT_ MIRRORING_ CONFIGURATION_NOT_ SUPPORTED	0x02	port mirroring configuration is not supported by Ethernet switch driver or by the Ethernet switch hardware
Description	Overlayed return value of Std_ReturnType for Ethernet switch driver API EthSwt_WritePortMirror Configuration, if the port mirroring configuration is not supported by Ethernet switch driver or by the Ethernet switch hardware (e.g. the configured mirrored traffic direction (see SWS_EthSwt_91017 "TrafficDirectionIngressBitMask" and "TrafficDirectionEgressBitMask") for ingress and egress traffic of the same port is not supported, or the addressed Ethernet switch ports within the port mirror configuration are not accessible by the Ethernet switch driver)		
Available via	Eth_GeneralTypes.h		

⌋*()*

8.2.8 EthSwt_MgmtOwner

[SWS_EthSwt_91035] Definition of datatype EthSwt_MgmtOwner ⌈

Name	EthSwt_MgmtOwner		
Kind	Enumeration		
Range	ETHSWT_MGMT_OBJ_UNUSED	0x00	Object unused
	ETHSWT_MGMT_OBJ_OWNED_BY_ETHSWT	0x01	Object used and EthSwt collects needed data
	ETHSWT_MGMT_OBJ_OWNED_BY_UPPER_LAYER	0x02	Object used and the upper layer does calculations
Description	Holds information if upper layer or EthSwt is owner of mgmt_obj.		
Available via	Eth_GeneralTypes.h		

⌋*()*

8.2.9 EthSwt_Mgmt_ObjectType

[SWS_EthSwt_91037] Definition of datatype EthSwt_MgmtObjectType ⌈

Name	EthSwt_MgmtObjectType	
Kind	Structure	
Elements	Validation	
	Type	EthSwt_MgmtObjectValidType
	Comment	The validation information for the mgmt_obj.
	IngressTimestamp	
	Type	TimeStampType
	Comment	The ingress timestamp value out of the switch.
	EgressTimestamp	
	Type	TimeStampType
	Comment	The egress timestamp value out of the switch.
	MgmtInfo	
	Type	EthSwt_MgmtInfoType
	Comment	Received/Transmitted Management information of the switches.
	Ownership	
	Type	EthSwt_MgmtOwner
	Comment	The ownership of MgmtObj.
Description	Provides information about all struct member elements. The ownership gives information whether EthSwt has finished its activities in providing all struct member elements.	
Available via	Eth_GeneralTypes.h	

⌋*()*

[SWS_EthSwt_00433] ⌈A MgmtObject is just allowed to be owned between EthSwt and only one <UPPER_LAYER>. The structure element can be identified unambiguously using the DataPtr in Rx- and BufIdx in Tx-context, because both elements are definitively unique within the RxIndication() / TxConfirmation() context.⌋*()*

8.2.10 EthSwt_MgmtObjectValidType

[SWS_EthSwt_91036] Definition of datatype EthSwt_MgmtObjectValidType ⌈

Name	EthSwt_MgmtObjectValidType	
Kind	Structure	
Elements	IngressTimestampValid	
	Type	Std_ReturnType
	Comment	IngressTimestampValid shall be set to E_NOT_OK if ingress timestamp is not available
	EgressTimestampValid	
	Type	Std_ReturnType
	Comment	EgressTimestampValid shall be set to E_NOT_OK if ingress timestamp is not available.
	MgmtInfoValid	
	Type	Std_ReturnType
	Comment	MgmtInfoValid shall be set to E_NOT_OK if ingress timestamp is not available(e.g. timeout).
Description	Will be set from EthSwt and marks EthSwt_MgmtObject as valid or not. So the upper layer will be able to detect inconsistencies.	
Available via	Eth_GeneralTypes.h	

⌋*()*

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 EthSwt_Init

[SWS_EthSwt_00006] Definition of API function EthSwt_Init ⌈

Service Name	EthSwt_Init	

Syntax	`void EthSwt_Init (`
	`  const EthSwt_ConfigType* CfgPtr`
	`)`	

Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CfgPtr	Points to the implementation specific structure
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Initializes the Ethernet Switch Driver	
Available via	EthSwt.h	

⌋*(SRS_BSW_00101)*

[SWS_EthSwt_00007] ⌈The function EthSwt_Init shall store the access to the configuration structure for subsequent API calls.⌋*(SRS_BSW_00101)*

[SWS_EthSwt_00008] ⌈The function EthSwt_Init shall change the state of all switches controlled by this Switch Driver from ETHSWT_STATE_UNINIT to ETHSWT_STATE_INIT.⌋*(SRS_BSW_00101)*

[SWS_EthSwt_00421] ⌈The EthSwt shall check for enabled port mirror configuration. The enabled port mirror configuration shall be activated by reconfiguring the Ethernet switch hardware according to the port mirror configuration, before frame forwarding is being enabled.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00422] ⌈If the PortMirrorState is set to 0x01 (port mirroring enabled), then the stored port mirror configuration for the given Ethernet switch shall be written to hardware registers of the given Ethernet switch and enable port mirroring.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00423] ⌈If the PortMirrorState is set to 0x00 (port mirroring disabled) the corresponding hardware registers of the given Ethernet switch shall be reset (to the HW's default values) and the port mirroring shall be disabled.⌋*()*

[SWS_EthSwt_00011] ⌈After initialization of the Ethernet switch within the EthSwt_BackgroundTask, the Ethernet switch shall enter an inactive or low power mode if EthSwtLowPowerModeSupport is set to TRUE. If EthSwtLowPowerModeSupport is not defined or set to FALSE the Ethernet switch shall enter an active state.⌋*(SRS_BSW_00101)*

Note: The execution of this function may take a long time (e.g. port structure, VLAN configuration, internal Ethernet switch engine ... a.s.o.) and therefore cannot be called by EcuM or BswM. Instead it should be called e.g. by a background task (see EthSwt_BackgroundTask).

[SWS_EthSwt_00374] ⌈All Ethernet switch HW ports which are not configured as a EthSwtPort shall be switched off during initialization. This Ethernet switch HW ports shall never be switched on during runtime⌋*()*

[SWS_EthSwt_00375] ⌈All EthSwtPorts shall be set to ETH_MODE_DOWN during initialization.⌋*()*

[SWS_EthSwt_00016] ⌈The function EthSwt_Init shall check the access to the Ethernet Switch hardware, i.e. by trying to read or write registers during the configuration of the switch. If the access to the registers fails, the function shall raise the extended production error ETHSWT_E_ACCESS and return E_NOT_OK.⌋*(SRS_BSW_00386)*

Note: Access to the Ethernet Switch hardware is device dependent, e.g. access through the Ethernet Controller Mii, access through SPI, ... etc.

8.3.2 EthSwt_SetSwitchPortMode

[SWS_EthSwt_00018] Definition of API function EthSwt_SetSwitchPortMode ⌈

Service Name	EthSwt_SetSwitchPortMode	

Syntax	`Std_ReturnType EthSwt_SetSwitchPortMode (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  Eth_ModeType PortMode`
	`)`	

Service ID [hex]	0x03	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
	PortMode	ETH_MODE_DOWN: Disable the addressed Ethernet switch port at the given Ethernet switch ETH_MODE_ACTIVE: Enable the addressed Ethernet switch port at the given Ethernet switch ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST: Enable the addressed Ethernet switch port at the given Ethernet switch and request to trigger a wake-up on the network. (This could be used e.g. for Ethernet hardware which is compatible with the OA TC10)
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: The indexed switch port could not be set to Port Mode, or the function is called in state `ETHSWT_STATE_UNINIT` or `ETHSWT_STATE_INIT`.
Description	Enables/disables the indexed switch port	
Available via	EthSwt.h	

⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00019] ⌈The function `EthSwt_SetSwitchPortMode` shall put the indexed port of the switch into the specified mode. If EthSwtPort references an EthTrcv then the function `EthTrcv_SetTransceiverMode` of the Ethernet Transceiver Driver shall additionally be called with the corresponding transceiver mode.⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00396] ⌈When calling the function `EthSwt_SetSwitchPortMode` with mode `ETH_MODE_DOWN`, the EthSwt shall disable the Ethernet switch port directly for reduction of power consumption, if it is possible.⌋*()*

[SWS_EthSwt_00397] ⌈When calling the function `EthSwt_SetSwitchPortMode`, the function shall check the access to the Ethernet switch driver. If the check fails, the function shall raise the extended production error `ETHSWT_E_ACCESS` and return `E_NOT_OK`, otherwise pass the extended production error `ETHSWT_E_ACCESS` and return `E_OK`.⌋*()*

[SWS_EthSwt_00398] ⌈If EthSwtPort does not references an EthTrcv, EthSwt shall indicate a mode of the port by the API `EthIf_SwitchPortModeIndication` latest during the next `EthSwt_MainFunction`.⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00022] ⌈The function EthSwt_SetSwitchPortMode shall be pre compile time configurable On/Off by the configuration parameter: EthSwtSetSwitchPortModeApi.⌋*(SRS_BSW_00171)*

[SWS_EthSwt_00023] ⌈If the switch is already in the requested mode E_OK shall be returned and no development error shall be raised.⌋*(SRS_Eth_00118)*

8.3.3 EthSwt_GetSwitchPortMode

[SWS_EthSwt_00025] Definition of API function EthSwt_GetSwitchPortMode ⌈

Service Name	EthSwt_GetSwitchPortMode	

Syntax	`Std_ReturnType EthSwt_GetSwitchPortMode (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  Eth_ModeType* SwitchModePtr`
	`)`	

Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	SwitchModePtr	ETH_MODE_DOWN: The Ethernet switch port of the given Ethernet switch is disabled ETH_MODE_ACTIVE: The Ethernet switch port of the given Ethernet switch is enabled
Return value	Std_ReturnType	E_OK: success E_NOT_OK: The mode of the indexed switch port could not be obtained, or the function is called in state ETHSWT_STATE_UNINIT or ETHSWT_STATE_INIT.
Description	Obtains the mode of the indexed switch port	
Available via	EthSwt.h	

⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00026] ⌈The function EthSwt_GetSwitchPortMode shall read the mode of the indexed port of the switch. If EthSwtPort references an EthTrcv then the function shall additionally call the corresponding function EthTrcv_GetTransceiverMode of the Ethernet Transceiver Driver.⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00439] ⌈The function shall report the active mode always as ETH_MODE_ACTIVE, even though the previous requested (via EthSwt_SetSwitchPortMode) mode was ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST.⌋*()*

[SWS_EthSwt_00399] ⌈If the obtained modes of the EthSwtPort and the EthTrcv are not aligned, the function EthSwt_GetSwitchPortMode shall raise the extended production error ETHSWT_E_SYNCPORT2PHY and return E_NOT_OK.

If EthTrcv_GetTransceiverMode returns E_NOT_OK, the EthSwt_GetSwitchPortMode shall also return E_NOT_OK without raising an error.⌋*()*

[SWS_EthSwt_00400] ⌈If the function EthSwt_GetSwitchPortMode is called, the function shall check the access to the Ethernet Switch Driver. If the check fails, the function shall raise the extended production error ETHSWT_E_ACCESS and return E_NOT_OK, otherwise pass the production error ETHSWT_E_ACCESS and return E_OK.⌋*()*

[SWS_EthSwt_00029] ⌈The function EthSwt_GetSwitchPortMode shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetSwitchPortModeApi.⌋*(SRS_BSW_00171)*

8.3.4 EthSwt_StartSwitchPortAutoNegotiation

[SWS_EthSwt_00031] Definition of API function EthSwt_StartSwitchPortAutoNegotiation ⌈

Service Name	EthSwt_StartSwitchPortAutoNegotiation	

Syntax	`Std_ReturnType EthSwt_StartSwitchPortAutoNegotiation (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx`
	`)`	

Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: success E_NOT_OK: Automatic negotiation could not be started for the indexed switch port, or the function is called in state ETHSWT_STATE_UNINIT or ETHSWT_STATE_INIT.
Description	Starts the auto-negotiation of the indexed switch port	
Available via	EthSwt.h	

⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00032] ⌈The function EthSwt_StartSwitchPortAutoNegotiation shall restart the automatic negotiation of the used transmission parameters of the referenced Ethernet transceiver driver by calling the function EthTrcv_StartAutoNegotiation.⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00035] ⌈The function EthSwt_StartSwitchPortAutoNegotiation shall be pre compile time configurable On/Off by the configuration parameter: EthSwtStartSwitchPortAutoNegotiationApi.⌋*(SRS_BSW_00171)*

8.3.5 EthSwt_CheckWakeup

[SWS_EthSwt_91003] Definition of API function EthSwt_CheckWakeup ⌈

Service Name	EthSwt_CheckWakeup	

Syntax	`Std_ReturnType EthSwt_CheckWakeup (`
	`  uint8 SwitchIdx`
	`)`	

Service ID [hex]	0x4c	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: request to check for a wake-up is accepted `E_NOT_OK`: request to check for a wake-up is not accepted
Description	API is called by EthIf. The Ethernet switch driver request to check for a wake-up at all Ethernet switch ports which reference an EthTrcv. For those Ethernet switch ports the call is forwarded to the referenced EthTrcv. The function could be called in context of an interrupt service routine or on task level Note: Interrupt service routine consuming time has to be considered, since all EthSwtPorts of the maintained Ethernet switches has to be checked. Therefore the call is forwarded to the referred EthTrcv where the request to check for wake-up is stored. The check of the Ethernet hardware is done asynchronously in the context of the EthTrcv_MainFunction.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00440] ⌈The function `EthSwt_CheckWakeup` shall iterate over the Ethernet switch ports of the indexed Ethernet switch and forward the call to `EthTrcv_CheckWakeup` for those Ethernet switch ports, which reference an EthTrcv.⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00441] ⌈The function `EthSwt_CheckWakeup` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtCheckWakeupApi`⌋ *(SRS_BSW_00171)*

8.3.6 EthSwt_GetSwitchPortWakeupReason

[SWS_EthSwt_91040] Definition of API function EthSwt_GetSwitchPortWakeupReason ⌈

Service Name	EthSwt_GetSwitchPortWakeupReason

Syntax	`Std_ReturnType EthSwt_GetSwitchPortWakeupReason (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  EthTrcv_WakeupReasonType Reason`
	`)`

Service ID [hex]	0x4b
Sync/Async	Synchronous

▽

△

Reentrancy	Reentrant	
Parameters (in)	SwitchIdx	Index of the Ethernet switch within the context of the Ethernet Switch driver
	SwitchPortIdx	Index of the Ethernet switch port index in the context of the Ethernet switch driver
Parameters (inout)	None	
Parameters (out)	Reason	Pointer to structure of least recent wakeup event, which was detected by the Ethernet switch port
Return value	Std_ReturnType	`E_OK`: Ethernet switch port wake up reason request has been accepted. `E_NOT_OK`: Ethernet switch port wake up reason request has not been accepted.
Description	This function obtains the wake up reasons of the the indexed Ethernet switch port by calling Eth Trcv_GetBusWuReason(...) of the referenced EthTrcv	
Available via	EthSwt.h	

⌋*(SRS_Eth_00107)*

[SWS_EthSwt_00442] ⌈The function `EthSwt_GetSwitchPortWakeupReason` shall read the current wake-up reason of the indexed Ethernet switch port by forwarding the call to `EthTrcv_GetBusWuReason` of the referenced EthTrcv. If the indexed Ethernet switch port has no reference to an EthTrcv, the function shall return `E_NOT_OK`.⌋*(SRS_Eth_00107)*

[SWS_EthSwt_00443] ⌈The function `EthSwt_GetSwitchPortWakeupReason` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtGetSwitchPortWakeupReasonApi`⌋*(SRS_BSW_00171)*

8.3.7 EthSwt_GetLinkState

[SWS_EthSwt_00037] Definition of API function EthSwt_GetLinkState ⌈

Service Name	EthSwt_GetLinkState	

Syntax	`Std_ReturnType EthSwt_GetLinkState (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  EthTrcv_LinkStateType* LinkStatePtr`
	`)`	

Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	LinkStatePtr	ETHTRCV_LINK_STATE_DOWN: Switch port is disconnected ETHTRCV_LINK_STATE_ACTIVE: Switch port is connected

▽

△

Return value	Std_ReturnType	E_OK: success E_NOT_OK: Link state of the indexed switch port could not be obtained, or the function is called in state ETHSWT_STATE_UNINIT or ETHSWT_STATE_INIT.
Description	Obtains the link state of the indexed switch port	
Available via	EthSwt.h	

⌋*(SRS_Eth_00119)*

[SWS_EthSwt_00038] ⌈The function EthSwt_GetLinkState shall read the current (link) state of the indexed switch port. If the indexed Ethernet port references an Ethernet transceiver, the link state shall be obtained by calling the function EthTrcv_GetLinkState of the Ethernet Transceiver Driver. If the indexed Ethernet Switch port does not reference an Ethernet transceiver, the state shall be obtained from the MAC interface of the Switch port. If the MAC interface is not able to provide a link state (e.g. Ethernet hardware does not support a link state of the MAC interface), the API shall return the following state which is derived from the current mode:

- If the current mode of the indexed switch port is ETH_MODE_ACTIVE, then ETHTRCV_LINK_STATE_ACTIVE shall be returned
- If the current mode of the indexed switch port is ETH_MODE_DOWN, then ETHTRCV_LINK_STATE_DOWN shall be returned

⌋*(SRS_Eth_00118, SRS_Eth_00119)*

[SWS_EthSwt_00042] ⌈The function EthSwt_GetLinkState shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetLinkStateApi.⌋*(SRS_BSW_00171)*

8.3.8 EthSwt_GetBaudRate

[SWS_EthSwt_00044] Definition of API function EthSwt_GetBaudRate ⌈

Service Name	EthSwt_GetBaudRate	

Syntax	Std_ReturnType EthSwt_GetBaudRate (
	  uint8 SwitchIdx,
	  uint8 SwitchPortIdx,
	  EthTrcv_BaudRateType* BaudRatePtr
	)	

Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	

▽

△

Parameters (out)	BaudRatePtr	ETHTRCV_BAUD_RATE_10MBIT: 10MBit connection ETHTRCV_BAUD_RATE_100MBIT: 100MBit connection ETHTRCV_BAUD_RATE_1000MBIT: 1000MBit connection ETHTRCV_BAUD_RATE_2500MBIT: 2500MBit connection
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: Baud rate of the indexed switch port could not be obtained, or the function is called in state `ETHSWT_STATE_UNINIT` or `ETHSWT_STATE_INIT`.
Description	Obtains the baud rate of the indexed switch port	
Available via	EthSwt.h	

⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00045] ⌈The function `EthSwt_GetBaudRate` shall read the current baud rate of the indexed switch port. If the indexed Ethernet port reference an Ethernet transceiver, the baud rate shall be obtained by the function `EthTrcv_GetBaudRate` of the Ethernet Transceiver Driver. If the indexed Ethernet Switch port does not reference an Ethernet transceiver, the baud rate shall be obtained from the MAC interface of the Switch port.⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00049] ⌈The function `EthSwt_GetBaudRate` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtGetBaudRateApi`.⌋*(SRS_BSW_00171)*

8.3.9 EthSwt_GetDuplexMode

[SWS_EthSwt_00051] Definition of API function EthSwt_GetDuplexMode ⌈

Service Name	EthSwt_GetDuplexMode	

Syntax	`Std_ReturnType EthSwt_GetDuplexMode (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  EthTrcv_DuplexModeType* DuplexModePtr`
	`)`	

Service ID [hex]	0x08	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	DuplexModePtr	ETHTRCV_DUPLEX_MODE_HALF: half duplex connections ETHTRCV_DUPLEXMODE_FULL: full duplex connection
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: duplex mode of the indexed switch port could not be obtained, or the function is called in state `ETHSWT_STATE_UNINIT` or `ETHSWT_STATE_INIT`.
Description	Obtains the duplex mode of the indexed switch port	
Available via	EthSwt.h	

⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00052] ⌈The function EthSwt_GetDuplexMode shall read the current duplex mode of the indexed switch port. If the indexed Ethernet port reference an Ethernet transceiver, the duplex mode shall be obtained by calling the function EthTrcv_GetDuplexMode of the Ethernet Transceiver Driver. If the indexed Ethernet Switch port does not reference an Ethernet transceiver, the duplex mode shall be obtained from the MAC interface of the Switch port.⌋*(SRS_Eth_00118)*

[SWS_EthSwt_00056] ⌈The function EthSwt_GetDuplexMode shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetDuplexModeApi.⌋*(SRS_BSW_00171)*

8.3.10 EthSwt_GetPortMacAddr

[SWS_EthSwt_00060] Definition of API function EthSwt_GetPortMacAddr ⌈

Service Name	EthSwt_GetPortMacAddr	

Syntax	`Std_ReturnType EthSwt_GetPortMacAddr (`
	`  uint8 SwitchIdx,`
	`  const uint8* MacAddrPtr,`
	`  uint8* PortIdxPtr`
	`)`	

Service ID [hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	MacAddrPtr	MAC-address for which a switch port is searched over which the node with this MAC-address can be reached.
Parameters (inout)	None	
Parameters (out)	PortIdxPtr	Pointer to the port index
Return value	Std_ReturnType	E_OK: success E_NOT_OK: multiple ports were found
Description	Obtains the port over which this MAC-address at the indexed switch can be reached. The result might be used for a DHCP-server which will need the port/MAC-resolution. If for the PortIdxPtr the maximal possible value (255) is returned the given MAC address cannot be reached via a port of this switch. If multiple ports were found the API returns E_NOT_OK.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00230] ⌈The function EthSwt_GetPortMacAddr shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetPortMacAddrApi.⌋*(SRS_BSW_00171)*

8.3.11 EthSwt_GetArlTable

[SWS_EthSwt_00111] Definition of API function EthSwt_GetArlTable ⌈

Service Name	EthSwt_GetArlTable	

Syntax	`Std_ReturnType EthSwt_GetArlTable (`
	`  uint8 switchIdx,`
	`  uint16* numberOfElements,`
	`  Eth_MacVlanType* arlTableListPointer`
	`)`	

Service ID [hex]	0x0a	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	switchIdx	Index of the switch within the context of the Ethernet Switch Driver
Parameters (inout)	numberOfElements	In: Maximum number of elements which can be written into the arlTable Out: Number of elements which are currently available in the EthSwitch module.
Parameters (out)	arlTableListPointer	Returns a pointer to the memory where the ARL table of the switch consisting of a list of structs with MAC-address, VLAN-ID and port shall be stored.
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: requested switchIdx is not valid or inactive
Description	Obtains the address resolution table of a switch and copies the list into a user provided buffer. The function will copy all or numberOfElements into the output list. If input value of numberOf Elements is 0 the function will not copy any data but only return the number of valid entries in the cache. arlTableListPointer may be NULL_PTR in this case.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00228] ⌈The function `EthSwt_GetArlTable` shall provide a list of structs with MAC-address, VLAN-ID and port for the indexed switch.⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00197] ⌈If the numberOfElements is greater 0x00, the arlTableListPointer shall be filled with up to numberOfElements elements. numberOfElements shall return the number of copied elements.⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00235] ⌈The `EthSwt_GetArlTable` API shall return only the numberOfElements if the numberOfElements is set to 0x00. In this case no data will be copied and a NULLPTR can be used for the arlTableListPointer.⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00229] ⌈The function `EthSwt_GetArlTable` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtGetArlTableApi`.⌋*(SRS_BSW_00171)*

8.3.12 EthSwt_GetCounterValues

[SWS_EthSwt_00231] Definition of API function EthSwt_GetCounterValues ⌈

Service Name	EthSwt_GetCounterValues	

Syntax	`Std_ReturnType EthSwt_GetCounterValues (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  Eth_CounterType* CounterPtr`
	`)`	

Service ID [hex]	0x0c	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	CounterPtr	counter values according to IETF RFC 1757, RFC 1643 and RFC 2233.
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: counter values read failure
Description	Reads a list with drop counter values of the corresponding port of the switch. The meaning of these values is described at Eth_CounterType.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00128)*

[SWS_EthSwt_00106] ⌈ `EthSwt_GetCounterValues` shall read a list with drop counter values of the corresponding port of the switch. The meaning of these values is described at `Eth_CounterType`.⌋*(SRS_Eth_00128)*

8.3.13 EthSwt_GetRxStats

[SWS_EthSwt_00198] Definition of API function EthSwt_GetRxStats ⌈

Service Name	EthSwt_GetRxStats	

Syntax	`Std_ReturnType EthSwt_GetRxStats (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  Eth_RxStatsType* RxStats`
	`)`	

Service ID [hex]	0x0d	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	RxStats	List of values according to IETF RFC 2819 (Remote Network Monitoring Management Information Base)
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: drop counter could not be obtained

▽

△

Description	Returns a list of statistic counters defined with Eth_RxTatsType. The majority of these Counters are derived from the IETF RFC2819.
Available via	EthSwt.h

⌋(*SRS_Eth_00128*)

[SWS_EthSwt_00199] ⌈EthSwt_GetRxStats shall return a list of statistic counters defined with Eth_RxStatsType. The majority of these Counters are derived from the IETF RFC2819.⌋(*SRS_Eth_00128*)

[SWS_EthSwt_00202] ⌈The function EthSwt_GetRxStats shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetRxStatsApi.⌋(*SRS_BSW_00171*)

8.3.14 EthSwt_GetTxStats

[SWS_EthSwt_91001] Definition of API function EthSwt_GetTxStats ⌈

Service Name	EthSwt_GetTxStats	

Syntax	`Std_ReturnType EthSwt_GetTxStats (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  Eth_TxStatsType* TxStats`
	`)`	

Service ID [hex]	0x20	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	TxStats	List of values to read statistic values for transmission.
Return value	Std_ReturnType	E_OK: success E_NOTOK: Tx-statistics could not be obtained
Description	Returns the list of Transmission Statistics out of IETF RFC1213 defined with Eth_TxStatsType, where the maximal possible value shall denote an invalid value, e.g. this counter is not available.	
Available via	EthSwt.h	

⌋(*SRS_Eth_00128*)

[SWS_EthSwt_00372] ⌈EthSwt_GetTxStats shall return the list of Transmission Statistics out of IETF RFC1213 defined with Eth_TxStatsType, where the maximal possible value shall denote an invalid value, e.g. this counter is not available.⌋(*SRS_Eth_00128*)

[SWS_EthSwt_00362] ⌈The function EthSwt_GetTxStats shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetTxStatsApi.⌋(*SRS_BSW_00171*)

8.3.15 EthSwt_GetTxErrorCounterValues

[SWS_EthSwt_91000] Definition of API function EthSwt_GetTxErrorCounterValues ⌈

Service Name	EthSwt_GetTxErrorCounterValues	

Syntax	`Std_ReturnType EthSwt_GetTxErrorCounterValues (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  Eth_TxErrorCounterValuesType* TxStats`
	`)`	

Service ID [hex]	0x21	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Drive
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	TxStats	List of values to read statistic error counter values for transmission.
Return value	Std_ReturnType	`E_OK`: success, `E_NOTOK`: Tx-statistics could not be obtained
Description	Returns the list of Transmission Error Counters out of IETF RFC1213 and RFC1643 defined with Eth_TxErrorCounterValuesType, where the maximal possible value shall denote an invalid value, e.g. this counter is not available.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00128)*

[SWS_EthSwt_00373] ⌈`EthSwt_GetTxErrorCounterValues` returns the list of Transmission Error Counters out of IETF RFC1213 and RFC1643 defined with `Eth_TxErrorCounterValuesType`, where the maximal possible value shall denote an invalid value, e.g. this counter is not available.⌋*(SRS_Eth_00128)*

[SWS_EthSwt_00370] ⌈The function `EthSwt_GetTxErrorCounterValues` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtGetTxErrorCounterValuesApi`.⌋*(SRS_BSW_00171)*

8.3.16 EthSwt_GetSwitchReg

[SWS_EthSwt_00206] Definition of API function EthSwt_GetSwitchReg ⌈

Service Name	EthSwt_GetSwitchReg

Syntax	`Std_ReturnType EthSwt_GetSwitchReg (`
	`  uint8 SwitchIdx,`
	`  uint32 page,`
	`  uint32 register,`
	`  uint32* registerContent`
	`)`

Service ID [hex]	0x0e
Sync/Async	Synchronous

▽

△

Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	page	Address of a register page
	register	Address of a register
Parameters (inout)	None	
Parameters (out)	registerContent	Content of the addresses register
Return value	Std_ReturnType	E_OK: success E_NOT_OK: drop counter could not be obtained
Description	Generic API for reading the content of a switch register	
Available via	EthSwt.h	

⌋*(SRS_Eth_00120)*

[SWS_EthSwt_00207] ⌈The function EthSwt_GetSwitchReg shall read the content of a switch register.⌋*(SRS_Eth_00120)*

[SWS_EthSwt_00210] ⌈The function EthSwt_GetSwitchReg shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetSwitchRegApi.⌋*(SRS_BSW_00171)*

8.3.17 EthSwt_SetSwitchReg

[SWS_EthSwt_00211] Definition of API function EthSwt_SetSwitchReg ⌈

Service Name	EthSwt_SetSwitchReg	

Syntax	`Std_ReturnType EthSwt_SetSwitchReg (`
	`  uint8 SwitchIdx,`
	`  uint32 page,`
	`  uint32 register,`
	`  uint32 registerContent`
	`)`	

Service ID [hex]	0x0f	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	page	Address of a register page
	register	Address of a register
	registerContent	Content of the addresses register
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: success E_NOT_OK: drop counter could not be obtained
Description	Generic API for writing the content of a switch register	
Available via	EthSwt.h	

⌋*(SRS_Eth_00120)*

[SWS_EthSwt_00212] ⌈The function EthSwt_SetSwitchReg shall write the content to the switch register.⌋*(SRS_Eth_00120)*

[SWS_EthSwt_00215] ⌈The function EthSwt_SetSwitchReg shall be pre compile time configurable On/Off by the configuration parameter: EthSwtSetSwitchRegApi.⌋*(SRS_BSW_00171)*

8.3.18 EthSwt_ReadTrcvRegister

[SWS_EthSwt_00216] Definition of API function EthSwt_ReadTrcvRegister ⌈

Service Name	EthSwt_ReadTrcvRegister	

Syntax	`Std_ReturnType EthSwt_ReadTrcvRegister (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  uint8 RegIdx,`
	`  uint16* RegValPtr`
	`)`	

Service ID [hex]	0x10	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
	RegIdx	Index of the register
Parameters (inout)	None	
Parameters (out)	RegValPtr	Pointer to the register content
Return value	Std_ReturnType	E_OK: success E_NOT_OK: Content of the transceiver could not be obtained, or the function is called in state ETHSWT_STATE_UNINIT or ETHSWT_STATE_INIT.
Description	Generic API for reading the content of a transceiver register	
Available via	EthSwt.h	

⌋*(SRS_Eth_00120)*

[SWS_EthSwt_00217] ⌈The function EthSwt_ReadTrcvRegister shall read the specified transceiver register through the MII or SPI of the indexed switch port.⌋*(SRS_Eth_00118, SRS_Eth_00120)*

[SWS_EthSwt_00220] ⌈The function EthSwt_ReadTrcvRegister shall be pre compile time configurable On/Off by the configuration parameter: EthSwtReadTrcvRegisterApi.⌋*(SRS_BSW_00171)*

8.3.19 EthSwt_WriteTrcvRegister

[SWS_EthSwt_00221] Definition of API function EthSwt_WriteTrcvRegister ⌈

Service Name	EthSwt_WriteTrcvRegister	

Syntax	`Std_ReturnType EthSwt_WriteTrcvRegister (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  uint8 RegIdx,`
	`  uint16 RegVal`
	`)`	

Service ID [hex]	0x11	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
	RegIdx	Index of the register
	RegVal	Content for the indexed register
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: Content given by RegVal could not be written to the given register (RegIdx) of the transceiver, or the function is called in state `ETHSWT_STATE_UNINIT` or `ETHSWT_STATE_INIT`.
Description	Generic API for writing the content of a transceiver register	
Available via	EthSwt.h	

⌋ *(SRS_Eth_00120)*

[SWS_EthSwt_00222] ⌈The function `EthSwt_WriteTrcvRegister` shall write the specified transceiver register through the MII or SPI of the indexed switch port.⌋ *(SRS_Eth_00118, SRS_Eth_00120)*

[SWS_EthSwt_00225] ⌈The function `EthSwt_WriteTrcvRegister` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtWriteTrcvRegisterApi`.⌋ *(SRS_BSW_00171)*

8.3.20 EthSwt_EnableVlan

[SWS_EthSwt_00172] Definition of API function EthSwt_EnableVlan ⌈

Service Name	EthSwt_EnableVlan

Syntax	`Std_ReturnType EthSwt_EnableVlan (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  uint16 VlanId,`
	`  boolean Enable`
	`)`

Service ID [hex]	0x12
Sync/Async	Synchronous

▽

△

Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
	VlanId	VLAN-ID to a preconfigured configuration on the given ingress port
	Enable	1 = VLAN-configuration enabled 0 = VLAN-configuration disabled (frames with given VLAN-ID will be dropped)
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: buffer level could not be obtained
Description	Enables or disables a pre-configured VLAN at a certain port of a switch.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00121, SRS_Eth_00114)*

[SWS_EthSwt_00173] ⌈The function `EthSwt_EnableVlan` shall enable or disable a pre-configured VLAN at a certain port of a switch.⌋*(SRS_Eth_00121, SRS_Eth_00114)*

[SWS_EthSwt_00177] ⌈The function `EthSwt_EnableVlan` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtEnableVlanApi`.⌋*(SRS_BSW_00171)*

8.3.21 EthSwt_StoreConfiguration

[SWS_EthSwt_00086] Definition of API function EthSwt_StoreConfiguration ⌈

Service Name	EthSwt_StoreConfiguration	

Syntax	`Std_ReturnType EthSwt_StoreConfiguration (`
	`  uint8 SwitchIdx`
	`)`	

Service ID [hex]	0x13	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Request to persistently store the MAC/Port table was accepted `E_NOT_OK`: Request to persistently store the MAC/Port table was not accepted
Description	Trigger the storage/reset of the configuration of the learned MAC/Port tables of a switch in a persistent manner and will be used by e.g. CDD.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00087, SRS_Eth_00122)*

[SWS_EthSwt_00090] ⌈The function `EthSwt_StoreConfiguration` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtStoreConfigurationApi`.⌋*(SRS_BSW_00171)*

8.3.22 EthSwt_ResetConfiguration

[SWS_EthSwt_00091] Definition of API function EthSwt_ResetConfiguration ⌈

Service Name	EthSwt_ResetConfiguration	

Syntax	`Std_ReturnType EthSwt_ResetConfiguration (`
	`  uint8 SwitchIdx`
	`)`	

Service ID [hex]	0x14	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Request to persistently reset the MAC/Port table was accepted `E_NOT_OK`: Request to persistently reset the MAC/Port table was not accepted
Description	The function shall request to reset and store the configuration of the learned MAC/Port tables of a Ethernet switch in a persistent manner. This could be used by e.g. a CDD.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00087, SRS_Eth_00122)*

[SWS_EthSwt_00095] ⌈The function `EthSwt_ResetConfiguration` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtResetConfigurationApi`.⌋*(SRS_BSW_00171)*

8.3.23 EthSwt_SetMacLearningMode

[SWS_EthSwt_00182] Definition of API function EthSwt_SetMacLearningMode ⌈

Service Name	EthSwt_SetMacLearningMode	

Syntax	`Std_ReturnType EthSwt_SetMacLearningMode (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  EthSwt_MacLearningType MacLearningMode`
	`)`	

Service ID [hex]	0x15	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
	MacLearningMode	Defines whether MAC addresses shall be learned and if they shall be learned in software or hardware.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: configuration could be persistently reset

▽

△

Description	Sets the MAC learning mode in one of the tree modes: 1.) HW learning enabled, 2.) Hardware learning disabled, 3.) Software learning enabled. Note: This feature is hardware dependent, i.e. the switch hardware needs to support the different learning modes.
Available via	EthSwt.h

⌋*(SRS_Eth_00087, SRS_Eth_00122)*

[SWS_EthSwt_00183] ⌈The function `EthSwt_SetMacLearningMode` shall set the MAC learning mode according to `EthSwt_MacLearningType`.⌋*(SRS_Eth_00122, SRS_Eth_00087)*

Note: This feature is hardware dependent, i.e. the switch hardware needs to support the different modes.

[SWS_EthSwt_00186] ⌈The function `EthSwt_SetMacLearningMode` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtSetMacLearningModeApi`.⌋*(SRS_BSW_00171)*

8.3.24 EthSwt_GetMacLearningMode

[SWS_EthSwt_00187] Definition of API function EthSwt_GetMacLearningMode ⌈

Service Name	EthSwt_GetMacLearningMode	

Syntax	`Std_ReturnType EthSwt_GetMacLearningMode (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  EthSwt_MacLearningType* MacLearningMode`
	`)`	

Service ID [hex]	0x16	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	SwitchPortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	MacLearningMode	Defines whether MAC addresses shall be learned and if they shall be learned in software or hardware.
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: configuration could be persistently reset
Description	Returns the MAC learning mode, i.e. 1.) HW learning enabled, 2.) Hardware learning disabled, 3.) Software learning enabled. Note: This feature is hardware dependent, i.e. the switch hardware needs to support the different learning modes	
Available via	EthSwt.h	

⌋*(SRS_Eth_00087)*

[SWS_EthSwt_00188] ⌈The function `EthSwt_GetMacLearningMode` shall return the MAC learning mode according to `EthSwt_MacLearningType`.⌋*(SRS_Eth_00087)*

Note: This feature is hardware dependent, i.e. the switch hardware needs to support the different learning modes.

[SWS_EthSwt_00191] ⌈The function EthSwt_GetMacLearningMode shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetMacLearningModeApi.⌋*(SRS_BSW_00171)*

8.3.25 EthSwt_NvmSingleBlockCallback

[SWS_EthSwt_00125] Definition of callback function EthSwt_NvmSingleBlockCallback ⌈

Service Name	EthSwt_NvmSingleBlockCallback	

Syntax	`Std_ReturnType EthSwt_NvmSingleBlockCallback (`
	`  NvM_BlockRequestType BlockRequest,`
	`  NvM_RequestResultType JobResult`
	`)`	

Service ID [hex]	0x17	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	BlockRequest	The request type (read, write, ... etc.) of the previous processed block job
	JobResult	Covers the job result of the previous processed single block job.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: success E_NOT_OK: Callback function has not been processed successfully
Description	Function will be called by the NVRAMManager after the switch configuration has been stored or resetted.	
Available via	EthSwt_NvM.h	

⌋*(SRS_Eth_00087, SRS_Eth_00122)*

[SWS_EthSwt_00126] ⌈The function EthSwt_NvmSingleBlockCallback shall be called by the NVRAMManager [16] after the switch configuration has been stored or reset in the the NV RAM.⌋*(SRS_Eth_00122, SRS_Eth_00087)*

[SWS_EthSwt_00196] ⌈The function EthSwt_NvmSingleBlockCallback shall call the function <user>_PersistentConfigurationResult to provide the JobResult to the caller of EthSwt_StoreConfiguration or EthSwt_ResetConfiguration.⌋*(SRS_Eth_00122, SRS_Eth_00087)*

[SWS_EthSwt_00127] ⌈The function EthSwt_NvmSingleBlockCallback shall always return E_OK according to SWS_NvM_00368.⌋*(SRS_Eth_00122, SRS_Eth_00087)*

[SWS_EthSwt_00128] ⌈The function EthSwt_NvmSingleBlockCallback shall raise a development error if the JobResult equals NVM_REQ_NOT_OK, i.e. the write request has been finished unsuccessfully.⌋*(SRS_BSW_00369)*

Note: Please note that a production error at this point is not necessary because the NvM will raise also a production error if the write to NV RAM was not successful.

[SWS_EthSwt_00129] ⌈The function EthSwt_NvmSingleBlockCallback shall be pre compile time configurable On/Off by the existence of the container EthSwtNvm.⌋ *(SRS_BSW_00171)*

8.3.26 EthSwt_GetVersionInfo

[SWS_EthSwt_00058] Definition of API function EthSwt_GetVersionInfo ⌈

Service Name	EthSwt_GetVersionInfo	

Syntax	`void EthSwt_GetVersionInfo (`
	`  Std_VersionInfoType* VersionInfoPtr`
	`)`	

Service ID [hex]	0x18	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	VersionInfoPtr	Pointer to where to store the version information of this module.
Return value	None	
Description	Returns the version information of this module.	
Available via	EthSwt.h	

⌋ *(SRS_BSW_00171)*

[SWS_EthSwt_00124] ⌈The function EthSwt_GetVersionInfo shall be pre compile time configurable On/Off by the configuration parameter: EthSwtVersionInfoApi.⌋ *(SRS_BSW_00171)*

8.3.27 EthSwt_EthRxProcessFrame

[SWS_EthSwt_91004] Definition of API function EthSwt_EthRxProcessFrame ⌈

Service Name	EthSwt_EthRxProcessFrame	

Syntax	`Std_ReturnType EthSwt_EthRxProcessFrame (`
	`  uint8 CtrlIdx,`
	`  Eth_BufIdxType BufIdx,`
	`  uint8** DataPtr,`
	`  uint16* LengthPtr,`
	`  boolean* IsMgmtFrameOnlyPtr`
	`)`	

Service ID [hex]	0x23	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Ethernet Controller index
	BufIdx	Ethernet Rx Buffer index

▽

△

Parameters (inout)	DataPtr	IN: Pointer to the position of the EtherType of a common Ethernet frame OUT: Pointer to the position of the EtherType in the management frame
	LengthPtr	IN: Pointer to the length of the frame received OUT: Pointer to the length decreased by the management information length.
Parameters (out)	IsMgmtFrameOnlyPtr	Information about the kind of frame FALSE: Frame is not only for management purpose, but also for normal communication. TRUE: Frame is only for management purpose and must not be processed in common receive process
Return value	Std_ReturnType	`E_OK`: Frame successfully processed `E_NOT_OK`: Frame processing failed
Description	Function inspects the Ethernet frame passed by the data pointer for management information and stores it for later use in EthSwt_EthRxFinishedIndication().	
Available via	EthSwt_Eth.h	

⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00249] ⌈The function `EthSwt_EthRxProcessFrame` shall be pre compile time configurable ON/OFF by the configuration parameter: `EthSwtManagementSupportApi` .⌋*(SRS_BSW_00171)*

8.3.28 EthSwt_EthRxFinishedIndication

[SWS_EthSwt_91005] Definition of API function EthSwt_EthRxFinishedIndication ⌈

Service Name	EthSwt_EthRxFinishedIndication	

Syntax	`Std_ReturnType EthSwt_EthRxFinishedIndication (`
	`  uint8 CtrlIdx,`
	`  Eth_BufIdxType BufIdx`
	`)`	

Service ID [hex]	0x24	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Ethernet Controller index
	BufIdx	Ethernet Rx Buffer index
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Frame successfully processed `E_NOT_OK`: Frame processing failed
Description	Indication for a finished receive process for a specific Ethernet frame, which results in providing the management information retrieved during EthSwt_EthRxProcessFrame().	
Available via	EthSwt_Eth.h	

⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00253] ⌈The function EthSwt_EthRxFinishedIndication shall be pre compile time configurable ON/OFF by the configuration parameter: EthSwtManagementSupportApi .⌋*(SRS_BSW_00171)*

8.3.29 EthSwt_EthTxPrepareFrame

[SWS_EthSwt_91006] Definition of API function EthSwt_EthTxPrepareFrame ⌈

Service Name	EthSwt_EthTxPrepareFrame	

Syntax	`Std_ReturnType EthSwt_EthTxPrepareFrame (`
	`  uint8 CtrlIdx,`
	`  Eth_BufIdxType BufIdx,`
	`  uint8** DataPtr,`
	`  uint16* LengthPtr`
	`)`	

Service ID [hex]	0x25	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Ethernet Controller index
	BufIdx	Ethernet Rx Buffer index
Parameters (inout)	DataPtr	IN: Pointer to the position of the EtherType of a common Ethernet frame OUT: Pointer to the position of the EtherType in the management frame
	LengthPtr	IN: Pointer to the length of the buffer without management information OUT: Pointer to the modified length needed for buffer and management information
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Frame successfully prepared E_NOT_OK: Frame preparation failed
Description	Prepares the Ethernet frame for common Ethernet communication (frame shall be handled by switch according to the common address resolution behavior) and stores the information for processing of EthSwt_EthTxFinishedIndication().	
Available via	EthSwt_Eth.h	

⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00257] ⌈The function EthSwt_EthTxPrepareFrame shall be pre compile time configurable ON/OFF by the configuration parameter: EthSwtManagementSupportApi .⌋*(SRS_BSW_00171)*

8.3.30 EthSwt_EthTxAdaptBufferLength

[SWS_EthSwt_91007] Definition of API function EthSwt_EthTxAdaptBufferLength ⌈

Service Name	EthSwt_EthTxAdaptBufferLength	

Syntax	`void EthSwt_EthTxAdaptBufferLength (`
	`  uint16* LengthPtr`
	`)`	

Service ID [hex]	0x26	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	LengthPtr	IN: Pointer to the length of the buffer without management information. OUT: Pointer to the modified length needed for buffer and management information.
Parameters (out)	None	
Return value	None	
Description	Modifies the buffer length to be able to insert management information.	
Available via	EthSwt_Eth.h	

⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00261] ⌈The function EthSwt_EthTxAdaptBufferLength shall be pre compile time configurable ON/OFF by the configuration parameter: EthSwtManagementSupportApi .⌋*(SRS_BSW_00171)*

8.3.31 EthSwt_SetMgmtInfo

[SWS_EthSwt_91008] Definition of API function EthSwt_SetMgmtInfo ⌈

Service Name	EthSwt_SetMgmtInfo	

Syntax	`Std_ReturnType EthSwt_SetMgmtInfo (`
	`  uint8 CtrlIdx,`
	`  Eth_BufIdxType BufIdx,`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr`
	`)`	

Service ID [hex]	0x27	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Ethernet Controller index
	BufIdx	Ethernet Rx Buffer index
	MgmtInfoPtr	Pointer to the management information
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Management infos successfully set `E_NOT_OK`: Setting of management infos failed

▽

△

Description	Extends the Ethernet frame prepared previously by EthSwt_EthTxPrepareFrame() with the management information to achieve transmission only on specific ports.
Available via	EthSwt.h

⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00264] ⌈The function `EthSwt_SetMgmtInfo` shall be pre compile time configurable ON/OFF by the configuration parameter: `EthSwtManagementSupportApi` .⌋*(SRS_BSW_00171)*

8.3.32 EthSwt_EthTxProcessFrame

[SWS_EthSwt_91009] Definition of API function EthSwt_EthTxProcessFrame ⌈

Service Name	EthSwt_EthTxProcessFrame	

Syntax	`Std_ReturnType EthSwt_EthTxProcessFrame (`
	`  uint8 CtrlIdx,`
	`  Eth_BufIdxType BufIdx,`
	`  uint8** DataPtr,`
	`  uint16* LengthPtr`
	`)`	

Service ID [hex]	0x28	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Ethernet Controller index
	BufIdx	Ethernet Rx Buffer index
Parameters (inout)	DataPtr	IN: Pointer to the position of the EtherType of a common Ethernet frame OUT: Pointer to the position of the EtherType in the management frame
	LengthPtr	IN: Pointer to the length of the received frame OUT: Pointer to the length decreased by the management information length
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Frame successfully processed `E_NOT_OK`: Frame processing failed
Description	Function inserts management information into the Ethernet frame.	
Available via	EthSwt_Eth.h	

⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00268] ⌈The function `EthSwt_EthTxProcessFrame` shall be pre compile time configurable ON/OFF by the configuration parameter: `EthSwtManagementSupportApi` .⌋*(SRS_BSW_00171)*

8.3.33 EthSwt_EthTxFinishedIndication

[SWS_EthSwt_91010] Definition of API function EthSwt_EthTxFinishedIndication ⌈

Service Name	EthSwt_EthTxFinishedIndication	

Syntax	`Std_ReturnType EthSwt_EthTxFinishedIndication (`
	`  uint8 CtrlIdx,`
	`  Eth_BufIdxType BufIdx`
	`)`	

Service ID [hex]	0x29	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Ethernet Controller index
	BufIdx	Ethernet Rx Buffer index
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Frame successfully processed `E_NOT_OK`: Frame processing failed
Description	Indication for a finished transmit process for a specific Ethernet frame.	
Available via	EthSwt_Eth.h	

⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00273] ⌈The function `EthSwt_EthTxFinishedIndication` shall be pre compile time configurable ON/OFF by the configuration parameter: `EthSwtManagementSupportApi` .⌋*(SRS_BSW_00171)*

8.3.34 EthSwt_PortEnableTimeStamp

[SWS_EthSwt_91028] Definition of API function EthSwt_PortEnableTimeStamp ⌈

Service Name	EthSwt_PortEnableTimeStamp	

Syntax	`Std_ReturnType EthSwt_PortEnableTimeStamp (`
	`  uint8 CtrlIdx,`
	`  Eth_BufIdxType BufIdx,`
	`  EthSwt_MgmtInfoType* MgmtInfoPtr`
	`)`	

Service ID [hex]	0x40	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Ethernet Controller index
	BufIdx	Ethernet Rx Buffer index
	MgmtInfoPtr	Management information including SwitchIdx and SwitchPortIdx
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Time stamping on egress successfully enabled `E_NOT_OK`: Enabling of time stamping on egress has been failed

▽

△

Description	Activates egress time stamping on a dedicated message object on a dedicated port of a Switch if EthSwtPortTimeStampSupport is set to TRUE for this port. The selective activation of dedicated message objects for time stamping reduces the number of notification calls only to the required calls. Some HW does store once the egress time stamp marker and some HW needs it always before transmission. There will be no disabled functionality, due to the fact, that the message type is always "time stamped" by network design.
Available via	EthSwt.h

⌋*(SRS_Eth_00125)*

[SWS_EthSwt_00379] ⌈The function EthSwt_PortEnableTimeStamp shall be pre compile time configurable ON/OFF by the configuration parameter: EthSwtGlobalTimeSupportApi .⌋*(SRS_BSW_00171)*

8.3.35 EthSwt_VerifyConfig

[SWS_EthSwt_91012] Definition of API function EthSwt_VerifyConfig ⌈

Service Name	EthSwt_VerifyConfig	

Syntax	`Std_ReturnType EthSwt_VerifyConfig (`
	`  uint8 SwitchIdx,`
	`  boolean* Result`
	`)`	

Service ID [hex]	0x31	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
Parameters (inout)	None	
Parameters (out)	Result	Result of verification, TRUE: configureation verified ok, FALSE: configuraton values found corrupted
Return value	Std_ReturnType	E_OK: Configuration verificaton succeeded, E_NOT_OK: Configuration verification not succeeded.
Description	Verifies the Switch Configuration depending on the HW-Architecture, HW-capability and the intended accuracy of this verification.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00126)*

[SWS_EthSwt_00287] ⌈The function EthSwt_VerifyConfig shall be compile time configurable On/Off by the configuration parameter: EthSwtVerifyConfigApi.⌋*(SRS_BSW_00171)*

8.3.36 EthSwt_SetForwardingMode

[SWS_EthSwt_91013] Definition of API function EthSwt_SetForwardingMode ⌈

Service Name	EthSwt_SetForwardingMode	

Syntax	`Std_ReturnType EthSwt_SetForwardingMode (`
	`  uint8 SwitchIdx,`
	`  boolean mode`
	`)`	

Service ID [hex]	0x32	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	mode	True Forewarding enabled, False Forwarding disabled
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: stopping of frame forwarding succeeded, `E_NOT_OK`: stopping of frame forwarding not succeeded.
Description	Configures switch to start or stop forwarding for all ports. This API call may be used during switch configuration verification.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00126)*

[SWS_EthSwt_00291] ⌈The function `EthSwt_SetForwardingMode` shall be compile time configurable On/Off by the configuration parameter: `EthSwtSetForwardingModeApi`.⌋*(SRS_BSW_00171)*

8.3.37 EthSwt_GetPortSignalQuality

[SWS_EthSwt_91014] Definition of API function EthSwt_GetPortSignalQuality ⌈

Service Name	EthSwt_GetPortSignalQuality	

Syntax	`Std_ReturnType EthSwt_GetPortSignalQuality (`
	`  uint8 SwitchIdx,`
	`  uint8 PortIdx,`
	`  uint32* SignalQualityPtr`
	`)`	

Service ID [hex]	0x33	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	SignalQualityPtr	Pointer to the memory where the signal quality shall be stored.
Return value	Std_ReturnType	`E_OK`: signal quality could be read. `E_NOT_OK`: signal quality could not be read (i.e. no Ethernet transceiver is available for this Ethernet switch port)

▽

△

Description	The function retrieves the signal quality of the link of the indexed Ethernet switch port. If no transceiver is referenced the signal quality shall be set to 0xFFFFFFFF.
Available via	EthSwt.h

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00293] ⌈The function EthSwt_GetPortSignalQuality shall obtain the signal quality by calling the function EthTrcv_GetPhySignalQuality of the referenced Ethernet Transceiver Driver. If the current signal quality is not available, the signal quality shall be set to 0xFFFFFFFF.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00297] ⌈The function EthSwt_GetPortSignalQuality shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetPortSignalQualityApi.⌋*(SRS_BSW_00171)*

8.3.38 EthSwt_GetPortIdentifier

[SWS_EthSwt_91015] Definition of API function EthSwt_GetPortIdentifier ⌈

Service Name	EthSwt_GetPortIdentifier	

Syntax	`Std_ReturnType EthSwt_GetPortIdentifier (`
	`  uint8 SwitchIdx,`
	`  uint8 PortIdx,`
	`  uint32* OrgUniqueIdPtr,`
	`  uint8* ModelNrPtr,`
	`  uint8* RevisionNrPtr`
	`)`	

Service ID [hex]	0x34	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	OrgUniqueIdPtr	Pointer to the memory where the Organizationally Unique Identifier (OUI) shall be stored.
	ModelNrPtr	Pointer to the memory where the Manufacturer's Model Number shall be stored.
	RevisionNrPtr	Pointer to the memory where the Revision Number shall be stored.
Return value	Std_ReturnType	E_OK: organizationally unique identifier of the Ethernet transceiver could be read. E_NOT_OK: organizationally unique identifier of the Ethernet transceiver could not be obtained (i.e. OUI is not available).
Description	This function retrieves the OUI (24 bit) of the indexed Ethernet switch port.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00299] ⌈The function EthSwt_GetPortIdentifier shall return the value of the organizationally unique identifier (OUI 24 bit) of the indexed Ethernet switch port that is connected to the indexed Ethernet switch. It shall set the 8 most significant bits of the OUI to 0xFFxxxxxx. If the Ethernet switch port references an Ethernet transceiver, the function shall obtain the OUI by calling the function EthTrcv_GetPhyIdentifier and set the 8 most significant bits of the OUI to 0x00xxxxxx.⌋(*SRS_Eth_00123*)

[SWS_EthSwt_00394] ⌈If neither the Ethernet switch port nor the Ethernet Transceiver Driver can provide an OUI the function EthSwt_GetPortIdentifier shall return E_NOT_OK.⌋*()*

[SWS_EthSwt_00303] ⌈The function EthSwt_GetPortIdentifier shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetPortIdentifierApi.⌋(*SRS_BSW_00171*)

8.3.39 EthSwt_GetSwitchIdentifier

[SWS_EthSwt_91016] Definition of API function EthSwt_GetSwitchIdentifier ⌈

Service Name	EthSwt_GetSwitchIdentifier	

Syntax	`Std_ReturnType EthSwt_GetSwitchIdentifier (`
	`  uint8 SwitchIdx,`
	`  uint32* OrgUniqueIdPtr`
	`)`	

Service ID [hex]	0x35	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
Parameters (inout)	None	
Parameters (out)	OrgUniqueIdPtr	Pointer to the memory where the Organizationally Unique Identifier shall be stored.
Return value	Std_ReturnType	E_OK: organizationally unique identifier of the Ethernet switch could be read. E_NOT_OK: organizationally unique identifier of the Ethernet switch could not be read (i.e. no OUI is available for this Ethernet switch)
Description	Obtain the Organizationally Unique Identifier that is given by the IEEE of the indexed Ethernet switch. This function shall provide the OUI of Ethernet switch. The OUI has a size of 24 bit. If a ethernet switch can provide the OUI the 8 most significant bits of the OUI shall be set to 0x00xxxxxx. If a Ethernet switch can not provide the OUI the 8 most significant bits of the OUI shall be set to 0xFFxxxxxx.	
Available via	EthSwt.h	

⌋(*SRS_Eth_00123*)

[SWS_EthSwt_00305] ⌈The function EthSwt_GetSwitchIdentifier shall return the value of the organizationally unique identifier of the indexed Ethernet switch.⌋(*SRS_Eth_00123*)

[SWS_EthSwt_00308] ⌈The function EthSwt_GetSwitchIdentifier shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetSwitchIdentifierApi.⌋(*SRS_BSW_00171*)

8.3.40 EthSwt_WritePortMirrorConfiguration

[SWS_EthSwt_91018] Definition of API function EthSwt_WritePortMirrorConfiguration ⌈

Service Name	EthSwt_WritePortMirrorConfiguration	

Syntax	`Std_ReturnType EthSwt_WritePortMirrorConfiguration (`
	`  uint8 MirroredSwitchIdx,`
	`  const EthSwt_PortMirrorCfgType* PortMirrorConfigurationPtr`
	`)`	

Service ID [hex]	0x36	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	MirroredSwitchIdx	Index of the switch within the context of the Ethernet Switch Driver, where the Ethernet switch port is located, that has to be mirrored
	PortMirrorConfiguration Ptr	Pointer of the port configuration, which shall be stored in a shadow buffer in the Ethernet switch driver
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: the port mirror configuration for the indexed Ethernet switch port was written. E_NOT_OK: the port mirror configuration for the indexed Ethernet switch port was not written. (i.e. indexed ethernet switch is not available) ETHSWT_PORT_MIRRORING_CONFIGURATION_NOT_SUPPORTED: port mirroring configuration is not supported by Ethernet switch driver or by the Ethernet switch hardware
Description	Store the given port mirror configuration in a shadow buffer in the Ethernet switch driver for the given MirroredSwitchIdx.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00309] ⌈The function EthSwt_WritePortMirrorConfiguration shall store the port mirror configuration of the given MirroredSwitchIdx in a shadow buffer. The MirroredSwitchIdx shall be used to identify the port mirror configuration within the Ethernet switch driver.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00312] ⌈The function EthSwt_WritePortMirrorConfiguration shall be pre compile time configurable On/Off by the configuration parameter: EthSwtWritePortMirrorConfigurationApi.⌋*(SRS_BSW_00171)*

[SWS_EthSwt_00424] ⌈The function shall return with ETHSWT_PORT_MIRRORING_CONFIGURATION_NOT_SUPPORTED, if the port mirroring configuration is not supported by the Ethernet switch driver or by the Ethernet switch hardware, e.g.:

- the configured mirrored traffic direction (see [SWS_EthSwt_91017] "TrafficDirectionIngressBitMask" and "TrafficDirectionEgressBitMask") for ingress and egress traffic of the same port is not supported
- mirrored ports and capture ports, respectively, are not available within the Ethernet switch driver

⌋(SRS_Eth_00123)

8.3.41 EthSwt_ReadPortMirrorConfiguration

[SWS_EthSwt_91019] Definition of API function EthSwt_ReadPortMirrorConfiguration ⌈

Service Name	EthSwt_ReadPortMirrorConfiguration	

Syntax	`Std_ReturnType EthSwt_ReadPortMirrorConfiguration (`
	`  uint8 MirroredSwitchIdx,`
	`  EthSwt_PortMirrorCfgType* PortMirrorConfigurationPtr`
	`)`	

Service ID [hex]	0x37	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	MirroredSwitchIdx	Index of the Ethernet switch within the context of the Ethernet Switch Driver, where the Ethernet switch ports are located, that have to be mirrored
Parameters (inout)	None	
Parameters (out)	PortMirrorConfiguration Ptr	Pointer to the memory where the port configuration shall be stored.
Return value	Std_ReturnType	E_OK: the port mirror configuration for the indexed Ethernet switch port was red successfully. E_NOT_OK: the port mirror configuration for the indexed Ethernet switch was not read successfully. (i.e. indexed Ethernet switch is not available)
Description	Obtain the port mirror configuration of the given Ethernet switch.	
Available via	EthSwt.h	

⌋(SRS_Eth_00123)

[SWS_EthSwt_00313] ⌈The function EthSwt_ReadPortMirrorConfiguration shall return the port mirror configuration identified by the given MirroredSwitchIdx. If no port mirror configuration is found for the MirroredSwitchIdx, the function shall return E_NOT_OK.⌋(SRS_Eth_00123)

[SWS_EthSwt_00317] ⌈The function EthSwt_ReadPortMirrorConfiguration shall be pre compile time configurable On/Off by the configuration parameter: EthSwtReadPortMirrorConfigurationApi.⌋(SRS_BSW_00171)

8.3.42 EthSwt_DeletePortMirrorConfiguration

[SWS_EthSwt_91034] Definition of API function EthSwt_DeletePortMirrorConfiguration ⌈

Service Name	EthSwt_DeletePortMirrorConfiguration	
Syntax	Std_ReturnType EthSwt_DeletePortMirrorConfiguration (uint8 MirroredSwitchIdx)	
Service ID [hex]	0x4a	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MirroredSwitchIdx. Non reentrant for the same SwitchIdx.	
Parameters (in)	MirroredSwitchIdx	Index of the switch within the context of the Ethernet Switch Driver.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Port mirror configuration was deleted successfully E_NOT_OK: Port mirror configuration was not deleted successfully. (e.g. the port mirroring is enabled)
Description	Delete the stored port mirror configuration of the given MirroredSwitchIdx. If no port mirror configuration was found for the given MirroredSwitchIdx, the return value shall be E_OK.	
Available via	EthSwt.h	

⌋*()*

[SWS_EthSwt_00425] ⌈The function EthSwt_DeletePortMirrorConfiguration shall mark the stored port mirror configuration in the shadow buffer of the given MirroredSwitchIdx as "to be deleted".⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00426] ⌈If a port mirroring for the given MirroredSwitchIdx is enabled, the request to delete the configuration shall be rejected by returning E_NOT_OK. Only those port configurations are allowed to be deleted, where the port mirroring of the given MirroredSwitchIdx is disabled.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00427] ⌈The function EthSwt_DeletePortMirrorConfiguration shall be pre compile time configurable On/Off by the configuration parameter: EthSwtDeletePortMirrorConfigurationApi.⌋*(SRS_BSW_00171)*

8.3.43 EthSwt_GetPortMirrorState

[SWS_EthSwt_91021] Definition of API function EthSwt_GetPortMirrorState ⌈

Service Name	EthSwt_GetPortMirrorState
Syntax	Std_ReturnType EthSwt_GetPortMirrorState (uint8 SwitchIdx, uint8 PortIdx, EthSwt_PortMirrorStateType* PortMirrorStatePtr)
Service ID [hex]	0x38

▽

△

Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	PortMirrorStatePtr	Pointer to the memory where the port mirroring state (either PORT_MIRRORING_ENABLED or PORT_MIRRORING_DISABLED)of the given Ethernet switch port shall be stored.
Return value	Std_ReturnType	`E_OK`: the port mirroring state for the indexed Ethernet switch port returned successfully. `E_NOT_OK`: the port mirror configuration for the indexed Ethernet switch returned not successfully. (i.e. indexed ethernet switch port is not available)
Description	Obtain the current status of the port mirroring for the indexed Ethernet switch port	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00318] ⌈The function `EthSwt_GetPortMirrorState` shall return the port mirroring state of the indexed ethernet switch port.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00322] ⌈The function `EthSwt_GetPortMirrorState` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtGetPortMirrorStateApi`.⌋*(SRS_BSW_00171)*

8.3.44 EthSwt_SetPortMirrorState

[SWS_EthSwt_91022] Definition of API function EthSwt_SetPortMirrorState ⌈

Service Name	EthSwt_SetPortMirrorState	

Syntax	`Std_ReturnType EthSwt_SetPortMirrorState (`
	`  uint8 MirroredSwitchIdx,`
	`  EthSwt_PortMirrorStateType PortMirrorState`
	`)`	

Service ID [hex]	0x39	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	MirroredSwitchIdx	Index of the Ethernet switch within the context of the Ethernet Switch Driver, where the port mirroring configuration is located that has to be enabled and disabled, repectively.
	PortMirrorState	Contain the requested port mirroring state either PORT_MIRRORING_ENABLED or PORT_MIRRORING_DISABLED
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: the requested port mirroring state for the indexed Ethernet switch port was set successfully. `E_NOT_OK`: the requested port mirroring state for the indexed Ethernet switch was not set successfully. (i.e. indexed Ethernet switch is not available, no port mirrior configuration is available)

▽

△

Description	Request to set the given port mirroring state of the port mirror configuration for the given Ethernet switch.
Available via	EthSwt.h

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00323] ⌈The function EthSwt_SetPortMirrorState shall request the given port mirroring state for the port mirroring configuration of the indexed Ethernet switch, and store the requested port mirror state in a shadow buffer.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00327] ⌈The function EthSwt_SetPortMirrorState shall be pre compile time configurable On/Off by the configuration parameter: EthSwtSetPortMirrorStateApi.⌋*(SRS_BSW_00171)*

8.3.45 EthSwt_SetPortTestMode

[SWS_EthSwt_91029] Definition of API function EthSwt_SetPortTestMode ⌈

Service Name	EthSwt_SetPortTestMode	

Syntax	`Std_ReturnType EthSwt_SetPortTestMode (`
	`  uint8 SwitchIdx,`
	`  uint8 PortIdx,`
	`  EthTrcv_PhyTestModeType Mode`
	`)`	

Service ID [hex]	0x3a	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
	Mode	Test mode to be activated
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: the port test mode for the indexed Ethernet switch port was set successfully. E_NOT_OK: the port test mode for the indexed Ethernet switch was not set successfully. (i.e. indexed Ethernet switch port is not available)
Description	Activates a given test mode of the indexed Ethernet switch port.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00328] ⌈The function EthSwt_SetPortTestMode shall forward the call with the given test mode by calling the function EthTrcv_SetPhyTestMode of the referenced Ethernet Transceiver Driver.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00332] ⌈The function EthSwt_SetPortTestMode shall be pre compile time configurable On/Off by the configuration parameter: EthSwtSetPortTestModeApi.⌋*(SRS_BSW_00171)*

8.3.46 EthSwt_SetPortLoopbackMode

[SWS_EthSwt_91023] Definition of API function EthSwt_SetPortLoopbackMode ⌈

Service Name	EthSwt_SetPortLoopbackMode	

Syntax	`Std_ReturnType EthSwt_SetPortLoopbackMode (`
	`uint8 SwitchIdx,`
	`uint8 PortIdx,`
	`EthTrcv_PhyLoopbackModeType Mode`
	`)`	

Service ID [hex]	0x3b	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
	Mode	Loop-back mode to be activated
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: the port mirroring loop-back back mode for the indexed Ethernet switch port was activated successfully. `E_NOT_OK`: the port mirroring loop-back mode for the indexed Ethernet switch port was not activated successfully. (i.e. indexed Ethernet switch port is not available)
Description	Activates a given test loop-back mode of the indexed Ethernet switch port.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00334] ⌈The function `EthSwt_SetPortLoopbackMode` shall forward the call with the given loop-back mode by calling the function `EthTrcv_SetPhyLoopbackMode` of the referenced Ethernet Transceiver Driver.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00338] ⌈The function `EthSwt_SetPortLoopbackMode` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtSetPortLoopbackModeApi`.⌋*(SRS_BSW_00171)*

8.3.47 EthSwt_SetPortTxMode

[SWS_EthSwt_91024] Definition of API function EthSwt_SetPortTxMode ⌈

Service Name	EthSwt_SetPortTxMode

Syntax	`Std_ReturnType EthSwt_SetPortTxMode (`
	`uint8 SwitchIdx,`
	`uint8 PortIdx,`
	`EthTrcv_PhyTxModeType Mode`
	`)`

Service ID [hex]	0x3c
Sync/Async	Synchronous
Reentrancy	Non Reentrant

▽

△

Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
	Mode	Transmission mode to be activated
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: the port Tx mode for the indexed Ethernet switch port was activated successfully. `E_NOT_OK`: the port Tx mode for the indexed Ethernet switch port was not activated successfully. (i.e. indexed Ethernet switch port is not available)
Description	Activates a given transmission mode of the indexed Ethernet switch port.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00340] ⌈The function `EthSwt_SetPortTxMode` shall forward the call with the given transmission mode by calling the function `EthTrcv_SetPhyTxMode` of the referenced Ethernet Transceiver Driver.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00344] ⌈The function `EthSwt_SetPortTxMode` shall be pre compile time configurable On/Off by the configuration parameter: `EthSwtSetPortTxModeApi`.⌋*(SRS_BSW_00171)*

8.3.48 EthSwt_RunPortCableDiagnostic

[SWS_EthSwt_91011] Definition of API function EthSwt_RunPortCableDiagnostic ⌈

Service Name	EthSwt_RunPortCableDiagnostic	

Syntax	`Std_ReturnType EthSwt_RunPortCableDiagnostic (`
	`uint8 SwitchIdxIdx,`
	`uint8 PortIdx`
	`)`	

Service ID [hex]	0x45	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different SwitchIdx and PortIdx. Non reentrant for the same SwitchIdx and Port Idx.	
Parameters (in)	SwitchIdxIdx	Index of the switch within the context of the Ethernet Switch Driver.
	PortIdx	Index of the port at the addressed switch.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The trigger to run the cable diagnostic has been accepted `E_NOT_OK`: The trigger to run the cable diagnostic has not been accepted
Description	Trigger the cable diagnostics of the given Ethernet Switch port (PortIdx) by calling EthTrcv_Run CableDiagnostic of the referenced Ethernet transceiver.	
Available via	EthSwt.h	

⌋*()*

[SWS_EthSwt_00429] ⌈The function EthSwt_RunPortCableDiagnostic shall forward the call by calling EthTrcv_RunCableDiagnostic of the referenced Ethernet Transceiver Driver.⌋*()*

8.3.49 EthSwt_GetPortCableDiagnosticsResult

[SWS_EthSwt_91025] Definition of API function EthSwt_GetPortCableDiagnosticsResult ⌈

Service Name	EthSwt_GetPortCableDiagnosticsResult	

Syntax	`Std_ReturnType EthSwt_GetPortCableDiagnosticsResult (`
	`  uint8 SwitchIdx,`
	`  uint8 PortIdx,`
	`  EthTrcv_CableDiagResultType* ResultPtr`
	`)`	

Service ID [hex]	0x3f	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	ResultPtr	Pointer to the location where the cable diagnostics result shall be stored
Return value	Std_ReturnType	E_OK:the port cable diagnostic result for the indexed Ethernet switch port was obtained successfully. E_NOT_OK: the port cable diagnostic result for the indexed Ethernet switch port was not obtained successfully. (i.e. indexed Ethernet switch port is not available)
Description	Retrieves the cable diagnostics result of the indexed Ethernet switch port respectively the referenced Ethernet Transceiver Driver.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00346] ⌈The function EthSwt_GetPortCableDiagnosticsResult shall obtain the cable diagnostics result by calling the function EthTrcv_GetCableDiagnosticsResult of the referenced Ethernet Transceiver Driver.⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00350] ⌈The function EthSwt_GetPortCableDiagnosticsResult shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetPortCableDiagnosticsResultApi.⌋*(SRS_BSW_00171)*

8.3.50 EthSwt_GetCfgDataRaw

[SWS_EthSwt_91030] Definition of API function EthSwt_GetCfgDataRaw ⌈

Service Name	EthSwt_GetCfgDataRaw	

Syntax	`Std_ReturnType EthSwt_GetCfgDataRaw (`
	`  uint8 SwitchIdx,`
	`  uint32 Offset,`
	`  uint16 Length,`
	`  uint8* BufferPtr`
	`)`	

Service ID [hex]	0x41	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the Ethernet switch within the context of the Ethernet Switch Driver
	Offset	Offset of the Ethernet switch memory from where the reading starts
	Length	Length of data in bytes that shall be copied
Parameters (inout)	None	
Parameters (out)	BufferPtr	Pointer to the location where the data shall be copied
Return value	Std_ReturnType	`E_OK`: the data read was triggered successfully `E_NOT_OK`: the data read was not triggered successfully (i.e. indexed Ethernet switch is not available)
Description	Retrieves the data in memory of the indexed Ethernet switch in variable length	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00403] ⌈The function `EthSwt_GetCfgDataRaw` shall only be available if parameter EthSwtGetCfgRaw is set to TRUE.⌋*(SRS_BSW_00171)*

[SWS_EthSwt_00404] ⌈When calling the function `EthSwt_GetCfgDataRaw`, the function shall check the access to the Ethernet switch driver. If the check fails, the function shall raise the extended production error `ETHSWT_E_ACCESS` and return `E_NOT_OK`, otherwise pass the extended production error `ETHSWT_E_ACCESS` and return `E_OK`.⌋*()*

8.3.51 EthSwt_GetCfgDataInfo

[SWS_EthSwt_91031] Definition of API function EthSwt_GetCfgDataInfo ⌈

Service Name	EthSwt_GetCfgDataInfo

Syntax	`Std_ReturnType EthSwt_GetCfgDataInfo (`
	`  uint8 SwitchIdx,`
	`  uint32* DataSizePtr,`
	`  uint32* DataAdressPtr`
	`)`

Service ID [hex]	0x42
Sync/Async	Synchronous

▽

△

Reentrancy	Reentrant	
Parameters (in)	SwitchIdx	Index of the Ethernet switch within the context of the Ethernet Switch Driver
Parameters (inout)	None	
Parameters (out)	DataSizePtr	Pointer to the location where the total size of the configuration data shall be copied
	DataAdressPtr	Pointer to the location where the start address of the configuration registers shall be copied
Return value	Std_ReturnType	E_OK: the data was obtained successfully E_NOT_OK: the data was not obtained successfully. (i.e. indexed Ethernet switch is not available)
Description	Retrieves the total size of data and the memory start address of the indexed Ethernet Switch.	
Available via	EthSwt.h	

⌋*(SRS_Eth_00123)*

[SWS_EthSwt_00405] ⌈The function EthSwt_GetCfgDataInfo shall only be available if parameter EthSwtGetCfgRaw is set to TRUE.⌋*(SRS_BSW_00171)*

[SWS_EthSwt_00406] ⌈When calling the function EthSwt_GetCfgDataInfo, the function shall check the access to the Ethernet switch driver. If the check fails, the function shall raise the extended production error ETHSWT_E_ACCESS and return E_NOT_OK, otherwise pass the extended production error ETHSWT_E_ACCESS and return E_OK.⌋*()*

8.3.52 EthSwt_PortLinkStateRequest

[SWS_EthSwt_91123] Definition of API function EthSwt_PortLinkStateRequest ⌈

Service Name	EthSwt_PortLinkStateRequest	

Syntax	`Std_ReturnType EthSwt_PortLinkStateRequest (`
	`  uint8 SwitchIdx,`
	`  uint8 PortIdx,`
	`  EthTrcv_LinkStateType PortLinkState`
	`)`	

Service ID [hex]	0x49	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different SwitchIdx and PortIdx. Non reentrant for the same SwitchIdx and Port Idx.	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver.
	PortIdx	Index of the port at the addressed switch.
	PortLinkState	The Ethernet link state of a physical Ethernet connection.
Parameters (inout)	None	
Parameters (out)	None	

▽

△

Return value	Std_ReturnType	E_OK: Request has been accepted and if the function call is in state ETHSWT_STATE_PORTINIT_COMPLETED or ETHSWT_STATE_ACTIVE E_NOT_OK: Request has not been accepted. (e.g. the indexed Ethernet switch port does not reference an EthTrcv)
Description	Request a link state by calling EthTrcv_TransceiverLinkStateRequest with the TrcvIdx of the Ethernet transceiver which is referenced by the Ethernet Switch port (PortIdx).	
Available via	EthSwt.h	

⌋*()*

[SWS_EthSwt_00415] ⌈The function EthSwt_PortLinkStateRequest shall request the given link state for the indexed Ethernet switch port of the switch by calling the EthTrcv_TransceiverLinkStateRequest with the given EthTrcv_LinkStateType. If the EthSwtPort does not reference an EthTrcv, then the function shall return E_NOT_OK.⌋*()*

8.3.53 EthSwt_GetMaxFIFOBufferFillLevel

[SWS_EthSwt_91050] Definition of API function EthSwt_GetMaxFIFOBufferFillLevel ⌈

Service Name	EthSwt_GetMaxFIFOBufferFillLevel	

Syntax	`Std_ReturnType EthSwt_GetMaxFIFOBufferFillLevel (`
	`  uint8 SwitchIdx,`
	`  uint8 SwitchPortIdx,`
	`  uint8 SwitchPortEgressFifoIdx,`
	`  uint32* SwitchPortEgressFifoBufferLevelPtr`
	`)`	

Service ID [hex]	0x48	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SwitchIdx and PortIdx. Non reentrant for the same SwitchIdx and Port Idx.	
Parameters (in)	SwitchIdx	Index of the Ethernet switch within the context of the Ethernet Switch Driver.
	SwitchPortIdx	Index of the Ethernet switch egress port at the addressed Ethernet switch.
	SwitchPortEgressFifoIdx	Index of the egress FIFO of the addressed Ethernet switch port
Parameters (inout)	None	
Parameters (out)	SwitchPortEgressFifo BufferLevelPtr	Pointer to a memory location, where the maximum amount of allocated FIFO buffer (in bytes) since the last read out shall be stored
Return value	Std_ReturnType	E_OK: The FIFO buffer fill level was written to the address pointed to by SwitchPortEgressFifoBufferLevelPtr. E_NOT_OK: The maximal FIFO buffer level could not be obtained
Description	The function retrieves the maximum amount of allocated FIFO buffer of the indexed Ethernet switch egress port. If the Ethernet switch hardware does not support Ethernet switch port based maximal FIFO buffer level, the content of SwitchPortEgressFifoBufferLevelPtr shall be set to 0xFFFFFFFF. This API may be called by e.g. a CDD.	
Available via	EthSwt.h	

⌋*()*

[SWS_EthSwt_00430] ⌈The function EthSwt_GetMaxFIFOBufferFillLevel shall read out the maximum amount of allocated FIFO buffer since the last read out.⌋ *(SRS_Eth_00119)*

[SWS_EthSwt_00431] ⌈When the maximum amount of allocated FIFO buffer is read out, the value shall be reset to 0x00000000 explicitly, if it is not done by the hardware.⌋ *(SRS_Eth_00119)*

[SWS_EthSwt_00432] ⌈The function EthSwt_GetMaxFIFOBufferFillLevel shall be pre compile time configurable On/Off by the configuration parameter: EthSwtGetMaxFIFOBufferFillLevelApi.⌋*(SRS_BSW_00171)*

8.3.54 EthSwt_GetRxMgmtObject

[SWS_EthSwt_91038] Definition of API function EthSwt_GetRxMgmtObject ⌈

Service Name	EthSwt_GetRxMgmtObject	

Syntax	`Std_ReturnType EthSwt_GetRxMgmtObject (`
	`  uint8 CtrlIdx,`
	`  Eth_DataType* DataPtr,`
	`  EthSwt_MgmtObjectType** MgmtObjectPtr`
	`)`	

Service ID [hex]	0x47	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	CtrlIdx	Index of an Ethernet Interface controller
	DataPtr	Ethernet data pointer
Parameters (inout)	None	
Parameters (out)	MgmtObjectPtr	Pointer to the management object.
Return value	Std_ReturnType	E_OK: success E_NOT_OK: management object could not be obtained
Description	Obtains the MgmtObject of the (in this context) unique DataPtr.	
Available via	EthSwt.h	

⌋*()*

8.3.55 EthSwt_GetTxMgmtObject

[SWS_EthSwt_91039] Definition of API function EthSwt_GetTxMgmtObject ⌈

Service Name	EthSwt_GetTxMgmtObject

Syntax	`Std_ReturnType EthSwt_GetTxMgmtObject (`
	`  uint8 CtrlIdx,`
	`  Eth_BufIdxType BufIdx,`
	`  EthSwt_MgmtObjectType** MgmtObjectPtr`
	`)`

Service ID [hex]	0x44

▽

△

Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	CtrlIdx	Index of an Ethernet Interface controller
	BufIdx	Ethernet Rx Buffer index
Parameters (inout)	None	
Parameters (out)	MgmtObjectPtr	Pointer to the management object.
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: management object could not be obtained
Description	Obtains the MgmtObject of the (in this context) unique BufIdx.	
Available via	EthSwt.h	

⌋*()*

8.3.56 EthSwt_MacSecUpdateSecY

[SWS_EthSwt_91124]{DRAFT} **Definition of API function EthSwt_MacSecUpdateSecY** ⌈

Service Name	EthSwt_MacSecUpdateSecY (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecUpdateSecY (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  const Mka_MacSecConfigType* MACsecCfgPtr,`
	`  uint64 TxSci`
	`)`	

Service ID [hex]	0x4d	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	MACsecCfgPtr	Pointer to the structure to configure a MACsec Entity (SecY)
	TxSci	Secure Channel Identifier for the MACsec's Transmission Secure channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Switch to update the SecY/PAC of the PHY with the provided parameters. A Transmission Secure Channel with the provided SCI shall be configured during the first call. A pointer to a MACsec Basic Parameters Configuration file shall be provided to create the Secure Channel. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.57 EthSwt_MacSecUpdateSecYNotification

[SWS_EthSwt_91135]{DRAFT} **Definition of callback function EthSwt_MacSecUpdateSecYNotification** ⌈

Service Name	EthSwt_MacSecUpdateSecYNotification (DRAFT)	

Syntax	`void EthSwt_MacSecUpdateSecYNotification (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr`
	`)`	

Service ID [hex]	0x58	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Callback to notify that EthSwt_MacSecUpdateSecY finished. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.58 EthSwt_MacSecInitRxSc

[SWS_EthSwt_91125]{DRAFT} **Definition of API function EthSwt_MacSecInitRxSc** ⌈

Service Name	EthSwt_MacSecInitRxSc (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecInitRxSc (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint64 Sci`
	`)`	

Service ID [hex]	0x4e	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	Sci	Secure Channel Identifier for the MACsec's Reception Secure channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Switch Driver to configure a Reception Secure Channel for the given Secure Channel Identifier. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.59 EthSwt_MacSecResetRxSc

[SWS_EthSwt_91126]{DRAFT} **Definition of API function EthSwt_MacSecResetRxSc** ⌈

Service Name	EthSwt_MacSecResetRxSc (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecResetRxSc (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint64 Sci`
	`)`	

Service ID [hex]	0x4f	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	Sci	Secure Channel Identifier for the MACsec's Reception Secure channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Switch Driver to reset to default the MACsec values of the Reception Secure Channel for the given Secure Channel Identifier. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.60 EthSwt_MacSecAddTxSa

[SWS_EthSwt_91127]{DRAFT} **Definition of API function EthSwt_MacSecAddTxSa** ⌈

Service Name	EthSwt_MacSecAddTxSa (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecAddTxSa (`
	`  EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint8 An,`
	`  uint64 NextPn,`
	`  uint32 Ssci,`
	`  const Mka_SakKeyPtrType* KeysPtr,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x50	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	An	Association Number to use in the MACsec's transmission secure association

▽

△

	NextPn	Next accepted Packet Number in the MACsec's transmission secure association
	Ssci	Short Secure Channel Identifiert used in the MACsec's transmission secure association
	KeysPtr	Pointer to the SAKs Key (and needed Key information) to use in the MACsec's transmission secure association
	Active	Boolean to enable/disable the MACsec's transmission secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Switch Driver to create a Transmission Secure Association in the Transceiver. The Short Secure Channel Identifier is included to support XPN configurations. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.61 EthSwt_MacSecAddTxSaNotification

[SWS_EthSwt_91136]{DRAFT} **Definition of callback function EthSwt_MacSecAddTxSaNotification** ⌈

Service Name	EthSwt_MacSecAddTxSaNotification (DRAFT)	

Syntax	`void EthSwt_MacSecAddTxSaNotification (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr`
	`)`	

Service ID [hex]	0x59	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Callback to notify that EthSwt_MacSecAddTxSa finished. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.62 EthSwt_MacSecUpdateTxSa

[SWS_EthSwt_91128]{DRAFT} **Definition of API function EthSwt_MacSecUpdateTxSa** ⌈

Service Name	EthSwt_MacSecUpdateTxSa (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecUpdateTxSa (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint8 An,`
	`  uint64 NextPn,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x51	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	An	Association Number to use in the MACsec's transmission secure association
	NextPn	Next accepted Packet Number in the MACsec's transmission secure association
	Active	Boolean to enable/disable the MACsec's transmission secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Switch Driver to update the Transmission Secure Association with the given Packet Number. The Active parameter is included to change the specified AN status. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.63 EthSwt_MacSecDeleteTxSa

[SWS_EthSwt_91129]{DRAFT} **Definition of API function EthSwt_MacSecDeleteTxSa** ⌈

Service Name	EthSwt_MacSecDeleteTxSa (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecDeleteTxSa (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint8 An`
	`)`	

Service ID [hex]	0x52	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.

▽

△

	An	Association Number to use in the MACsec's transmission secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Switch Driver to remove the Transmission Secure Association identified by the provided Association Number. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋()

8.3.64 EthSwt_MacSecAddRxSa

[SWS_EthSwt_91130]{DRAFT} **Definition of API function EthSwt_MacSecAddRxSa** ⌈

Service Name	EthSwt_MacSecAddRxSa (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecAddRxSa (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint8 An,`
	`  uint64 LowestPn,`
	`  uint32 Ssci,`
	`  const Mka_SakKeyPtrType* KeysPtr,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x53	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	An	Association Number to use in the MACsec's reception secure association
	LowestPn	Lowest accepted Packet Number in the MACsec's reception secure association
	Ssci	Short Secure Channel Identifiert used in the MACsec's reception secure association
	KeysPtr	Pointer to the SAKs Key (and needed Key information) to use in the MACsec's reception secure association
	Active	Boolean to enable/disable the MACsec's reception secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Switch Driver to create a Reception Secure Association in the Transceiver. The Short Secure Channel Identifier is included to support XPN configurations. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋()

8.3.65 EthSwt_MacSecAddRxSaNotification

[SWS_EthSwt_91137]{DRAFT} **Definition of callback function EthSwt_MacSecAddRxSaNotification** ⌈

Service Name	EthSwt_MacSecAddRxSaNotification (DRAFT)	

Syntax	`void EthSwt_MacSecAddRxSaNotification (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr`
	`)`	

Service ID [hex]	0x5a	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Callback to notify that EthSwt_MacSecAddRxSa finished. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.66 EthSwt_MacSecUpdateRxSa

[SWS_EthSwt_91131]{DRAFT} **Definition of API function EthSwt_MacSecUpdateRxSa** ⌈

Service Name	EthSwt_MacSecUpdateRxSa (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecUpdateRxSa (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint8 An,`
	`  uint64 LowestPn,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x54	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	An	Association Number to use in the MACsec's reception secure association
	LowestPn	Lowest accepted Packet Number in the MACsec's reception secure association
	Active	Boolean to enable/disable the MACsec's reception secure association
Parameters (inout)	None	
Parameters (out)	None	

▽

△

Return value	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
Description	Request the Ethernet Switch Driver to update the Reception Secure Association with the given Packet Number. The Active parameter is included to change the specified AN status. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋()

8.3.67 EthSwt_MacSecDeleteRxSa

[SWS_EthSwt_91132]{DRAFT} **Definition of API function EthSwt_MacSecDeleteRxSa** ⌈

Service Name	EthSwt_MacSecDeleteRxSa (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecDeleteRxSa (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint8 An`
	`)`	

Service ID [hex]	0x55	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	An	Association Number to use in the MACsec's reception secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
Description	Request the Ethernet Switch Driver to remove the Reception Secure Association identified by the provided Association Number. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋()

8.3.68 EthSwt_MacSecGetTxSaNextPn

[SWS_EthSwt_91133]{DRAFT} **Definition of API function EthSwt_MacSecGetTxSaNextPn** ⌈

Service Name	EthSwt_MacSecGetTxSaNextPn (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecGetTxSaNextPn (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  uint8 An,`
	`  uint64* NextPnPtr`
	`)`	

Service ID [hex]	0x56	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	An	Association Number to use in the MACsec's reception secure association
Parameters (inout)	None	
Parameters (out)	NextPnPtr	Pointer to the Next Packet Number read out from the MACsec Entity (SecY)
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Switch Driver to return the Packet Number that is used for the next packet in the given Transmission Secure Association. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.69 EthSwt_MacSecGetMacSecStats

[SWS_EthSwt_91134]{DRAFT} **Definition of API function EthSwt_MacSecGetMacSecStats** ⌈

Service Name	EthSwt_MacSecGetMacSecStats (DRAFT)	

Syntax	`Std_ReturnType EthSwt_MacSecGetMacSecStats (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr`
	`)`	

Service ID [hex]	0x57	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted

▽

△

Description	Request the Ethernet Switch Driver to provide MACsec statistics. **Tags:** atp.Status=DRAFT
Available via	EthSwt.h

⌋*()*

8.3.70 EthSwt_MacSecGetMacSecStatsNotification

[SWS_EthSwt_91138]{DRAFT} **Definition of callback function EthSwt_MacSecGetMacSecStatsNotification** ⌈

Service Name	EthSwt_MacSecGetMacSecStatsNotification (DRAFT)	

Syntax	`void EthSwt_MacSecGetMacSecStatsNotification (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  const Mka_Stats_SecYType* MacSecStats`
	`)`	

Service ID [hex]	0x5b	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	MacSecStats	Pointer to a structure including the MACsec statistics of an MKA participant
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Callback to notify that EthSwt_MacSecGetMacSecStats finished and provide the requested statistics. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.71 EthSwt_MacSecSetControlledPortEnabled

[SWS_EthSwt_91139]{DRAFT} **Definition of API function EthSwt_MacSecSetControlledPortEnabled** ⌈

Service Name	EthSwt_MacSecSetControlledPortEnabled (DRAFT)

Syntax	`Std_ReturnType EthSwt_MacSecSetControlledPortEnabled (`
	`  const EthSwt_MgmtInfoType* MgmtInfoPtr,`
	`  boolean ControlledPortEnabled`
	`)`

Service ID [hex]	0x5c
Sync/Async	Synchronous

▽

△

Reentrancy	Reentrant for different MgmtInfoPtr, Non reentrant for the same MgmtInfoPtr	
Parameters (in)	MgmtInfoPtr	Pointer to the management information within the context of an Ethernet Switch Driver. SwitchIdx in context of EthIf, PortIdx in context of EthSwt.
	ControlledPortEnabled	Boolean to activate the Controlled Port of the PAE
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests to set the Controlled Port enabled parameter of a PAE. **Tags:** atp.Status=DRAFT	
Available via	EthSwt.h	

⌋*()*

8.3.72 EthSwt_ExtractStreamHandleIdx

[SWS_EthSwt_91043]{DRAFT} **Definition of API function EthSwt_ExtractStreamHandleIdx** ⌈

Service Name	EthSwt_ExtractStreamHandleIdx (draft)	

Syntax	`Std_ReturnType EthSwt_ExtractStreamHandleIdx (`
	`  const Eth_DataType* DataPtr,`
	`  uint16 LenByte,`
	`  uint8* StreamHandleIdxPtr`
	`)`	

Service ID [hex]	0x5d	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	DataPtr	Pointer to payload of received Ethernet frame.
	LenByte	Length (bytes) of the payload in received frame.
Parameters (inout)	None	
Parameters (out)	StreamHandleIdxPtr	Pointer to the StreamHandleIdx extracted from the network packet
Return value	Std_ReturnType	E_OK: The operation was carried out successfully E_NOT_OK: The operation failed to execute
Description	Extracts the StreamHandleIdx from the switch vendor specific part of the network packet header **Tags:** atp.Status=draft	
Available via	EthSwt.h	

⌋*(FO_RS_Fw_00011)*

8.3.73 EthSwt_GetStreamHandleIdxStatistics

[SWS_EthSwt_91042]{DRAFT} **Definition of API function EthSwt_GetStreamHandleIdxStatistics** ⌈

Service Name	EthSwt_GetStreamHandleIdxStatistics (draft)	

Syntax	`void EthSwt_GetStreamHandleIdxStatistics (`
	`  uint8 SwitchIdx`
	`)`	

Service ID [hex]	0x5e	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Requests the bucket counter values from the switch **Tags:** atp.Status=draft	
Available via	EthSwt.h	

⌋*(FO_RS_Fw_00011)*

8.3.74 EthSwt_SetStreamHandleIdxConfiguration

[SWS_EthSwt_91041]{DRAFT} **Definition of API function EthSwt_SetStreamHandleIdxConfiguration** ⌈

Service Name	EthSwt_SetStreamHandleIdxConfiguration (draft)	

Syntax	`void EthSwt_SetStreamHandleIdxConfiguration (`
	`  uint8 SwitchIdx,`
	`  uint8 StreamHandleIdx,`
	`  boolean StreamHandleIdxActivityStatus`
	`)`	

Service ID [hex]	0x5f	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	StreamHandleIdx	Pointer to the StreamHandleIdx for which the status shall be set
	StreamHandleIdxActivity Status	Activity status of the StreamHandleIdx (True = active, False = inactive) to be set
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This function is called by the EthIf module to control the activity status of a StreamHandleIdx in the switch. **Tags:** atp.Status=draft	
Available via	EthSwt.h	

⌋*(FO_RS_Fw_00011)*

8.4 Callback notifications

8.4.1 EthSwtPersistentConfigurationResultCallback

[SWS_EthSwt_00193] Definition of callback function <EthSwtPersistentConfigurationResultCallback> ⌈

Service Name	<EthSwtPersistentConfigurationResultCallback>	

Syntax	`void <EthSwtPersistentConfigurationResultCallback> (`
	`  NvM_RequestResultType JobResult`
	`)`	

Service ID [hex]	0x1b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	JobResult	Covers the job result of the previous processed single block job.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Job end notification of EthSwt_StoreConfiguration or EthSwt_ResetConfiguration	
Available via	EthSwtExternals.h	

⌋*(SRS_Eth_00122, SRS_Eth_00087)*

[SWS_EthSwt_00194] ⌈The callback function <EthSwtPersistentConfigurationResultCallback> shall be called by the `EthSwt_NvmSingleBlockCallback` to inform the caller of `EthSwt_StoreConfiguration` or `EthSwt_ResetConfiguration` about the state of the past calls.⌋*(SRS_Eth_00122, SRS_Eth_00087)*

8.5 Scheduled functions

8.5.1 EthSwt_MainFunction

[SWS_EthSwt_00114] Definition of scheduled function EthSwt_MainFunction ⌈

Service Name	EthSwt_MainFunction

Syntax	`void EthSwt_MainFunction (`
	`  void`
	`)`

Service ID [hex]	0x1c
Description	Service to support asynchronous behavior of API calls
Available via	EthSwt_SchM.h

⌋*(SRS_BSW_00433)*

[SWS_EthSwt_00115] ⌈The `EthSwt_MainFunction` support asynchronous behavior of API calls. This function is directly called by Basic Software Scheduler.⌋*(SRS_BSW_00433)*

8.5.2 EthSwt_BackgroundTask

[SWS_EthSwt_91104] Definition of API function EthSwt_BackgroundTask ⌈

Service Name	EthSwt_BackgroundTask

Syntax	`void EthSwt_BackgroundTask (`
	`  void`
	`)`

Service ID [hex]	0x46
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	The background task should be scheduled as often as possible when no other task runs. It may be used for switch and port initialization in case the EthSwt_Init function needs too much time.
Available via	EthSwt.h

⌋*()*

8.6 Expected interfaces

In this chapter all external interfaces required from other modules are listed.

8.6.1 Mandatory Interfaces

This chapter defines all external interfaces which are required to fulfill the core functionality of the module.

No mandatory Interfaces defined.

8.6.2 Optional Interfaces

This chapter defines all external interfaces which are required to fulfill an optional functionality of the module.

[SWS_EthSwt_00098] Definition of optional interfaces in module EthSwt ⌈

API Function	*Header File*	*Description*
Dem_SetEventStatus	Dem.h	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value. This API will be available only if ({Dem/DemConfigSet/DemEventParameter/DemEventReportingType} == STANDARD_REPORTING)
Det_ReportError	Det.h	Service to report development errors.
Eth_ReadMii	Eth.h	Reads a transceiver register
Eth_WriteMii	Eth.h	Configures a transceiver register or triggers a function offered by the receiver
EthIf_StreamHandleIdxConfiguration (draft)	EthIf_Cbk.h	The function is called by the EthSwtDrv once it has successfully set the StreamHandleIdx activity status in the switch. **Tags:** atp.Status=draft
EthIf_StreamHandleIdxStatistics (draft)	EthIf_Cbk.h	The function is called by the EthSwtDrv once it has successfully retrieved the bucket counter values from the switch. **Tags:** atp.Status=draft
EthTrcv_GetBaudRate	EthTrcv.h	Obtains the baud rate of the indexed transceiver
EthTrcv_GetDuplexMode	EthTrcv.h	Obtains the duplex mode of the indexed transceiver
EthTrcv_GetLinkState	EthTrcv.h	Obtains the link state of the indexed transceiver
EthTrcv_GetTransceiverMode	EthTrcv.h	Obtains the state of the indexed transceiver
EthTrcv_SetTransceiverMode	EthTrcv.h	Enables / disables the indexed transceiver
EthTrcv_StartAutoNegotiation	EthTrcv.h	Restarts the negotiation of the transmission parameters used by the indexed transceiver
NvM_GetErrorStatus	NvM.h	Service to read the block dependent error/status information.
NvM_ReadBlock	NvM.h	Service to copy the data of the NV block to its corresponding RAM block.
NvM_WriteBlock	NvM.h	Service to copy the data of the RAM block to its corresponding NV block.
Spi_AsyncTransmit	Spi.h	Service to transmit data on the SPI bus.
Spi_Cancel	Spi.h	Service cancels the specified on-going sequence transmission.
Spi_ReadIB	Spi.h	Service for reading synchronously one or more data from an IB SPI Handler/Driver Channel specified by parameter.
Spi_SetAsyncMode	Spi.h	Service to set the asynchronous mechanism mode for SPI busses handled asynchronously.
Spi_SetupEB	Spi.h	Service to setup the buffers and the length of data for the EB SPI Handler/Driver Channel specified.
Spi_SyncTransmit	Spi.h	Service to transmit data on the SPI bus
Spi_WriteIB	Spi.h	Service for writing one or more data to an IB SPI Handler/Driver Channel specified by parameter.

⌋*(SRS_Eth_00122, SRS_Eth_00118, SRS_Eth_00119, SRS_Eth_00120, SRS_Eth_00087, SRS_Eth_00125, SRS_BSW_00375)*

[SWS_EthSwt_00192] ⌈The NvM APIs will only be used if the respective block is not configured for `NvM_ReadAll` and `NvM_WriteAll`.⌋*(SRS_Eth_00122)*

8.6.3 Configurable interfaces

In this chapter all interfaces are listed where the target function could be configured.

The names of these kind of interfaces are not fixed because they are configurable.

8.6.3.1 <EthSwtLinkDownCallout>

[SWS_EthSwt_00117] Definition of callback function <EthSwtLinkDownCallout> ⌈

Service Name	<EthSwtLinkDownCallout>	

Syntax	`void <EthSwtLinkDownCallout> (`
	`  uint8 SwitchIdx,`
	`  uint8 PortIdx`
	`)`	

Service ID [hex]	0x19	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Is called, if a link which is configured goes down.	
Available via	EthSwt_Externals.h	

⌋*(SRS_Eth_00119, SRS_Eth_00087)*

[SWS_EthSwt_00118] ⌈The function `<EthSwtLinkDownCallout>` shall be called if a link, which is configured, goes down (link loss). The function provides the Switch index and the Port index, such that the port which went down can be identified.⌋*(SRS_Eth_00119, SRS_Eth_00087)*

8.6.3.2 <EthSwtLinkUpCallout>

[SWS_EthSwt_00203] Definition of callback function <EthSwtLinkUpCallout> ⌈

Service Name	<EthSwtLinkUpCallout>

Syntax	`void <EthSwtLinkUpCallout> (`
	`  uint8 SwitchIdx,`
	`  uint8 PortIdx`
	`)`

Service ID [hex]	0x1a
Sync/Async	Synchronous
Reentrancy	Non Reentrant

▽

△

Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch Driver
	PortIdx	Index of the port at the addressed switch
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Is called, if a link which is configured goes up	
Available via	EthSwt_Externals.h	

⌋*(SRS_Eth_00119, SRS_Eth_00087)*

[SWS_EthSwt_00204] ⌈The function `<EthSwtLinkUpCallout>` shall be called if a link, which is configured, goes up. The function provides the Switch index and the Port index, such that the port which went up can be identified.⌋*(SRS_Eth_00119, SRS_Eth_00087)*

Note: If the hardware cannot signal a link up with an interrupt, the status of the link has to be determined in polling mode by checking the state of the link.

8.6.3.3 <GetCfgDataRawDone>

[SWS_EthSwt_91032] Definition of callback function <GetCfgDataRawDone> ⌈

Service Name	<GetCfgDataRawDone>	

Syntax	`void <GetCfgDataRawDone> (`
	`  uint8 SwitchIdx`
	`)`	

Service ID [hex]	0x43	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SwitchIdx	Index of the Ethernet switch where the Configuration is read.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	The call of the function EthSwt_GetCfgDataRaw() triggers a asynchrony read of a certain memory section of the Ethernet switch driver. If the read is done, the configured callout function <GetCfgDataRawDone> shall be called]	
Available via	EthSwt_Externals.h	

⌋*(SRS_Eth_00123)*

8.7 Service Interfaces

No direct access is necessary from the application layer.

9 Sequence diagrams

The following sequence diagram shows the interaction between the DHCP-Server in the TCP/IP-module and the Ethernet Switch Driver:

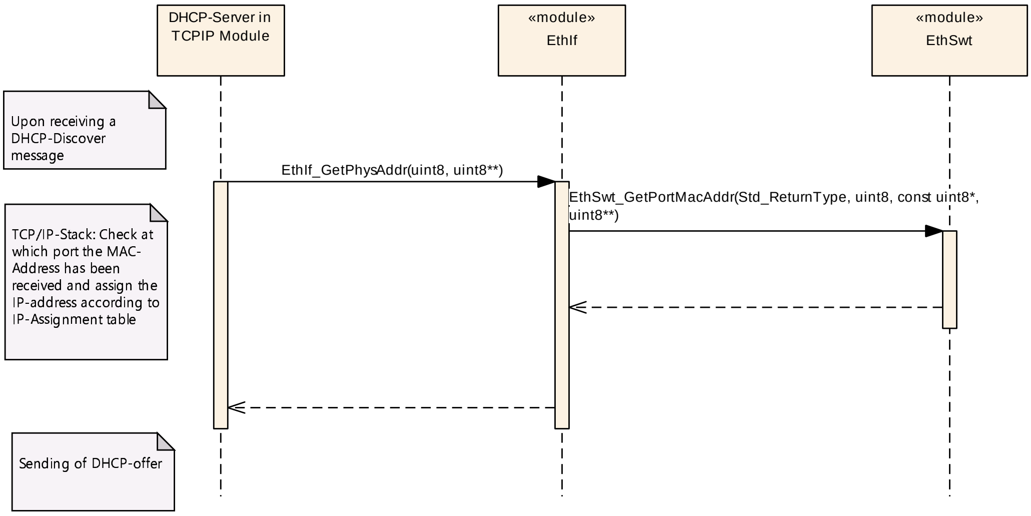

DHCP-Server in TCPIP Module
«module» EthIf
«module» EthSwt
Upon receiving a DHCP-Discover message
EthIf_GetPhysAddr(uint8, uint8**)
EthSwt_GetPortMacAddr(Std_ReturnType, uint8, const uint8*, uint8**)
TCP/IP-Stack: Check at which port the MAC-Address has been received and assign the IP-address according to IP-Assignment table
Sending of DHCP-offer

Figure 9.1

The following sequence diagram shows the interaction between the EthIf, EthSwt and the EthTrcv for API calls to the EthIf:

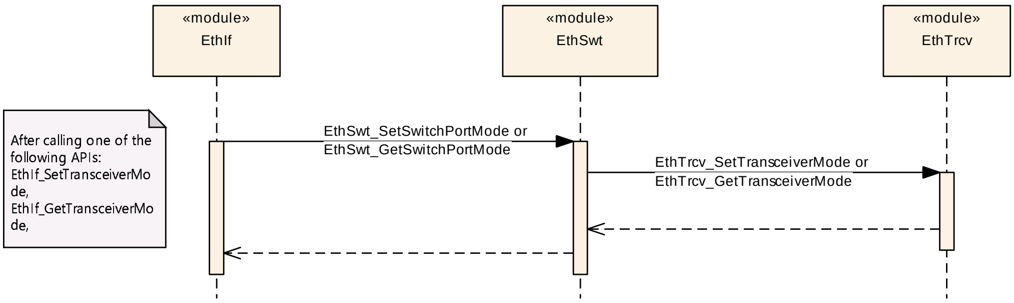

«module» EthIf
«module» EthSwt
«module» EthTrcv
After calling one of the following APIs: EthIf_SetTransceiverMode, EthIf_GetTransceiverMode,
EthSwt_SetSwitchPortMode or EthSwt_GetSwitchPortMode
EthTrcv_SetTransceiverMode or EthTrcv_GetTransceiverMode

Figure 9.2

The following sequence diagram shows the interaction between the EthIf, EthSwt, and the EthTrcv for API calls which are initiated by the EthIf:

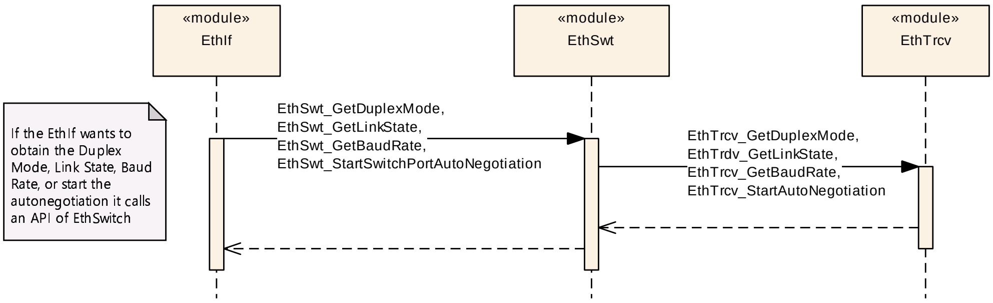

«module» EthIf
«module» EthSwt
«module» EthTrcv
If the EthIf wants to obtain the Duplex Mode, Link State, Baud Rate, or start the autonegotiation it calls an API of EthSwitch
EthSwt_GetDuplexMode, EthSwt_GetLinkState, EthSwt_GetBaudRate, EthSwt_StartSwitchPortAutoNegotiation
EthTrcv_GetDuplexMode, EthTrdv_GetLinkState, EthTrcv_GetBaudRate, EthTrcv_StartAutoNegotiation

Figure 9.3

9.1 Switch Management support

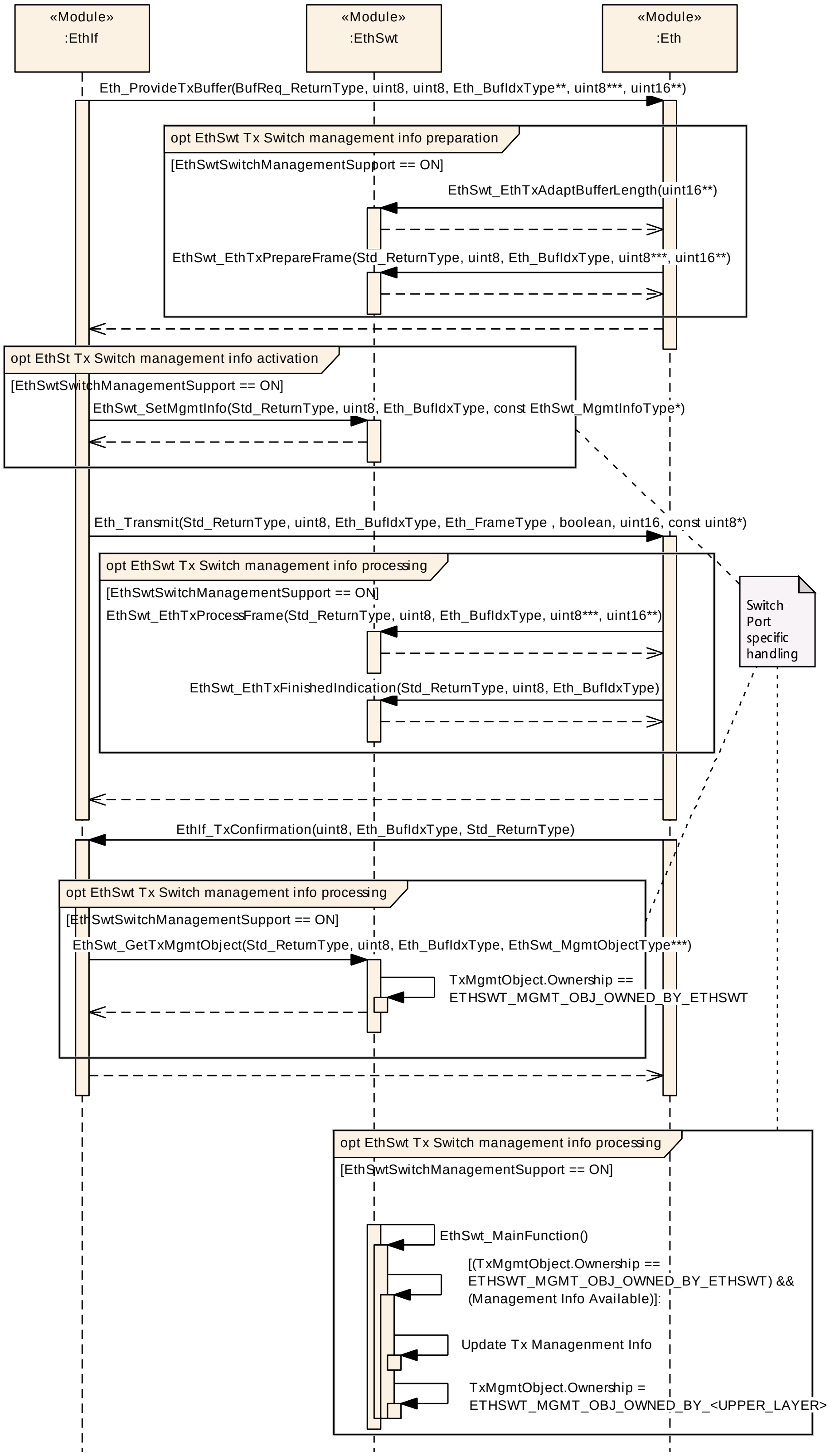

«Module»
:EthIf
«Module»
:EthSwt
«Module»
:Eth
Eth_ProvideTxBuffer(BufReq_ReturnType, uint8, uint8, Eth_BufIdxType**, uint8***, uint16**)
opt EthSwt Tx Switch management info preparation
[EthSwtSwitchManagementSupport == ON]
EthSwt_EthTxAdaptBufferLength(uint16**)
EthSwt_EthTxPrepareFrame(Std_ReturnType, uint8, Eth_BufIdxType, uint8***, uint16**)
opt EthSt Tx Switch management info activation
[EthSwtSwitchManagementSupport == ON]
EthSwt_SetMgmtInfo(Std_ReturnType, uint8, Eth_BufIdxType, const EthSwt_MgmtInfoType*)
Eth_Transmit(Std_ReturnType, uint8, Eth_BufIdxType, Eth_FrameType , boolean, uint16, const uint8*)
opt EthSwt Tx Switch management info processing
[EthSwtSwitchManagementSupport == ON]
EthSwt_EthTxProcessFrame(Std_ReturnType, uint8, Eth_BufIdxType, uint8***, uint16**)
EthSwt_EthTxFinishedIndication(Std_ReturnType, uint8, Eth_BufIdxType)
Switch-
Port
specific
handling
EthIf_TxConfirmation(uint8, Eth_BufIdxType, Std_ReturnType)
opt EthSwt Tx Switch management info processing
[EthSwtSwitchManagementSupport == ON]
EthSwt_GetTxMgmtObject(Std_ReturnType, uint8, Eth_BufIdxType, EthSwt_MgmtObjectType***)
TxMgmtObject.Ownership ==
ETHSWT_MGMT_OBJ_OWNED_BY_ETHSWT
opt EthSwt Tx Switch management info processing
[EthSwtSwitchManagementSupport == ON]
EthSwt_MainFunction()
[(TxMgmtObject.Ownership ==
ETHSWT_MGMT_OBJ_OWNED_BY_ETHSWT) &&
(Management Info Available)]:
Update Tx Managenment Info
TxMgmtObject.Ownership =
ETHSWT_MGMT_OBJ_OWNED_BY_<UPPER_LAYER>

Figure 9.4: Switch Management support for transmission

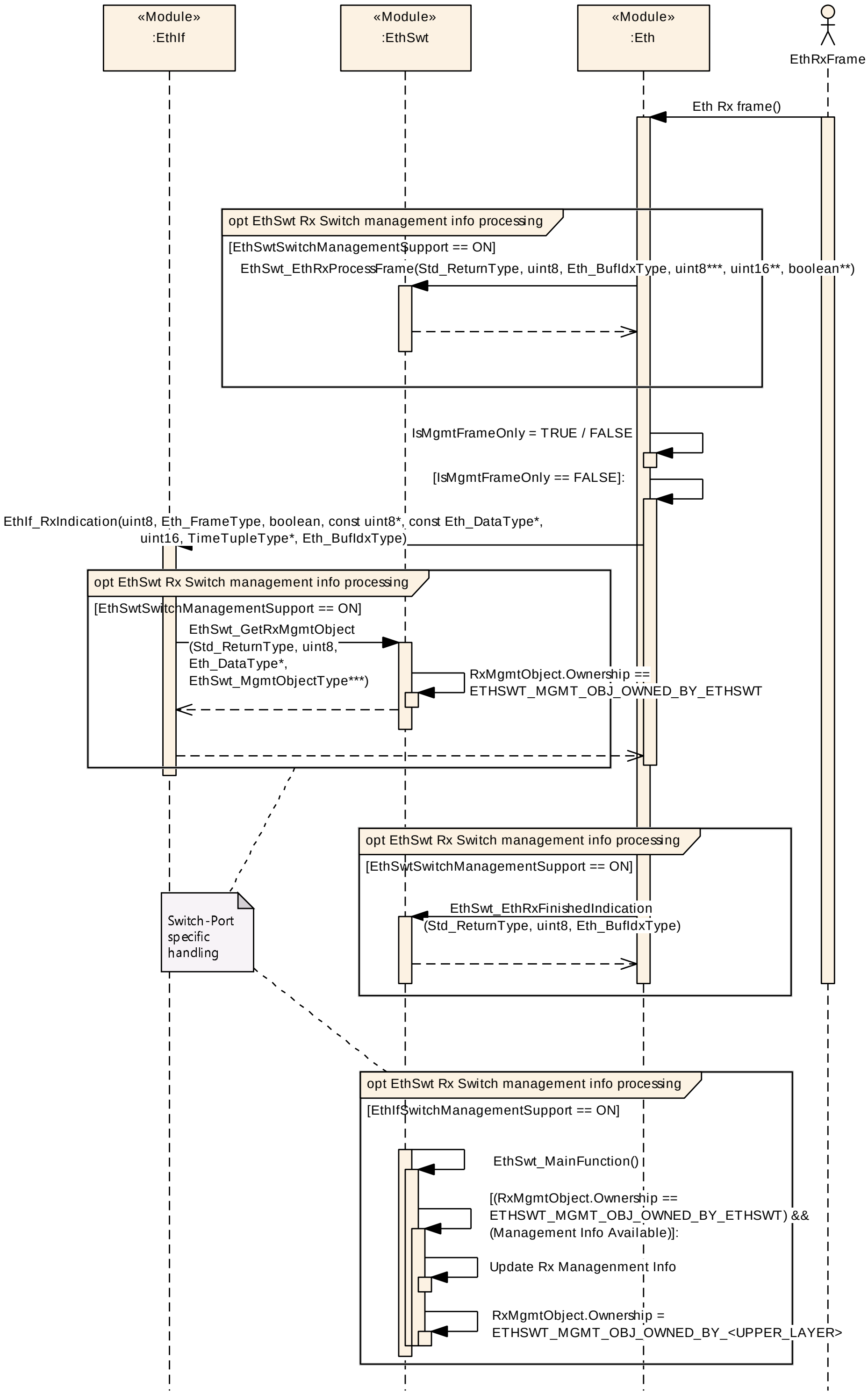

«Module»
:EthIf
«Module»
:EthSwt
«Module»
:Eth
EthRxFrame
Eth Rx frame()
opt EthSwt Rx Switch management info processing
[EthSwtSwitchManagementSupport == ON]
EthSwt_EthRxProcessFrame(Std_ReturnType, uint8, Eth_BufIdxType, uint8***, uint16**, boolean**)
IsMgmtFrameOnly = TRUE / FALSE
[IsMgmtFrameOnly == FALSE]:
EthIf_RxIndication(uint8, Eth_FrameType, boolean, const uint8*, const Eth_DataType*, uint16, TimeTupleType*, Eth_BufIdxType)
opt EthSwt Rx Switch management info processing
[EthSwtSwitchManagementSupport == ON]
EthSwt_GetRxMgmtObject (Std_ReturnType, uint8, Eth_DataType*, EthSwt_MgmtObjectType***)
RxMgmtObject.Ownership == ETHSWT_MGMT_OBJ_OWNED_BY_ETHSWT
opt EthSwt Rx Switch management info processing
[EthSwtSwitchManagementSupport == ON]
EthSwt_EthRxFinishedIndication (Std_ReturnType, uint8, Eth_BufIdxType)
Switch-Port specific handling
opt EthSwt Rx Switch management info processing
[EthIfSwitchManagementSupport == ON]
EthSwt_MainFunction()
[(RxMgmtObject.Ownership == ETHSWT_MGMT_OBJ_OWNED_BY_ETHSWT) && (Management Info Available)]:
Update Rx Managenment Info
RxMgmtObject.Ownership = ETHSWT_MGMT_OBJ_OWNED_BY_<UPPER_LAYER>

Figure 9.5: Management support for reception

10 Configuration specification

section 10.2 specifies the structure (containers) and the parameters of the module EthSwt.

10.1 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe chapter 7 and chapter 8.

[SWS_EthSwt_00414] ⌈The Ethernet Switch Driver module shall reject configurations with partition mappings which are not supported by the implementation.⌋*()*

10.1.1 EthSwt

SWS Item	**[ECUC_EthSwt_00046]**
Module Name	EthSwt
Description	Configuration of the EthSwt (Ethernet Switch Driver) module.
Post-Build Variant Support	true
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtConfig	1..*	Configuration of one Ethernet Switch.
EthSwtGeneral	1	General configuration of Ethernet Switch Driver module.

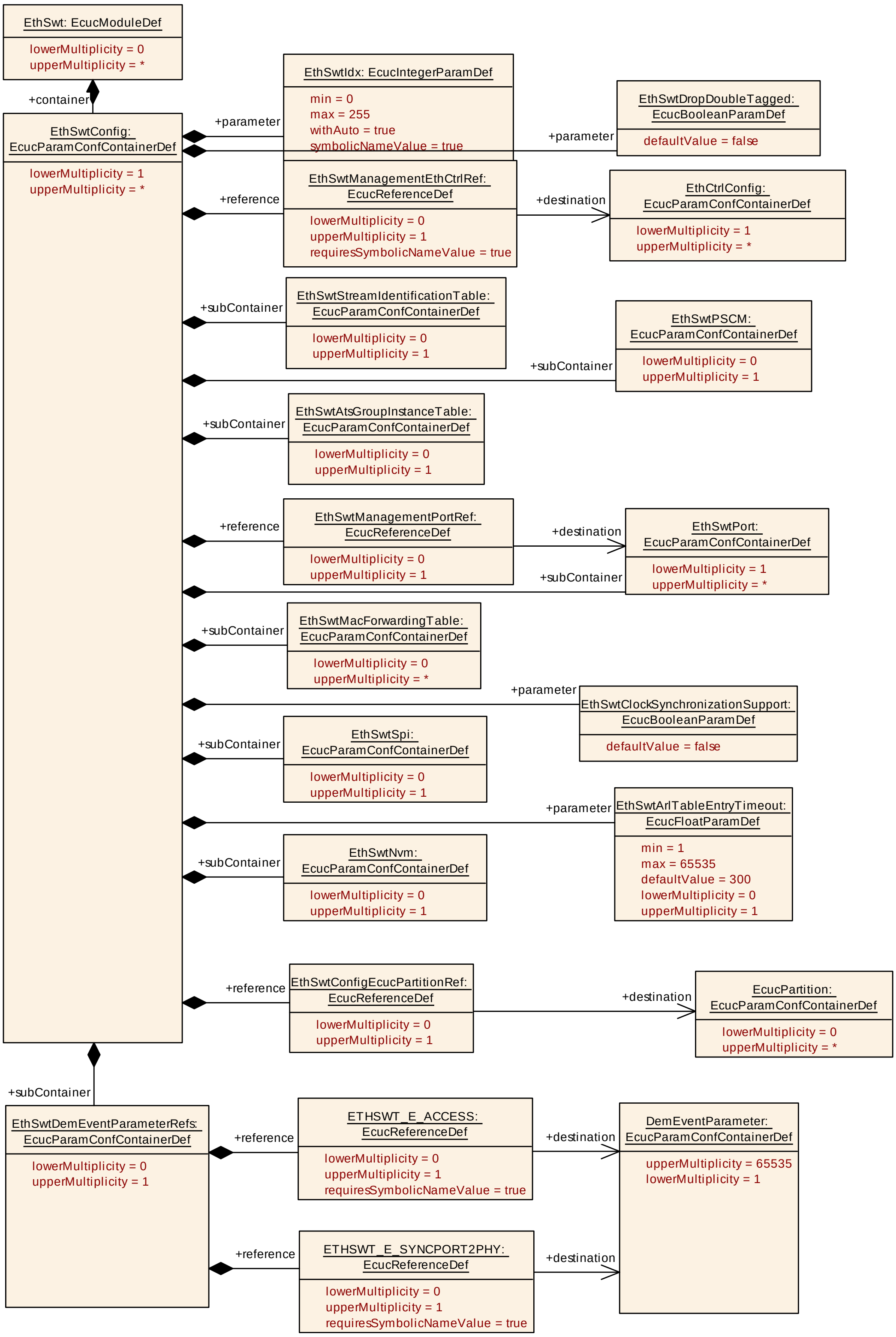


Figure 10.1: EthSwt

10.1.2 EthSwtGeneral

SWS Item	**[ECUC_EthSwt_00003]**
Container Name	EthSwtGeneral
Parent Container	EthSwt
Description	General configuration of Ethernet Switch Driver module.
Configuration Parameters	

SWS Item	**[ECUC_EthSwt_00136]**		
Parameter Name	EthSwtCheckWakeupApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_CheckWakeup API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00133]**		
Parameter Name	EthSwtDeletePortMirrorConfigurationApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_DeletePortMirrorConfiguration API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00002]**		
Parameter Name	EthSwtDevErrorDetect		
Parent Container	EthSwtGeneral		
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.		
Multiplicity	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		

SWS Item	**[ECUC_EthSwt_00135]**		
Parameter Name	EthSwtEnableCableDiagnosticApi		
Parent Container	EthSwtGeneral		
Description	Enable/disable the APIs for cable diagnostic: EthSwt_RunPortCableDiagnostic, EthSwt_GetPortCableDiagnosticsResult		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00055]**		
Parameter Name	EthSwtEnableVlanApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_EnableVLAN API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00052]**		
Parameter Name	EthSwtGetArlTableApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetArlTable API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00121]**
Parameter Name	EthSwtGetBaudRateApi
Parent Container	EthSwtGeneral
Description	Enables / Disables EthSwt_GetBaudRate API
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–

▽

△

Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00124]**		
Parameter Name	EthSwtGetCfgDataRawDone		
Parent Container	EthSwtGeneral		
Description	Defines the function name for <GetCfgDataRawDone>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: The function GetCfgDataRawDone shall only be configured if parameter EthSwtGetCfgRaw is set to TRUE.		

SWS Item	**[ECUC_EthSwt_00123]**		
Parameter Name	EthSwtGetCfgRaw		
Parent Container	EthSwtGeneral		
Description	Disable /Enable support of reading raw data from switch memory		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00053]**
Parameter Name	EthSwtGetCounterValuesApi
Parent Container	EthSwtGeneral
Description	Enables / Disables EthSwt_GetCounterValues API
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Post-Build Variant Value	false

▽

△

Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00122]**		
Parameter Name	EthSwtGetDuplexModeApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetDuplexMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00120]**		
Parameter Name	EthSwtGetLinkStateApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetLinkState API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00061]**		
Parameter Name	EthSwtGetMacLearningModeApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetMacLearningMode API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00131]**		
Parameter Name	EthSwtGetMaxFIFOBufferFillLevelApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetMaxFIFOBufferFillLevel API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00092]**		
Parameter Name	EthSwtGetPortCableDiagnosticsResultApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetPortCableDiagnosticsResult API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00083]**		
Parameter Name	EthSwtGetPortIdentifierApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetPortIdentifier API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00051]**
Parameter Name	EthSwtGetPortMacAddrApi
Parent Container	EthSwtGeneral
Description	Enables / Disables EthSwt_GetPortMacAddr API.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Post-Build Variant Value	false

▽

△

Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00087]**		
Parameter Name	EthSwtGetPortMirrorStateApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetPortMirrorState API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00082]**		
Parameter Name	EthSwtGetPortSignalQualityApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetPortSignalQuality API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00065]**		
Parameter Name	EthSwtGetRxStatsApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetRxStats API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00084]**
Parameter Name	EthSwtGetSwitchIdentifierApi
Parent Container	EthSwtGeneral

▽

△

Description	Enables / Disables EthSwt_GetSwitchIdentifier API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00118]**		
Parameter Name	EthSwtGetSwitchPortModeApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetSwitchPortMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00137]**		
Parameter Name	EthSwtGetSwitchPortWakeupReasonApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetSwitchPortWakeupReason API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00066]**		
Parameter Name	EthSwtGetSwitchRegApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_GetSwitchReg API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

▽

△

Scope / Dependency	scope: local

SWS Item	**[ECUC_EthSwt_00100]**		
Parameter Name	EthSwtGetTxErrorCounterValuesApi		
Parent Container	EthSwtGeneral		
Description	Enables/Disables Eth_GetTxErrorCounterValues API.		
Multiplicity	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		

SWS Item	**[ECUC_EthSwt_00099]**		
Parameter Name	EthSwtGetTxStatsApi		
Parent Container	EthSwtGeneral		
Description	Enables/Disables Eth_GetTxStats API.		
Multiplicity	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		

SWS Item	**[ECUC_EthSwt_00107]**		
Parameter Name	EthSwtGlobalTimeSupportApi		
Parent Container	EthSwtGeneral		
Description	Enables/Disables the Global Time APIs used amongst others by Global Time Synchronization over Ethernet.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00033]**
Parameter Name	EthSwtIndex
Parent Container	EthSwtGeneral
Description	Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.

▽

△

Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00115]**		
Parameter Name	EthSwtLinkDownCallout		
Parent Container	EthSwtGeneral		
Description	Defines the function name for the <EthSwtLinkDownCallout> callout.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00116]**		
Parameter Name	EthSwtLinkUpCallout		
Parent Container	EthSwtGeneral		
Description	Defines the function name for the <EthSwtLinkUpCallout> callout.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00102]**		
Parameter Name	EthSwtLowPowerModeSupport		
Parent Container	EthSwtGeneral		
Description	Disable / Enable support of low power mode.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00071]**		
Parameter Name	EthSwtMainFunctionPeriod		
Parent Container	EthSwtGeneral		
Description	The cycle time of the periodic main function of EthSwt. Defined in seconds .		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

SWS Item	**[ECUC_EthSwt_00108]**		
Parameter Name	EthSwtManagementSupportApi		
Parent Container	EthSwtGeneral		
Description	Enables/Disables the Switch management APIs to support a Switch-port specific communication attribute access.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00062]**		
Parameter Name	EthSwtPersistentConfigurationResult		
Parent Container	EthSwtGeneral		
Description	Enables / Disables the callback API <User>_PersistentConfigurationResult.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00063]**		
Parameter Name	EthSwtPersistentConfigurationResultCallback		
Parent Container	EthSwtGeneral		
Description	Defines the function name for <EthSwtPersistentConfigurationResultCallback>.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00064]**		
Parameter Name	EthSwtPublicCddHeaderFile		
Parent Container	EthSwtGeneral		
Description	Defines header files for callback functions which shall be included in case of CDDs.		
Multiplicity	0..*		
Type	EcucStringParamDef		
Default value	–		
Length	1-32		
Regular Expression	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	

▽

△

	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00086]**		
Parameter Name	EthSwtReadPortMirrorConfigurationApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_ReadPortMirrorConfiguration API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00069]**		
Parameter Name	EthSwtReadTrcvRegisterApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_ReadTrcvRegister API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00049]**		
Parameter Name	EthSwtResetConfigurationApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_ResetConfiguration API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00104]**
Parameter Name	EthSwtSetForwardingModeApi
Parent Container	EthSwtGeneral
Description	Enables /disables EthSwt_SetForwardingMode API.
Multiplicity	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		

SWS Item	**[ECUC_EthSwt_00060]**		
Parameter Name	EthSwtSetMacLearningModeApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_SetMacLearningMode API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00090]**		
Parameter Name	EthSwtSetPortLoopbackModeApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_SetPortLoopbackModeApi API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00088]**		
Parameter Name	EthSwtSetPortMirrorStateApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_SetPortMirrorState API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00089]**		
Parameter Name	EthSwtSetPortTestModeApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_SetPortTestMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00091]**		
Parameter Name	EthSwtSetPortTxModeApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_SetPortTxModeApi API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00117]**		
Parameter Name	EthSwtSetSwitchPortModeApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_SetSwitchPortMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00067]**
Parameter Name	EthSwtSetSwitchRegApi
Parent Container	EthSwtGeneral
Description	Enables / Disables EthSwt_SetSwitchReg API.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Post-Build Variant Value	false

▽

△

Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00119]**		
Parameter Name	EthSwtStartSwitchPortAutoNegotiationApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_StartSwitchPortAutoNegotiation API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00050]**		
Parameter Name	EthSwtStoreConfigurationApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_StoreConfiguration API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00105]**		
Parameter Name	EthSwtVerifyConfigApi		
Parent Container	EthSwtGeneral		
Description	Enables /disables EthSwt_VerifyConfig API.		
Multiplicity	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		

SWS Item	**[ECUC_EthSwt_00031]**		
Parameter Name	EthSwtVersionInfoApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables version info API.		
Multiplicity	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		

SWS Item	**[ECUC_EthSwt_00085]**		
Parameter Name	EthSwtWritePortMirrorConfigurationApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_WritePortMirrorConfiguration API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00070]**		
Parameter Name	EthSwtWriteTrcvRegisterApi		
Parent Container	EthSwtGeneral		
Description	Enables / Disables EthSwt_WriteTrcvRegister API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00129]**
Parameter Name	EthSwtEcucPartitionRef
Parent Container	EthSwtGeneral
Description	Maps the Ethernet switch driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Ethernet switch driver will operate as an independent instance in each of the partitions.
Multiplicity	0..*
Type	Reference to EcucPartition
Post-Build Variant Multiplicity	true

▽

△

Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

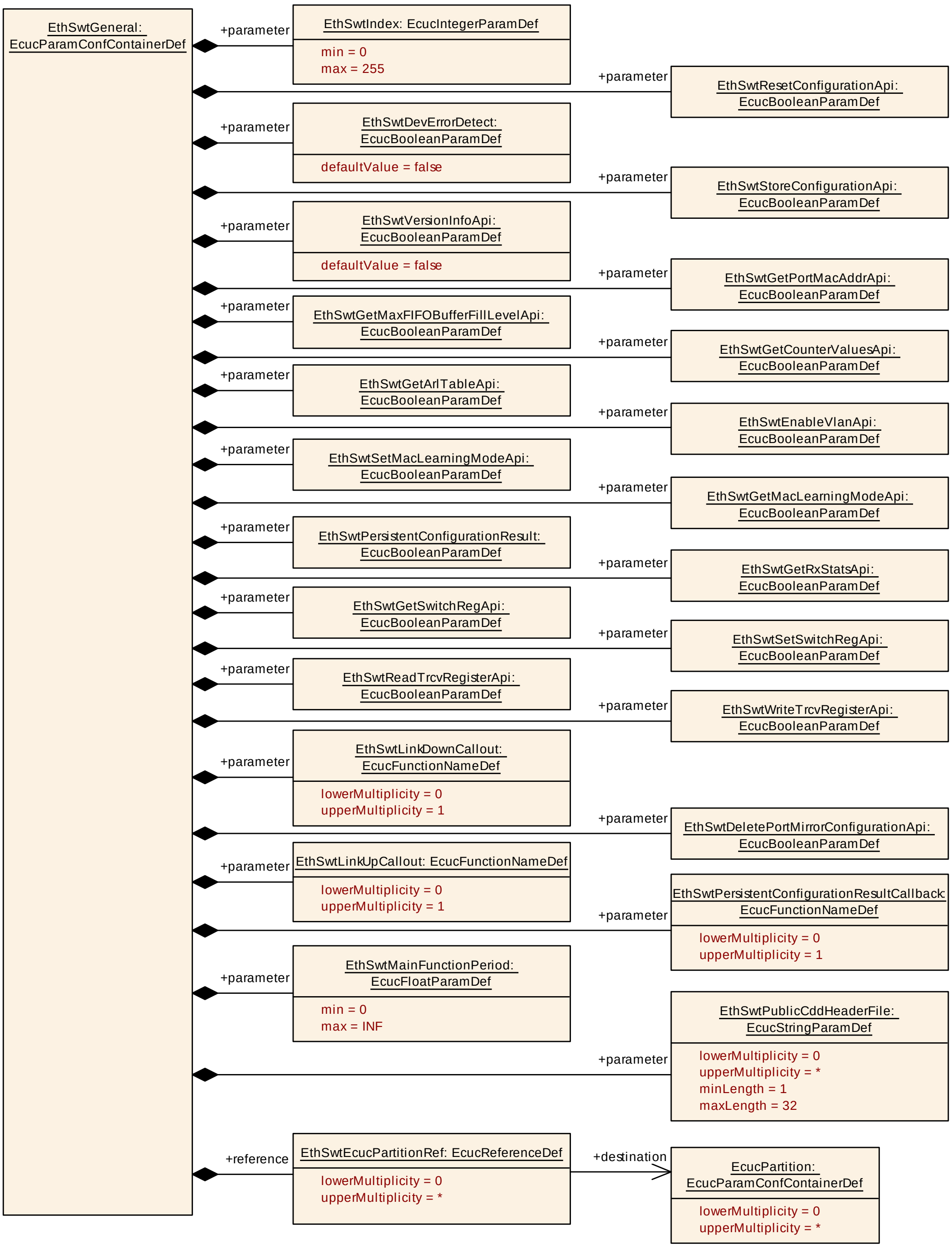

EthSwtGeneral: EcucParamConfContainerDef
+parameter
EthSwtIndex: EcucIntegerParamDef
min = 0
max = 255
+parameter
EthSwtResetConfigurationApi: EcucBooleanParamDef
+parameter
EthSwtDevErrorDetect: EcucBooleanParamDef
defaultValue = false
+parameter
EthSwtStoreConfigurationApi: EcucBooleanParamDef
+parameter
EthSwtVersionInfoApi: EcucBooleanParamDef
defaultValue = false
+parameter
EthSwtGetPortMacAddrApi: EcucBooleanParamDef
+parameter
EthSwtGetMaxFIFOBufferFillLevelApi: EcucBooleanParamDef
+parameter
EthSwtGetCounterValuesApi: EcucBooleanParamDef
+parameter
EthSwtGetArlTableApi: EcucBooleanParamDef
+parameter
EthSwtEnableVlanApi: EcucBooleanParamDef
+parameter
EthSwtSetMacLearningModeApi: EcucBooleanParamDef
+parameter
EthSwtGetMacLearningModeApi: EcucBooleanParamDef
+parameter
EthSwtPersistentConfigurationResult: EcucBooleanParamDef
+parameter
EthSwtGetRxStatsApi: EcucBooleanParamDef
+parameter
EthSwtGetSwitchRegApi: EcucBooleanParamDef
+parameter
EthSwtSetSwitchRegApi: EcucBooleanParamDef
+parameter
EthSwtReadTrcvRegisterApi: EcucBooleanParamDef
+parameter
EthSwtWriteTrcvRegisterApi: EcucBooleanParamDef
+parameter
EthSwtLinkDownCallout: EcucFunctionNameDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtDeletePortMirrorConfigurationApi: EcucBooleanParamDef
+parameter
EthSwtLinkUpCallout: EcucFunctionNameDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtPersistentConfigurationResultCallback: EcucFunctionNameDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtMainFunctionPeriod: EcucFloatParamDef
min = 0
max = INF
+parameter
EthSwtPublicCddHeaderFile: EcucStringParamDef
lowerMultiplicity = 0
upperMultiplicity = *
minLength = 1
maxLength = 32
+reference
EthSwtEcucPartitionRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = *
+destination
EcucPartition: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *

Figure 10.2: EthSwtGeneral (1/2)

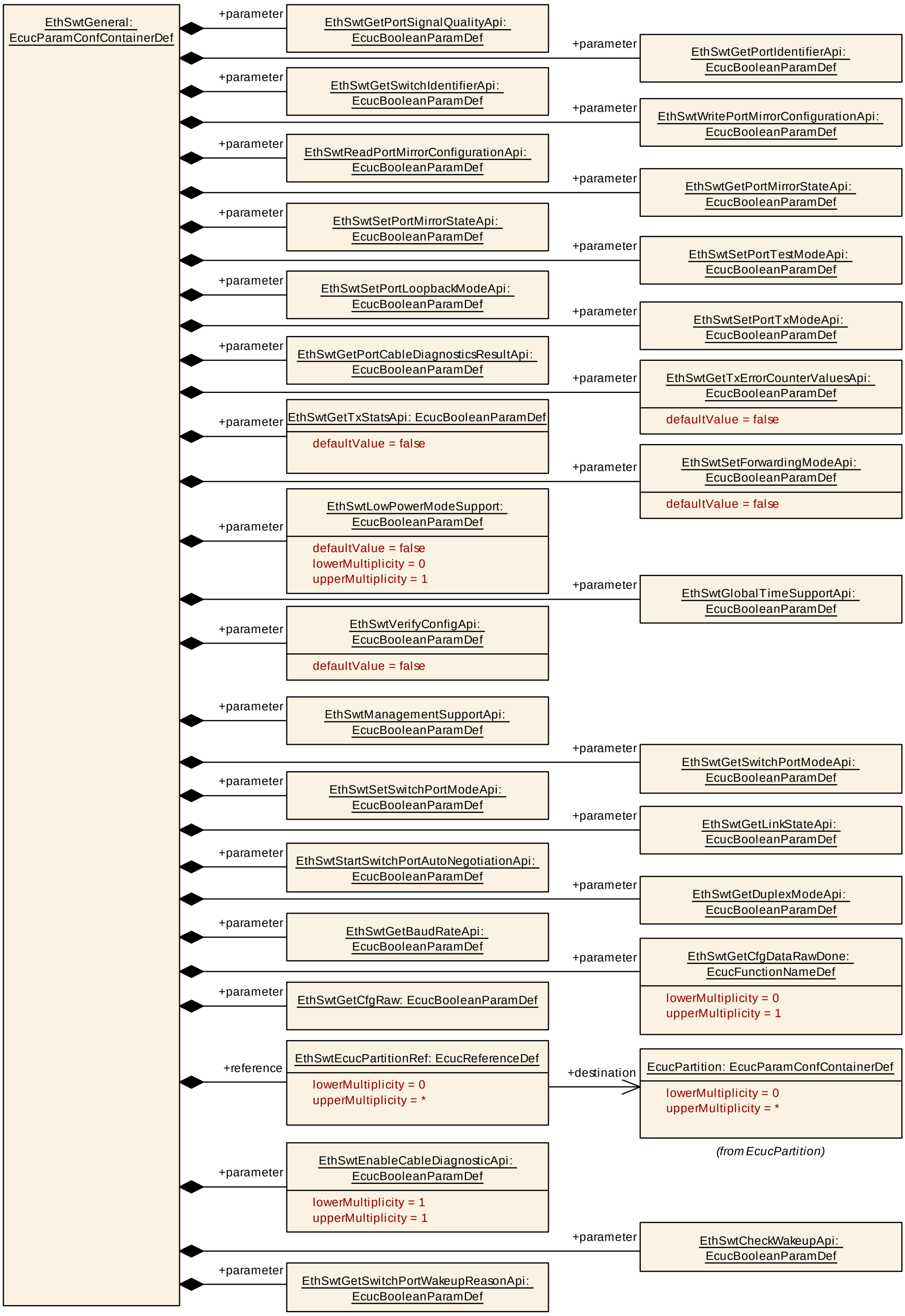

EthSwtGeneral: EcucParamConfContainerDef
+parameter
EthSwtGetPortSignalQualityApi: EcucBooleanParamDef
EthSwtGetPortIdentifierApi: EcucBooleanParamDef
EthSwtGetSwitchIdentifierApi: EcucBooleanParamDef
EthSwtWritePortMirrorConfigurationApi: EcucBooleanParamDef
EthSwtReadPortMirrorConfigurationApi: EcucBooleanParamDef
EthSwtGetPortMirrorStateApi: EcucBooleanParamDef
EthSwtSetPortMirrorStateApi: EcucBooleanParamDef
EthSwtSetPortTestModeApi: EcucBooleanParamDef
EthSwtSetPortLoopbackModeApi: EcucBooleanParamDef
EthSwtSetPortTxModeApi: EcucBooleanParamDef
EthSwtGetPortCableDiagnosticsResultApi: EcucBooleanParamDef
EthSwtGetTxErrorCounterValuesApi: EcucBooleanParamDef
defaultValue = false
EthSwtGetTxStatsApi: EcucBooleanParamDef
defaultValue = false
EthSwtSetForwardingModeApi: EcucBooleanParamDef
defaultValue = false
EthSwtLowPowerModeSupport: EcucBooleanParamDef
defaultValue = false
lowerMultiplicity = 0
upperMultiplicity = 1
EthSwtGlobalTimeSupportApi: EcucBooleanParamDef
EthSwtVerifyConfigApi: EcucBooleanParamDef
defaultValue = false
EthSwtManagementSupportApi: EcucBooleanParamDef
EthSwtGetSwitchPortModeApi: EcucBooleanParamDef
EthSwtSetSwitchPortModeApi: EcucBooleanParamDef
EthSwtGetLinkStateApi: EcucBooleanParamDef
EthSwtStartSwitchPortAutoNegotiationApi: EcucBooleanParamDef
EthSwtGetDuplexModeApi: EcucBooleanParamDef
EthSwtGetBaudRateApi: EcucBooleanParamDef
EthSwtGetCfgDataRawDone: EcucFunctionNameDef
lowerMultiplicity = 0
upperMultiplicity = 1
EthSwtGetCfgRaw: EcucBooleanParamDef
+reference
EthSwtEcucPartitionRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = *
+destination
EcucPartition: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
(from EcucPartition)
EthSwtEnableCableDiagnosticApi: EcucBooleanParamDef
lowerMultiplicity = 1
upperMultiplicity = 1
EthSwtCheckWakeupApi: EcucBooleanParamDef
EthSwtGetSwitchPortWakeupReasonApi: EcucBooleanParamDef

Figure 10.3: EthSwtGeneral (2/2)

10.1.3 EthSwtConfig

SWS Item	**[ECUC_EthSwt_00001]**		
Container Name	EthSwtConfig		
Parent Container	EthSwt		
Description	Configuration of one Ethernet Switch.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00127]**		
Parameter Name	EthSwtArlTableEntryTimeout		
Parent Container	EthSwtConfig		
Description	If present, this parameter specifies the timeout in seconds for removing unused entries from the ARL table of the Ethernet switch. If the parameter is not configured, entries are not removed automatically.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[1 .. 65535]		
Default value	300		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00128]**		
Parameter Name	EthSwtClockSynchronizationSupport		
Parent Container	EthSwtConfig		
Description	This parameter defines, if a Ethernet switch shall enable clock synchronization with another Ethernet switch to which it is connected via uplink port. If this parameter is set to TRUE the clock synchronization between connected Ethernet switches is activated and the clocks of the Ethernet switches are synchronized. If this parameter is set to FALSE the clock synchronization between connected Ethernet switches is deactivated. This parameter shall only be set to TRUE if the Ethernet switch hardware supports clock synchronization.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
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		

SWS Item	**[ECUC_EthSwt_00073]**		
Parameter Name	EthSwtDropDoubleTagged		
Parent Container	EthSwtConfig		
Description	This parameter defines if a switch shall drop double tagged (Q in Q) frames. If this parameter is set to TRUE double tagged frames are dropped at all ports. If this parameter is set to FALSE, then double tagged frames are forwarded. If double tagging is used as a feature, this parameter must be set to FALSE. This parameter shall only be set to TRUE when Switch-HW supports the filtering of double tagged frames as filtering by SW is NOT possible!		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
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		

SWS Item	**[ECUC_EthSwt_00004]**		
Parameter Name	EthSwtIdx		
Parent Container	EthSwtConfig		
Description	Specifies the instance ID of the configured Ethernet Switch.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU withAuto = true		

SWS Item	**[ECUC_EthSwt_00130]**		
Parameter Name	EthSwtConfigEcucPartitionRef		
Parent Container	EthSwtConfig		
Description	Maps the configuration of one single Ethernet switch to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Ethernet switch driver is mapped to.		
Multiplicity	0..1		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants

▽

△

	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_EthSwt_00110]**		
Parameter Name	EthSwtManagementEthCtrlRef		
Parent Container	EthSwtConfig		
Description	Reference to the Ethernet controller connected to the management port where the management frames will be transmitted/received.		
Multiplicity	0..1		
Type	Symbolic name reference to EthCtrlConfig		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00111]**		
Parameter Name	EthSwtManagementPortRef		
Parent Container	EthSwtConfig		
Description	Reference to the port where the management CPU is connected to.		
Multiplicity	0..1		
Type	Reference to EthSwtPort		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtAtsGroupInstanceTable	0..1	Collection of AtsGroupInstanceEntrys. **Tags:** atp.Status=draft
EthSwtDemEventParameterRefs	0..1	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.
EthSwtMacForwardingTable	0..*	Represents a MAC forwarding table.
EthSwtNvm	0..1	Configuration of one Ethernet Switch Nvm usage in case the module requires non volatile memory in the Ecu to store switch configuration.
EthSwtPSCM	0..1	Per stream classification and metering. **Tags:** atp.Status=draft
EthSwtPort	1..*	Configuration of one Ethernet Switch Port.
EthSwtSpi	0..1	Configuration of one Ethernet Switch SPI access (if SPI is used).
EthSwtStreamIdentificationTable	0..1	Configuration of a stream identification table. **Tags:** atp.Status=draft
EthSwtVlanMembership	0..4095	Determines the membership of this Ethernet switch and the referenced ports to the virtual network, i.e. frames with this VID can be received and transmitted via the referenced ports.

10.1.4 EthSwtAtsGroupInstanceTable

SWS Item	**[ECUC_EthSwt_00229]**		
Container Name	EthSwtAtsGroupInstanceTable		
Parent Container	EthSwtConfig		
Description	Collection of AtsGroupInstanceEntrys. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtAtsGroupInstanceEntry	0..*	AtsGroupInstanceEntry. **Tags:** atp.Status=draft

10.1.5 EthSwtAtsGroupInstanceEntry

SWS Item	**[ECUC_EthSwt_00230]**		
Container Name	EthSwtAtsGroupInstanceEntry		
Parent Container	EthSwtAtsGroupInstanceTable		
Description	AtsGroupInstanceEntry. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00195]**		
Parameter Name	EthSwtAtsGroupMaximumResidenceTime		
Parent Container	EthSwtAtsGroupInstanceEntry		
Description	The parameter defines the maximum duration limit for which frames can reside in a bridge in seconds. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
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		

No Included Containers

10.1.6 EthSwtDemEventParameterRefs

SWS Item	**[ECUC_EthSwt_00016]**		
Container Name	EthSwtDemEventParameterRefs		
Parent Container	EthSwtConfig		
Description	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The Event Id is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00006]**		
Parameter Name	ETHSWT_E_ACCESS		
Parent Container	EthSwtDemEventParameterRefs		
Description	Reference to the DemEventParameter which shall be issued when the error "Ethernet Switch Access Failure" has occurred.		
Multiplicity	0..1		
Type	Symbolic name reference to DemEventParameter		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00125]**		
Parameter Name	ETHSWT_E_SYNCPORT2PHY		
Parent Container	EthSwtDemEventParameterRefs		
Description	Reference to the DemEventParameter which shall be issued when the error "Ethernet switch port and the referenced Ethernet transceiver are in contradicting modes" has occurred.		
Multiplicity	0..1		
Type	Symbolic name reference to DemEventParameter		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.1.7 EthSwtMacForwardingTable

SWS Item	**[ECUC_EthSwt_00205]**		
Container Name	EthSwtMacForwardingTable		
Parent Container	EthSwtConfig		
Description	Represents a MAC forwarding table.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE

▽

△

	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00206]**		
Parameter Name	EthSwtPredefinedMacAddress		
Parent Container	EthSwtMacForwardingTable		
Description	Specifies a 48-bit physical addresses (MAC addresses) network byte order, which can be reached via the referenced port and if available via the referenced VLAN . Note that further addresses can be learned during runtime.		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Length	17-17		
Regular Expression	([0-9a-fA-F]\{2}:)\{5}[0-9a-fA-F]\{2}		
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: ECU		

SWS Item	**[ECUC_EthSwt_00207]**		
Parameter Name	EthSwtMacForwardingTablePortRef		
Parent Container	EthSwtMacForwardingTable		
Description	References the ports the MAC shall be assigned to.		
Multiplicity	0..255		
Type	Reference to EthSwtPort		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.1.8 EthSwtNvm

SWS Item	**[ECUC_EthSwt_00043]**
Container Name	EthSwtNvm
Parent Container	EthSwtConfig

▽

△

Description	Configuration of one Ethernet Switch Nvm usage in case the module requires non volatile memory in the Ecu to store switch configuration.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00134]**		
Parameter Name	EthSwtConfigurationNvmBlockDescriptorRef		
Parent Container	EthSwtNvm		
Description	Reference to the Nvm block description in the Nvm module configuration to store e.g. the port mirror configurations		
Multiplicity	1		
Type	Symbolic name reference to NvMBlockDescriptor		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00044]**		
Parameter Name	EthSwtTableNvmBlockDescriptorRef		
Parent Container	EthSwtNvm		
Description	Reference to the Nvm block description in the Nvm module configuration to store e.g. the learned ARL table		
Multiplicity	1		
Type	Symbolic name reference to NvMBlockDescriptor		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

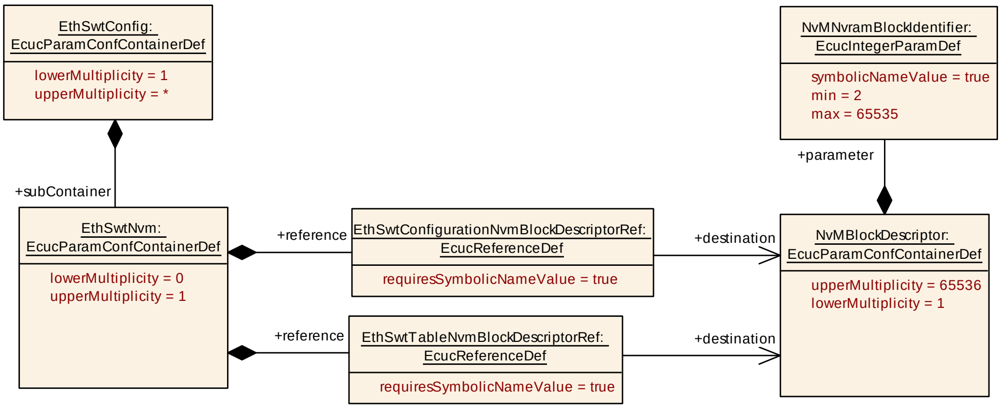

EthSwtConfig: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
EthSwtNvm: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+reference
EthSwtConfigurationNvmBlockDescriptorRef: EcucReferenceDef
requiresSymbolicNameValue = true
+reference
EthSwtTableNvmBlockDescriptorRef: EcucReferenceDef
requiresSymbolicNameValue = true
+destination
+destination
NvMNvramBlockIdentifier: EcucIntegerParamDef
symbolicNameValue = true
min = 2
max = 65535
+parameter
NvMBlockDescriptor: EcucParamConfContainerDef
upperMultiplicity = 65536
lowerMultiplicity = 1

Figure 10.4: **EthSwtNvm**

10.1.9 EthSwtPSCM

SWS Item	[ECUC_EthSwt_00218]		
Container Name	EthSwtPSCM		
Parent Container	EthSwtConfig		
Description	Per stream classification and metering. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtAtsInstanceTable	0..1	EthSwtAtsInstanceTable **Tags:** atp.Status=draft
EthSwtPSFP	0..1	Configuration of Per-stream Filtering and Policing (PSFP). **Tags:** atp.Status=draft

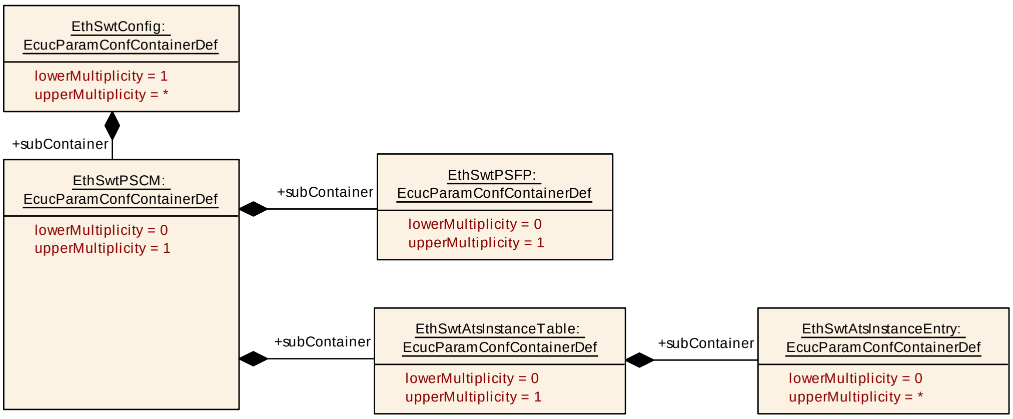


Figure 10.5: EthSwtPSCM

10.1.10 EthSwtAtsInstanceTable

SWS Item	[ECUC_EthSwt_00226]		
Container Name	EthSwtAtsInstanceTable		
Parent Container	EthSwtPSCM		
Description	EthSwtAtsInstanceTable **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	

▽

△

	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtAtsInstanceEntry	0..*	Configuration of an Asynchronous Traffic Scheduler configuration in the scope of the PSFP. **Tags:** atp.Status=draft

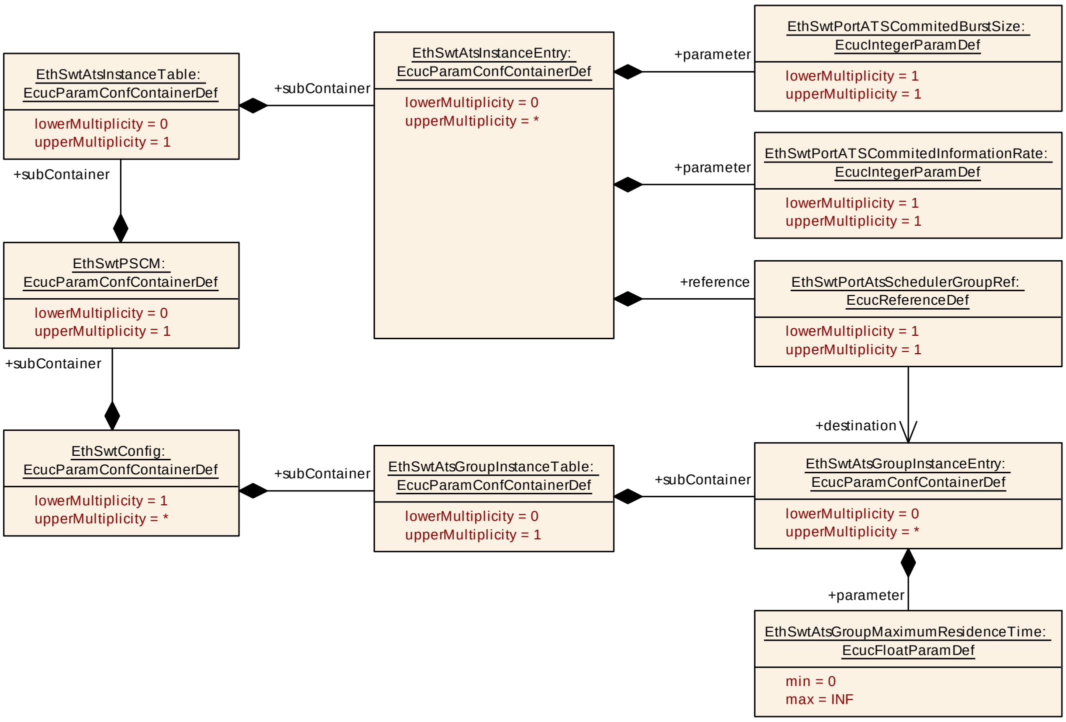


Figure 10.6: `EthSwtPortEgressShaperATS`

10.1.11 EthSwtAtsInstanceEntry

SWS Item	**[ECUC_EthSwt_00228]**		
Container Name	EthSwtAtsInstanceEntry		
Parent Container	EthSwtAtsInstanceTable		
Description	Configuration of an Asynchronous Traffic Scheduler configuration in the scope of the PSFP. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00197]**		
Parameter Name	EthSwtPortATSCommitedBurstSize		
Parent Container	EthSwtAtsInstanceEntry		
Description	Maximum token capacity of the token bucket. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
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		

SWS Item	**[ECUC_EthSwt_00198]**		
Parameter Name	EthSwtPortATSCommitedInformationRate		
Parent Container	EthSwtAtsInstanceEntry		
Description	Defines the rate at which the token bucket is refilled with tokens. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
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		

SWS Item	**[ECUC_EthSwt_00196]**		
Parameter Name	EthSwtPortAtsSchedulerGroupRef		
Parent Container	EthSwtAtsInstanceEntry		
Description	Defines to which ATS scheduler group this ATS scheduler belongs to. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	Reference to EthSwtAtsGroupInstanceEntry		
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		

No Included Containers

10.1.12 EthSwtPSFP

SWS Item	**[ECUC_EthSwt_00154]**		
Container Name	EthSwtPSFP		
Parent Container	EthSwtPSCM		
Description	Configuration of Per-stream Filtering and Policing (PSFP). **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtFilterMaxSduSizeTable	0..1	EthSwtFilterMaxSduSizeTable represents a table of sdu size values, where each value (table entry) could be referenced by a EthSwtStreamFilterEntry. **Tags:** atp.Status=draft
EthSwtFlowMeteringTable	0..1	EthSwtFlowMeteringTable represents a table of flowmeter configurations, where each flowmeter configuration (table entry) could be referenced by a EthSwtStreamFilterEntry. **Tags:** atp.Status=draft
EthSwtStreamFilterTable	0..1	EthSwtStreamFilterTable represents a table of stream filter configurations, where each stream filter configuration (table entry) could reference a EthSwtFilterMaxSduSizeEntry, EthSwt FlowMeteringEntry and EthSwtStreamGateEntry. **Tags:** atp.Status=draft
EthSwtStreamGateTable	0..1	EthSwtStreamGateTable represents a table of stream gate configurations, where each stream gate configuration (table entry) could be referenced by a EthSwtStreamFilterEntry. **Tags:** atp.Status=draft

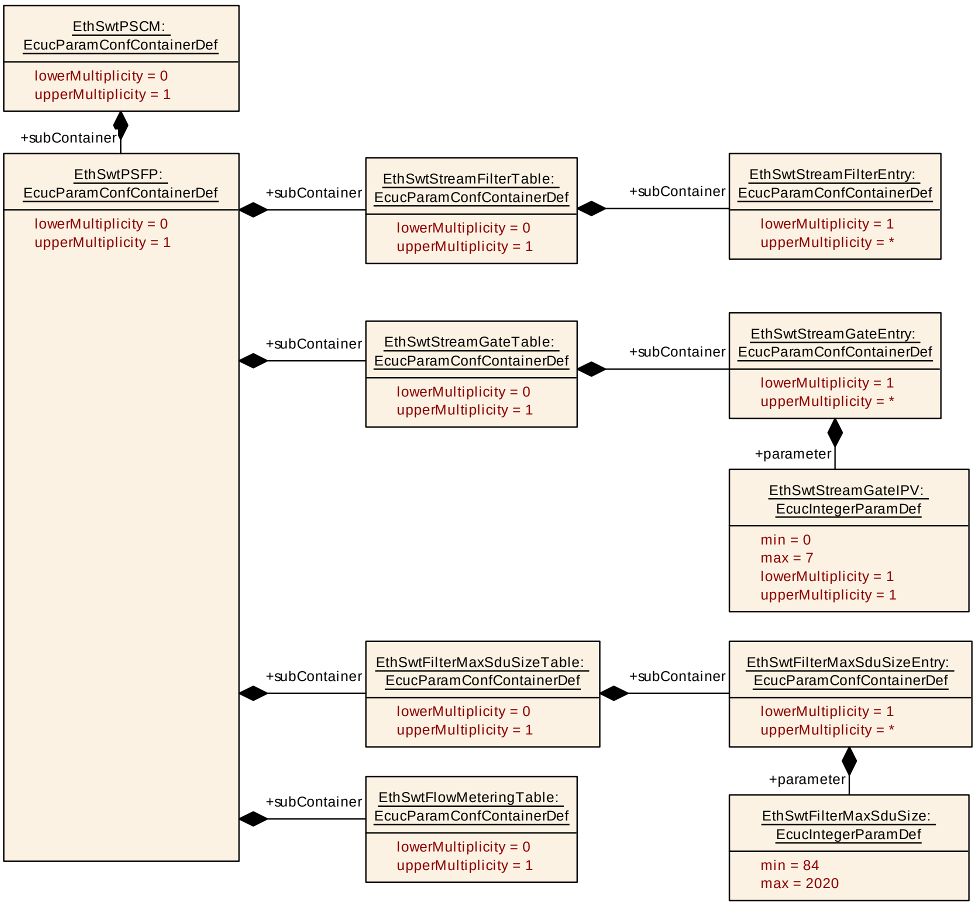


Figure 10.7: EthSwtPSFP

10.1.13 EthSwtFilterMaxSduSizeTable

SWS Item	**[ECUC_EthSwt_00222]**		
Container Name	EthSwtFilterMaxSduSizeTable		
Parent Container	EthSwtPSFP		
Description	EthSwtFilterMaxSduSizeTable represents a table of sdu size values, where each value (table entry) could be referenced by a EthSwtStreamFilterEntry. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtFilterMaxSduSizeEntry	1..*	EthSwtFilterMaxSduSizeEntry defines the maximum SDU size (size of an Ethernet packet) which is acceptable to be processed by the Ethernet switch. The value of EthSwtFilterMaxSduSizeEntry consider the size of the following parts of an Ethernet packet: • Preamble (7 byte) • SFD (start of frame delimiter) (1 byte) • Ethernet frame (Dst MAC,Src MAC, VLAN-tag, TypeField, Payload, CRC Checksum) • Minimum IPG (inter package gap) (12 byte times). **Tags:** atp.Status=draft

10.1.14 EthSwtFilterMaxSduSizeEntry

SWS Item	**[ECUC_EthSwt_00224]**		
Container Name	EthSwtFilterMaxSduSizeEntry		
Parent Container	EthSwtFilterMaxSduSizeTable		
Description	EthSwtFilterMaxSduSizeEntry defines the maximum SDU size (size of an Ethernet packet) which is acceptable to be processed by the Ethernet switch. The value of EthSwtFilterMaxSduSizeEntry consider the size of the following parts of an Ethernet packet: • Preamble (7 byte) • SFD (start of frame delimiter) (1 byte) • Ethernet frame (Dst MAC,Src MAC, VLAN-tag, TypeField, Payload, CRC Checksum) • Minimum IPG (inter package gap) (12 byte times). **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00225]**		
Parameter Name	EthSwtFilterMaxSduSize		
Parent Container	EthSwtFilterMaxSduSizeEntry		
Description	Max Sdu size. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	84 .. 2020		
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		

No Included Containers

10.1.15 EthSwtFlowMeteringTable

SWS Item	**[ECUC_EthSwt_00219]**		
Container Name	EthSwtFlowMeteringTable		
Parent Container	EthSwtPSFP		
Description	EthSwtFlowMeteringTable represents a table of flowmeter configurations, where each flowmeter configuration (table entry) could be referenced by a EthSwtStreamFilterEntry. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtFlowMeteringEntry	1..*	Configuration of a flow metering. **Tags:** atp.Status=draft

10.1.16 EthSwtFlowMeteringEntry

SWS Item	**[ECUC_EthSwt_00157]**		
Container Name	EthSwtFlowMeteringEntry		
Parent Container	EthSwtFlowMeteringTable		
Description	Configuration of a flow metering. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00162]**
Parameter Name	EthSwtFlowMeterCF
Parent Container	EthSwtFlowMeteringEntry

▽

△

Description	Coupling Flag that defines if unused "green" tokens in the first bucket are transferred to the second bucket as "yellow" tokens. Note: this parameter maps to IEEE802.1Q parameter "ieee8021PSFPFlowMeterCF". **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
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		

SWS Item	**[ECUC_EthSwt_00159]**		
Parameter Name	EthSwtFlowMeteringCBS		
Parent Container	EthSwtFlowMeteringEntry		
Description	Committed Burst Size (accepted burst size in green token bucket). Note: this parameter maps to IEEE802.1Q parameter "ieee8021PSFPFlowMeterCBS". **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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		

SWS Item	**[ECUC_EthSwt_00158]**		
Parameter Name	EthSwtFlowMeteringCIR		
Parent Container	EthSwtFlowMeteringEntry		
Description	Committed Information Rate (accepted rate in green token bucket) in bits per second. Note: this parameter maps to IEEE802.1Q parameter "ieee8021PSFPFlowMeterCIR". **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
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		

SWS Item	**[ECUC_EthSwt_00163]**		
Parameter Name	EthSwtFlowMeteringColorMode		
Parent Container	EthSwtFlowMeteringEntry		
Description	Parameter that defines if color-aware or color-blind mode is used. The mode indicates if a color that might be assigned at ingress is used to chose the bucket from which to take tokens; only green and yellow can be assigned; basically, in color-blind mode, all frames are treated like green frames. Note: this parameter maps to IEEE802.1Q parameter "ieee8021PSFPFlowMeterCM". **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ETHSWT_COLOR_AWARE	color aware color mode. **Tags:** atp.Status=draft	
	ETHSWT_COLOR_BLIND	color blind color mode. **Tags:** atp.Status=draft	
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		

SWS Item	**[ECUC_EthSwt_00161]**		
Parameter Name	EthSwtFlowMeteringEBS		
Parent Container	EthSwtFlowMeteringEntry		
Description	Excess burst size (accepted burst size in yellow token bucket). Note: this parameter maps to IEEE802.1Q parameter "ieee8021PSFPFlowMeterEBS". **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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		

SWS Item	**[ECUC_EthSwt_00160]**	
Parameter Name	EthSwtFlowMeteringEIR	
Parent Container	EthSwtFlowMeteringEntry	
Description	Excess Information Rate (accepted rate in yellow token bucket) in bits per second. Note: this parameter maps to IEEE802.1Q parameter "ieee8021PSFPFlowMeterEIR". **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 4294967295	

▽

△

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		

No Included Containers

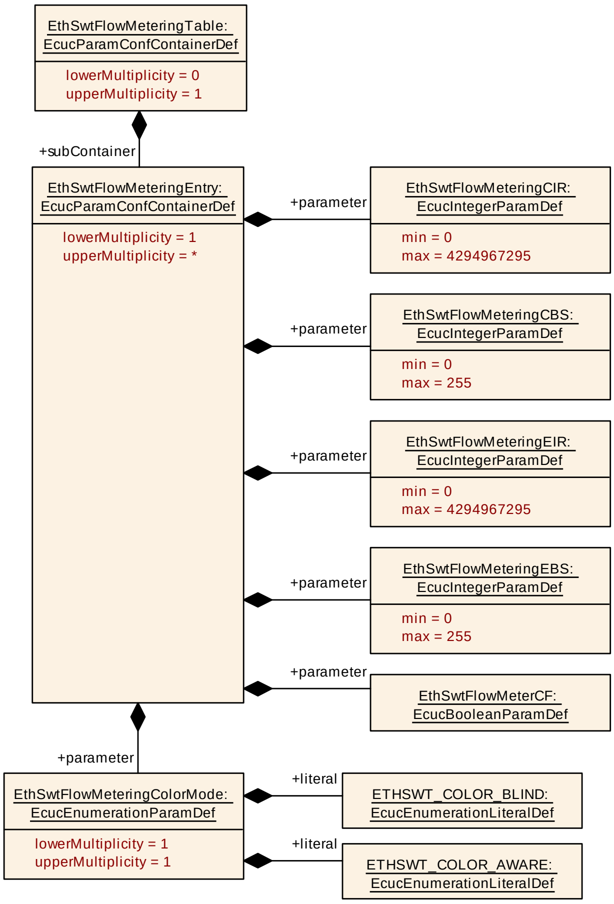

EthSwtFlowMeteringTable: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+subContainer
EthSwtFlowMeteringEntry: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtFlowMeteringCIR: EcucIntegerParamDef
min = 0
max = 4294967295
+parameter
EthSwtFlowMeteringCBS: EcucIntegerParamDef
min = 0
max = 255
+parameter
EthSwtFlowMeteringEIR: EcucIntegerParamDef
min = 0
max = 4294967295
+parameter
EthSwtFlowMeteringEBS: EcucIntegerParamDef
min = 0
max = 255
+parameter
EthSwtFlowMeterCF: EcucBooleanParamDef
+parameter
EthSwtFlowMeteringColorMode: EcucEnumerationParamDef
lowerMultiplicity = 1
upperMultiplicity = 1
+literal
ETHSWT_COLOR_BLIND: EcucEnumerationLiteralDef
+literal
ETHSWT_COLOR_AWARE: EcucEnumerationLiteralDef

Figure 10.8: EthSwtFlowMeteringEntry

10.1.17 EthSwtStreamFilterTable

SWS Item	**[ECUC_EthSwt_00214]**
Container Name	EthSwtStreamFilterTable
Parent Container	EthSwtPSFP

▽

△

Description	EthSwtStreamFilterTable represents a table of stream filter configurations, where each stream filter configuration (table entry) could reference a EthSwtFilterMaxSduSizeEntry, EthSwtFlowMeteringEntry and EthSwtStreamGateEntry. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtStreamFilterEntry	1..*	This container represents a stream filter, where EthSwtStreamFilterPriority and EthSwtAssignedStreamHandle are used to detect a matching Ethernet frame. **Tags:** atp.Status=draft

10.1.18 EthSwtStreamFilterEntry

SWS Item	**[ECUC_EthSwt_00217]**		
Container Name	EthSwtStreamFilterEntry		
Parent Container	EthSwtStreamFilterTable		
Description	This container represents a stream filter, where EthSwtStreamFilterPriority and EthSwtAssignedStreamHandle are used to detect a matching Ethernet frame. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00216]**		
Parameter Name	EthSwtStreamFilterEntryPosition		
Parent Container	EthSwtStreamFilterEntry		
Description	Specifies the position as unique ID within an ordered list of EthSwtStreamFilterEntrys. The ordered list shall start with 0 and continue as linear list with no gaps. Note: The list is processed in ascending order. The instance of EthSwtStreamFilterEntry with position 0 is processed first. **Tags:** atp.Status=draft		
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: ECU		

SWS Item	**[ECUC_EthSwt_00215]**		
Parameter Name	EthSwtStreamFilterPriority		
Parent Container	EthSwtStreamFilterEntry		
Description	This parameter represents an assigned priority of this stream filter, which is used to detect a matching Ethernet frame. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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		

SWS Item	**[ECUC_EthSwt_00227]**		
Parameter Name	EthSwtAtsInstanceEntryRef		
Parent Container	EthSwtStreamFilterEntry		
Description	Reference to an entry of an ATS table, where the entry represents a configuration for asynchronous traffic shaping. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Reference to EthSwtAtsInstanceEntry		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00223]**
Parameter Name	EthSwtFilterMaxSduSizeRef
Parent Container	EthSwtStreamFilterEntry
Description	Reference to an entry of a max-sdu-size table, where the entry represents a particular value. **Tags:** atp.Status=draft
Multiplicity	0..1
Type	Reference to EthSwtFilterMaxSduSizeEntry
Post-Build Variant Multiplicity	true
Post-Build Variant Value	true

▽

△

Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00221]**		
Parameter Name	EthSwtFlowMeteringEntryRef		
Parent Container	EthSwtStreamFilterEntry		
Description	Reference to an entry of a flow metering table, where the entry represents a configuration for flow metering. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Reference to EthSwtFlowMeteringEntry		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00213]**		
Parameter Name	EthSwtStreamGateEntryRef		
Parent Container	EthSwtStreamFilterEntry		
Description	Reference to an entry of a gate table, where the entry represents a configuration for a gate. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Reference to EthSwtStreamGateEntry		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
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		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtAssignedStreamHandle	1	This container represents an assigned stream handle id of this stream filter, which is used to detect a matching Ethernet frame. The EthSwtAssignedStreamHandle could represent a list of assigned stream handle ids. Additionally it could set a wildcard flag (EthSwtStreamHandleId Wildcard), where any assigned stream handle id carried by an Ethernet frame would match to this stream filter. **Tags:** atp.Status=draft

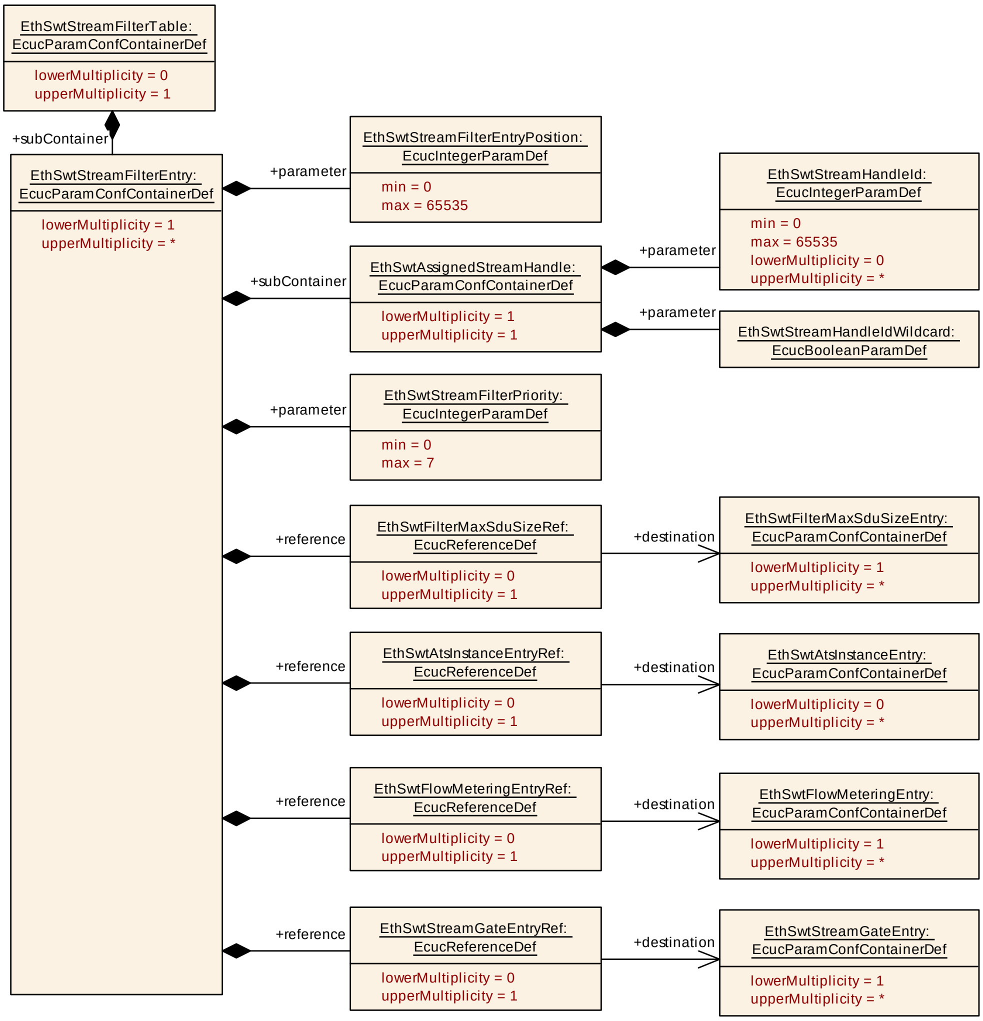

EthSwtStreamFilterTable: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+subContainer
EthSwtStreamFilterEntry: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtStreamFilterEntryPosition: EcucIntegerParamDef
min = 0
max = 65535
+subContainer
EthSwtAssignedStreamHandle: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = 1
+parameter
EthSwtStreamHandleId: EcucIntegerParamDef
min = 0
max = 65535
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
EthSwtStreamHandleIdWildcard: EcucBooleanParamDef
+parameter
EthSwtStreamFilterPriority: EcucIntegerParamDef
min = 0
max = 7
+reference
EthSwtFilterMaxSduSizeRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
+destination
EthSwtFilterMaxSduSizeEntry: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+reference
EthSwtAtsInstanceEntryRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
+destination
EthSwtAtsInstanceEntry: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+reference
EthSwtFlowMeteringEntryRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
+destination
EthSwtFlowMeteringEntry: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+reference
EthSwtStreamGateEntryRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
+destination
EthSwtStreamGateEntry: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *

Figure 10.9: EthSwtStreamFilterEntry

10.1.19 EthSwtAssignedStreamHandle

SWS Item	**[ECUC_EthSwt_00231]**
Container Name	EthSwtAssignedStreamHandle
Parent Container	EthSwtStreamFilterEntry
Description	This container represents an assigned stream handle id of this stream filter, which is used to detect a matching Ethernet frame. The EthSwtAssignedStreamHandle could represent a list of assigned stream handle ids. Additionally it could set a wildcard flag (EthSwtStreamHandleIdWildcard), where any assigned stream handle id carried by an Ethernet frame would match to this stream filter. **Tags:** atp.Status=draft
Configuration Parameters	

SWS Item	**[ECUC_EthSwt_00210]**		
Parameter Name	EthSwtStreamHandleId		
Parent Container	EthSwtAssignedStreamHandle		
Description	Assigned stream handle id of this stream filter, which is used for evaluation of a matching Ethernet frame. **Tags:** atp.Status=draft		
Multiplicity	0..*		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00209]**		
Parameter Name	EthSwtStreamHandleIdWildcard		
Parent Container	EthSwtAssignedStreamHandle		
Description	Defines whether this EthSwtAssignedStreamHandle includes the wildcard. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
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		

No Included Containers

10.1.20 EthSwtStreamGateTable

SWS Item	**[ECUC_EthSwt_00212]**		
Container Name	EthSwtStreamGateTable		
Parent Container	EthSwtPSFP		
Description	EthSwtStreamGateTable represents a table of stream gate configurations, where each stream gate configuration (table entry) could be referenced by a EthSwtStreamFilter Entry. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtStreamGateEntry	1..*	Configuration of a stream gate. **Tags:** atp.Status=draft

10.1.21 EthSwtStreamGateEntry

SWS Item	**[ECUC_EthSwt_00155]**		
Container Name	EthSwtStreamGateEntry		
Parent Container	EthSwtStreamGateTable		
Description	Configuration of a stream gate. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00156]**	
Parameter Name	EthSwtStreamGateIPV	
Parent Container	EthSwtStreamGateEntry	
Description	Internal Priority Value (IPV), a priority value that determines the assigned traffic class. Note: Only the least 3 significant bits shall be configured. The remaining bits shall be ignored. **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 7	
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		

No Included Containers

10.1.22 EthSwtPort

SWS Item	**[ECUC_EthSwt_00005]**		
Container Name	EthSwtPort		
Parent Container	EthSwtConfig		
Description	Configuration of one Ethernet Switch Port.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00013]**		
Parameter Name	EthSwtPortIdx		
Parent Container	EthSwtPort		
Description	Specifies the instance ID of the configured Ethernet Switch Port.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU withAuto = true		

SWS Item	**[ECUC_EthSwt_00114]**
Parameter Name	EthSwtPortMacLayerSpeed	
Parent Container	EthSwtPort	
Description	Defines the baud rate of the MAC layer.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	ETH_MAC_LAYER_SPEED_100M	–
	ETH_MAC_LAYER_SPEED_10G	–
	ETH_MAC_LAYER_SPEED_10M	–

▽

△

	ETH_MAC_LAYER_SPEED_1G	–	
	ETH_MAC_LAYER_SPEED_2500M	–	
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		

SWS Item	**[ECUC_EthSwt_00113]**		
Parameter Name	EthSwtPortMacLayerSubType		
Parent Container	EthSwtPort		
Description	Defines the MAC layer subtype of this EthSwtPort.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	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	
	STANDARD	standard media-independent interface	
	UNIVERSAL_SERIAL	Universal low-power and low pin-count serial 8b/10b-coded media-independent interface	
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		

SWS Item	**[ECUC_EthSwt_00072]**
Parameter Name	EthSwtPortMacLayerType
Parent Container	EthSwtPort
Description	Defines the MAC layer type of this EthSwtPort.
Multiplicity	0..1
Type	EcucEnumerationParamDef

▽

△

Range	ETHSWT_PORT_MAC_LAYER_TYPE_XGMII	MAC layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)	
	ETHSWT_PORT_MAC_LAYER_TYPE_XMII	MAC layer interface (data) bandwith class 100Mbit/s (e.g. MII, RMII, RvMII, SMII, RvMII)	
	ETHSWT_PORT_MAC_LAYER_TYPE_XXGMII	MAC layer interface (data) bandwith class 10Gbit/s	
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		

SWS Item	**[ECUC_EthSwt_00054]**		
Parameter Name	EthSwtPortPhysicalLayerType		
Parent Container	EthSwtPort		
Description	Defines the physical layer type of this EthSwtPort.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ETHSWT_PORT_1000BASE_T	physical layer interface 1000BASE-T (1Gbit/s, 4 pairs). Used for consumer electronic.	
	ETHSWT_PORT_1000BASE_T1	physical layer interface 1000BASE-T1 (1Gbit/s, 1 pair). Used for automotive.	
	ETHSWT_PORT_100BASE_T1	physical layer interface 100BASE-T1 (100Mbit/s, 1 pair). Used for automotive.	
	ETHSWT_PORT_100BASE_TX	physical layer interface 100BASE-TX (100Mbit/s, 2 pairs). Used for consumer electronic.	
	ETHSWT_PORT_10BASE_T1S	physical layer interface 10BASE-T1S (10Mbit/s, 1 pair). Used for automotive.	
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 dependency: If a EthSwtPort has an EthSwtPortPhysicalLayerType then EthSwtPort shall reference an EthTrcv.		

SWS Item	**[ECUC_EthSwt_00101]**		
Parameter Name	EthSwtPortRole		
Parent Container	EthSwtPort		
Description	Set a special role of the Ethernet switch port. It is either a host port or a up link port. If not configured it is a standard port.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ETHSWT_HOST_PORT	The hostPort is connected to an ECU (host ecu). The host ECU controls the connected Coupling Element (e.g. Ethernet switch).	
	ETHSWT_UP_LINK_PORT	A CouplingPort can be connected to another CouplingPort of a CouplingElement located on the same ECU (CouplingElement.ecuInstance) using the CouplingPortConnection. This is used to model a cascaded switch.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: One Ethernet switch shall have either exactly one host port or at least one up link port. In case of having a host port also multiple up link port can exist. A master switch shall be connected by one host port with the host ecu. A slave switch shall be connected to a master switch by one up link port.		

SWS Item	**[ECUC_EthSwt_00112]**		
Parameter Name	EthSwtPortTimeStampSupport		
Parent Container	EthSwtPort		
Description	Enables/Disables the Switch-port specific timestamping.		
Multiplicity	1		
Type	EcucBooleanParamDef		
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 dependency: EthSwtPortTimeStampSupport can only be set to TRUE, * if (EthSwt ClockSynchronizationSupport is FALSE) OR * if ((EthSwtClockSynchronizationSupport is TRUE) AND (EthSwtPortRole is NOT ETHSWT_UP_LINK_PORT))		

SWS Item	**[ECUC_EthSwt_00041]**
Parameter Name	EthSwtPortTrcvRef
Parent Container	EthSwtPort
Description	Reference to the Ethernet transceiver driver this EthSwtPort is connected with.
Multiplicity	0..1

▽

△

Type	Symbolic name reference to EthTrcvConfig		
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
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU dependency: If EthSwtPortPhysicalLayerType is defined, then EthSwtPortTrcvRef holds the reference to the corresponding EthTrcv.		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtPortEgress	1	Configuration of one Ethernet Switch Port Egress behavior.
EthSwtPortIngress	1	Configuration of one Ethernet Switch Port ingress behavior.

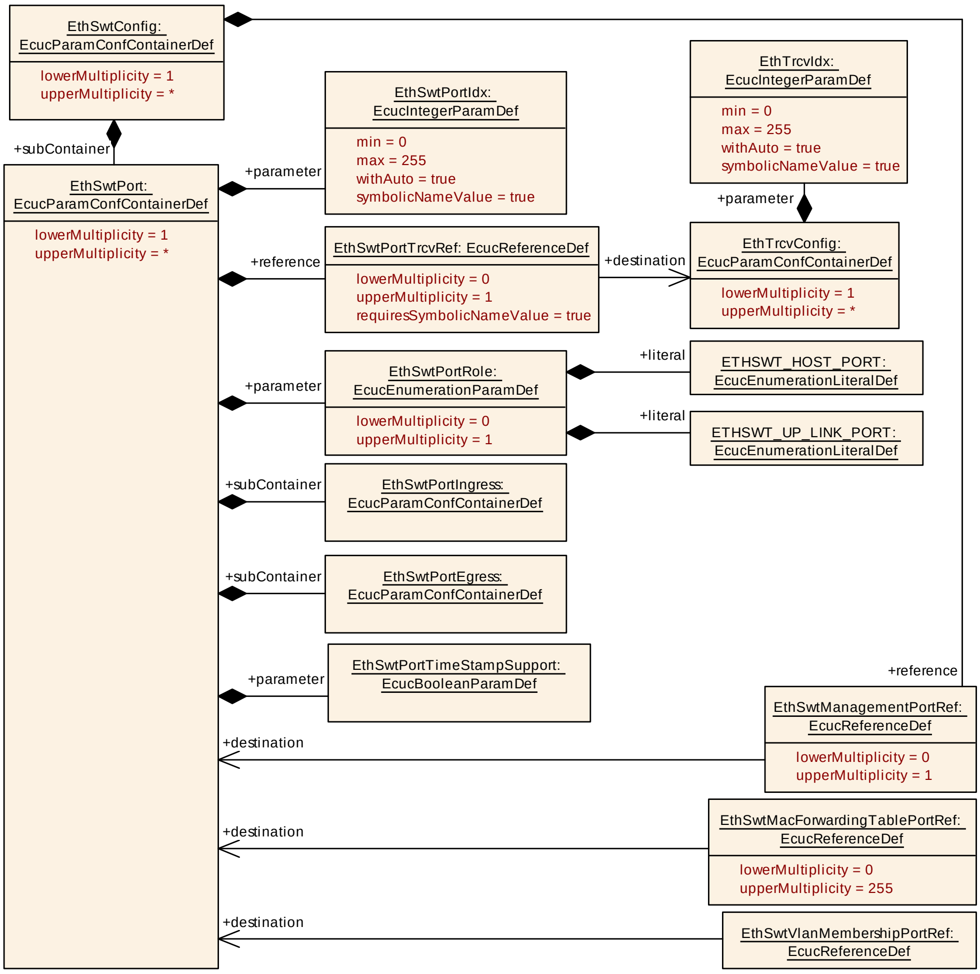

EthSwtConfig: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
EthSwtPort: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtPortIdx: EcucIntegerParamDef
min = 0
max = 255
withAuto = true
symbolicNameValue = true
EthTrcvIdx: EcucIntegerParamDef
min = 0
max = 255
withAuto = true
symbolicNameValue = true
+parameter
+reference
EthSwtPortTrcvRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
requiresSymbolicNameValue = true
+destination
EthTrcvConfig: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtPortRole: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+literal
ETHSWT_HOST_PORT: EcucEnumerationLiteralDef
+literal
ETHSWT_UP_LINK_PORT: EcucEnumerationLiteralDef
+subContainer
EthSwtPortIngress: EcucParamConfContainerDef
+subContainer
EthSwtPortEgress: EcucParamConfContainerDef
+parameter
EthSwtPortTimeStampSupport: EcucBooleanParamDef
+reference
EthSwtManagementPortRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
+destination
EthSwtMacForwardingTablePortRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 255
+destination
EthSwtVlanMembershipPortRef: EcucReferenceDef
+destination

Figure 10.10: EthSwtPort (1/2)

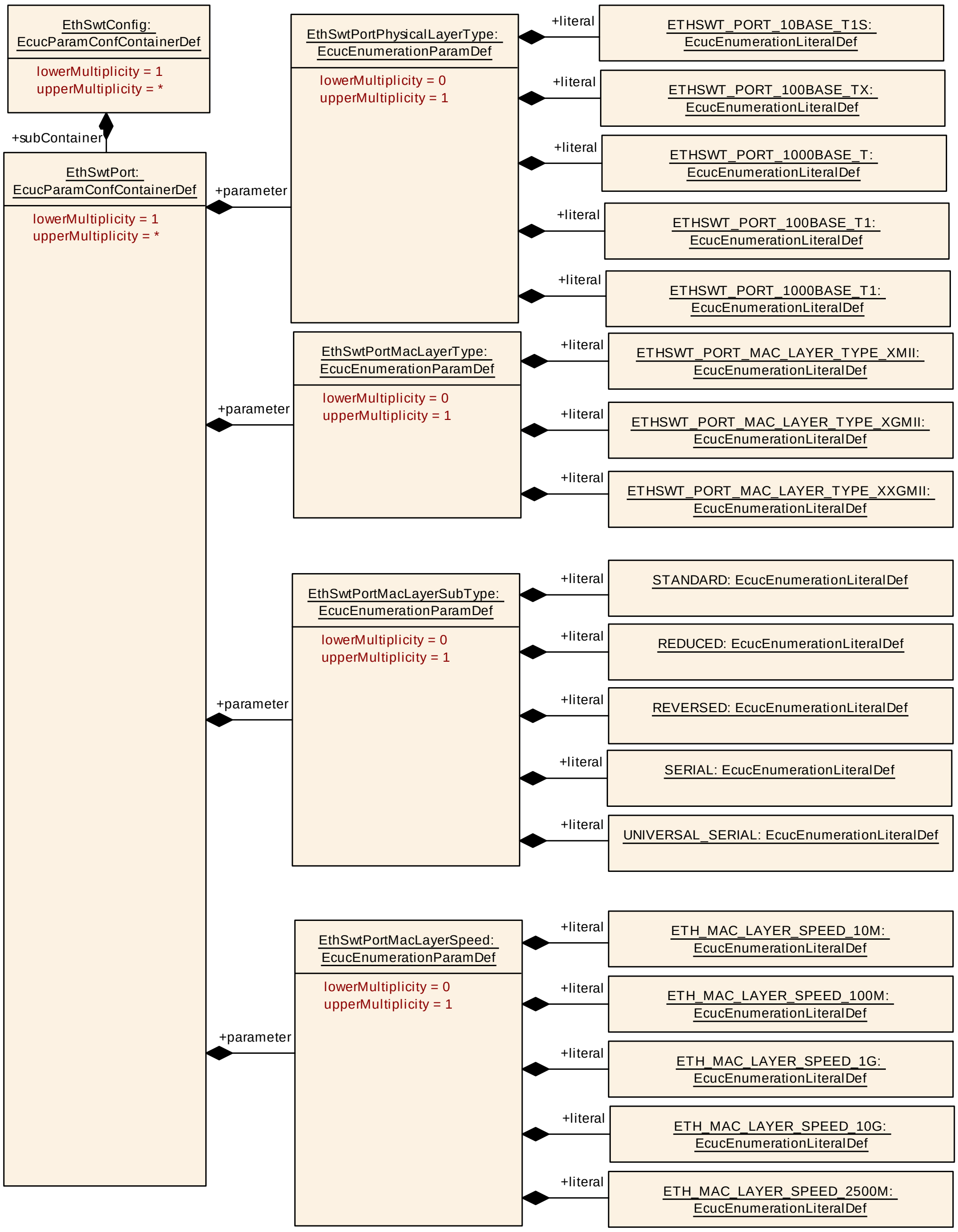

EthSwtConfig: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
EthSwtPort: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtPortPhysicalLayerType: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+literal
ETHSWT_PORT_10BASE_T1S: EcucEnumerationLiteralDef
ETHSWT_PORT_100BASE_TX: EcucEnumerationLiteralDef
ETHSWT_PORT_1000BASE_T: EcucEnumerationLiteralDef
ETHSWT_PORT_100BASE_T1: EcucEnumerationLiteralDef
ETHSWT_PORT_1000BASE_T1: EcucEnumerationLiteralDef
EthSwtPortMacLayerType: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
ETHSWT_PORT_MAC_LAYER_TYPE_XMII: EcucEnumerationLiteralDef
ETHSWT_PORT_MAC_LAYER_TYPE_XGMII: EcucEnumerationLiteralDef
ETHSWT_PORT_MAC_LAYER_TYPE_XXGMII: EcucEnumerationLiteralDef
EthSwtPortMacLayerSubType: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
STANDARD: EcucEnumerationLiteralDef
REDUCED: EcucEnumerationLiteralDef
REVERSED: EcucEnumerationLiteralDef
SERIAL: EcucEnumerationLiteralDef
UNIVERSAL_SERIAL: EcucEnumerationLiteralDef
EthSwtPortMacLayerSpeed: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
ETH_MAC_LAYER_SPEED_10M: EcucEnumerationLiteralDef
ETH_MAC_LAYER_SPEED_100M: EcucEnumerationLiteralDef
ETH_MAC_LAYER_SPEED_1G: EcucEnumerationLiteralDef
ETH_MAC_LAYER_SPEED_10G: EcucEnumerationLiteralDef
ETH_MAC_LAYER_SPEED_2500M: EcucEnumerationLiteralDef

Figure 10.11: EthSwtPort (2/2)

Please note that the functional behavior of the ingress and egress port of a switch is implemented in hardware in the switch devices (see [13]). Thus, the configuration of EthSwtPort and described in the following has to be written to the switch device or is related to the switch configuration.

10.1.23 EthSwtPortEgress

SWS Item	[ECUC_EthSwt_00007]
Container Name	EthSwtPortEgress
Parent Container	EthSwtPort
Description	Configuration of one Ethernet Switch Port Egress behavior.
Configuration Parameters	

SWS Item	[ECUC_EthSwt_00008]		
Parameter Name	EthSwtPortEgressLastSchedulerRef		
Parent Container	EthSwtPortEgress		
Description	Reference to the port scheduler which is the last in the egress port structure.		
Multiplicity	1		
Type	Reference to EthSwtPortEgressScheduler		
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		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
EthSwtPortEgressScheduler	1..*	Represents a Scheduler in the egress port.
EthSwtPortFifo	1..*	Represents a Fifo in the egress port. **Tags:** atp.Status=obsolete
EthSwtPortQueue	1..8	Represents a Queue at the egress port. **Tags:** atp.Status=draft
EthSwtPortShaper	0..*	Represents a Shaper in the egress port. **Tags:** atp.Status=obsolete

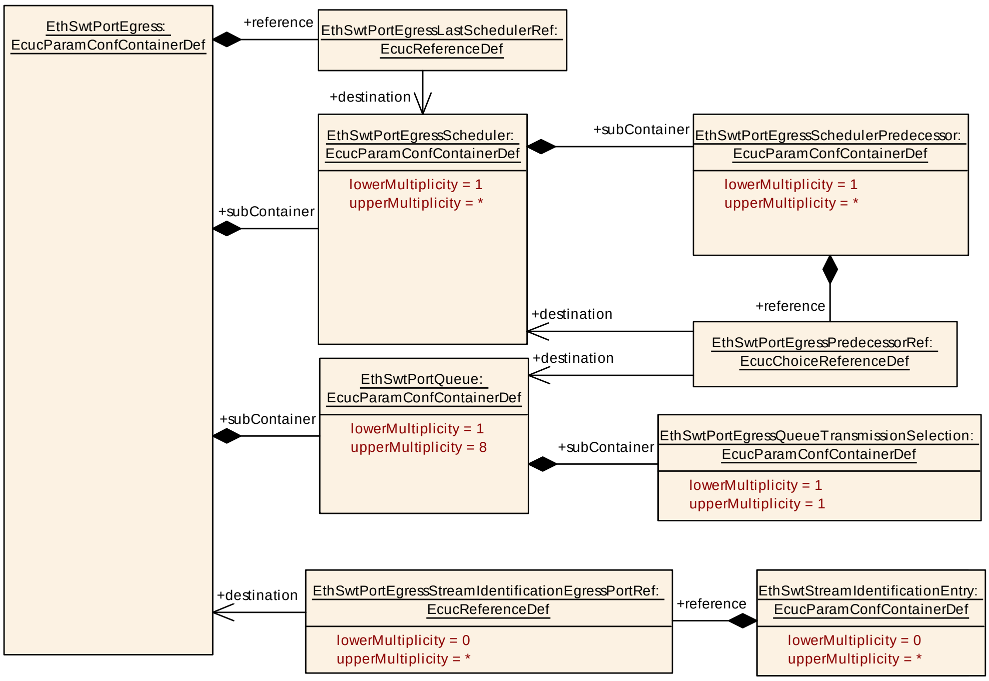


Figure 10.12: EthSwtPortEgress (1/3)

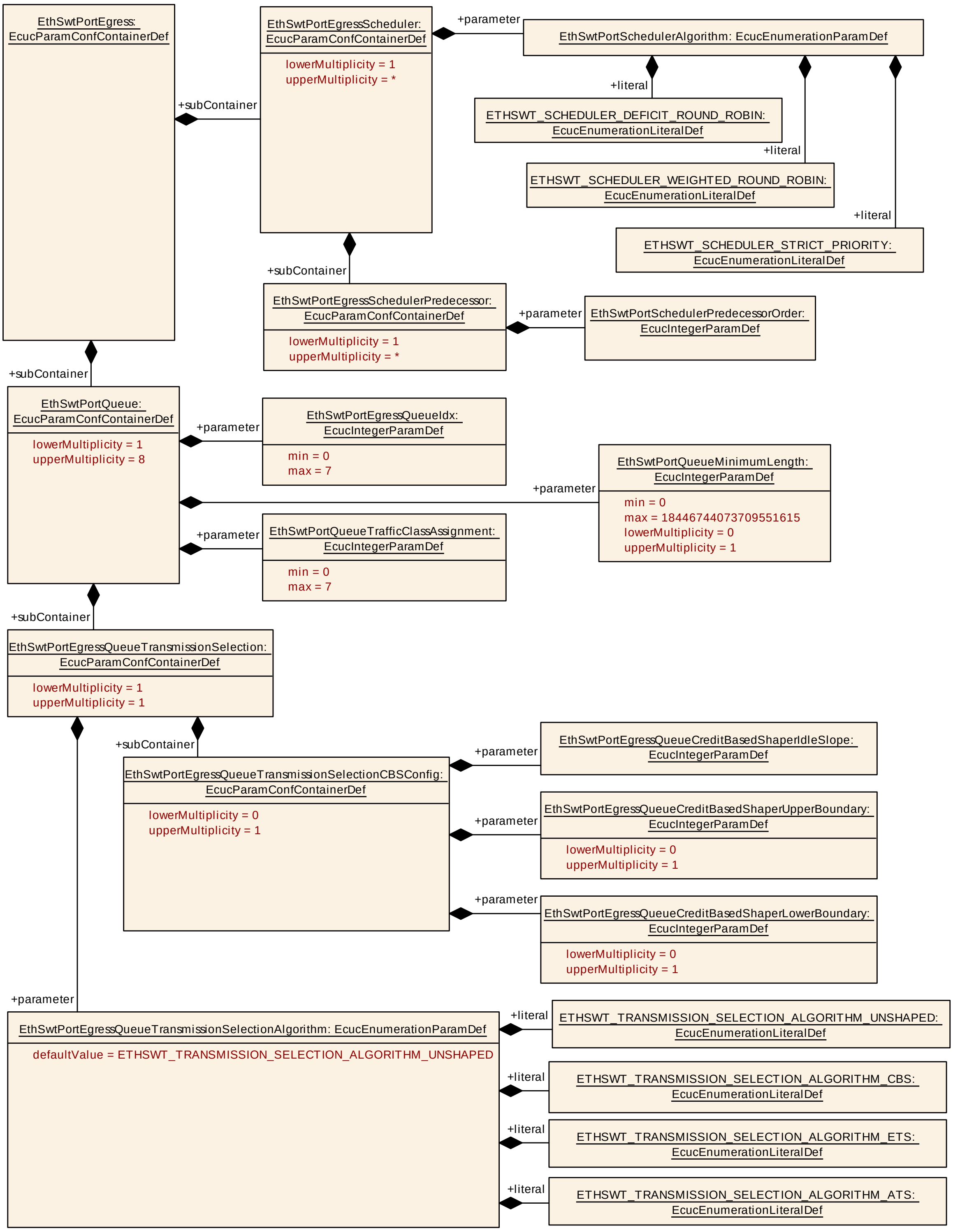

EthSwtPortEgress:
EcucParamConfContainerDef
+subContainer
EthSwtPortEgressScheduler:
EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtPortSchedulerAlgorithm: EcucEnumerationParamDef
+literal
ETHSWT_SCHEDULER_DEFICIT_ROUND_ROBIN:
EcucEnumerationLiteralDef
+literal
ETHSWT_SCHEDULER_WEIGHTED_ROUND_ROBIN:
EcucEnumerationLiteralDef
+literal
ETHSWT_SCHEDULER_STRICT_PRIORITY:
EcucEnumerationLiteralDef
+subContainer
EthSwtPortEgressSchedulerPredecessor:
EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtPortSchedulerPredecessorOrder:
EcucIntegerParamDef
+subContainer
EthSwtPortQueue:
EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = 8
+parameter
EthSwtPortEgressQueueIdx:
EcucIntegerParamDef
min = 0
max = 7
+parameter
EthSwtPortQueueMinimumLength:
EcucIntegerParamDef
min = 0
max = 18446744073709551615
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtPortQueueTrafficClassAssignment:
EcucIntegerParamDef
min = 0
max = 7
+subContainer
EthSwtPortEgressQueueTransmissionSelection:
EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = 1
+subContainer
EthSwtPortEgressQueueTransmissionSelectionCBSConfig:
EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtPortEgressQueueCreditBasedShaperIdleSlope:
EcucIntegerParamDef
+parameter
EthSwtPortEgressQueueCreditBasedShaperUpperBoundary:
EcucIntegerParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtPortEgressQueueCreditBasedShaperLowerBoundary:
EcucIntegerParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtPortEgressQueueTransmissionSelectionAlgorithm: EcucEnumerationParamDef
defaultValue = ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_UNSHAPED
+literal
ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_UNSHAPED:
EcucEnumerationLiteralDef
+literal
ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_CBS:
EcucEnumerationLiteralDef
+literal
ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_ETS:
EcucEnumerationLiteralDef
+literal
ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_ATS:
EcucEnumerationLiteralDef

Figure 10.13: EthSwtPortEgress (2/3)

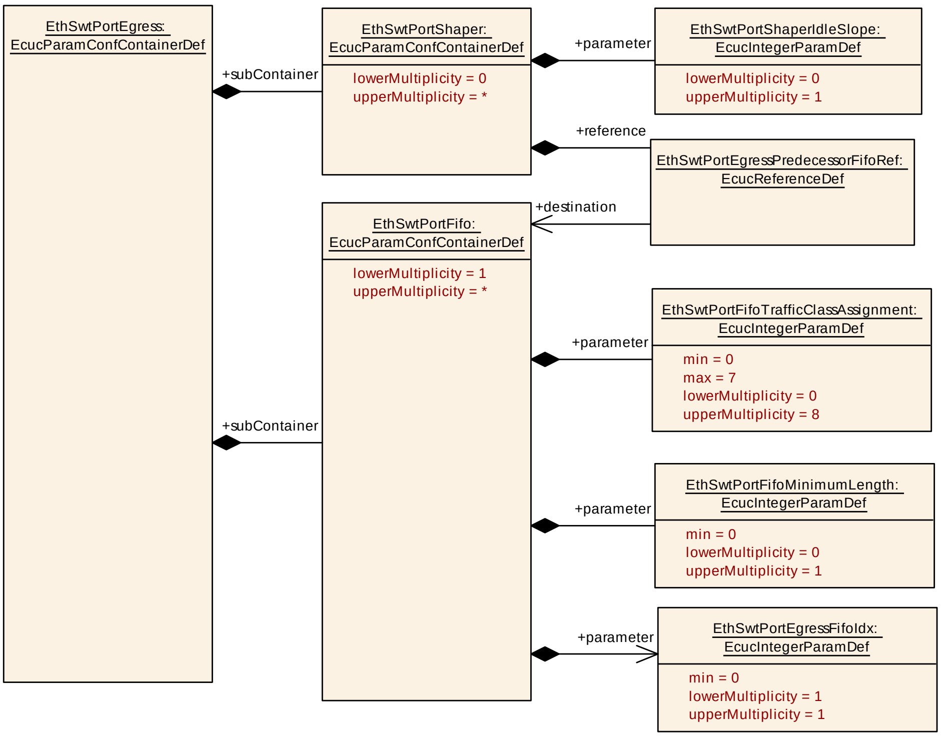

EthSwtPortEgress: EcucParamConfContainerDef
+subContainer
EthSwtPortShaper: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
EthSwtPortShaperIdleSlope: EcucIntegerParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+reference
EthSwtPortEgressPredecessorFifoRef: EcucReferenceDef
+destination
EthSwtPortFifo: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtPortFifoTrafficClassAssignment: EcucIntegerParamDef
min = 0
max = 7
lowerMultiplicity = 0
upperMultiplicity = 8
+subContainer
+parameter
EthSwtPortFifoMinimumLength: EcucIntegerParamDef
min = 0
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtPortEgressFifoIdx: EcucIntegerParamDef
min = 0
lowerMultiplicity = 1
upperMultiplicity = 1

Figure 10.14: `EthSwtPortEgressShaperFifo` (3/3)

10.1.24 EthSwtPortEgressScheduler

SWS Item	**[ECUC_EthSwt_00017]**		
Container Name	EthSwtPortEgressScheduler		
Parent Container	EthSwtPortEgress		
Description	Represents a Scheduler in the egress port.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00018]**	
Parameter Name	EthSwtPortSchedulerAlgorithm	
Parent Container	EthSwtPortEgressScheduler	
Description	Defines the scheduler algorithm.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	ETHSWT_SCHEDULER_DEFICIT_ROUND_ROBIN	deficit round robin
	ETHSWT_SCHEDULER_STRICT_PRIORITY	strict priority

▽

△

	ETHSWT_SCHEDULER_ WEIGHTED_ROUND_ROBIN	weighted round robin	
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		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtPortEgressScheduler Predecessor	1..*	Defines an ordered list of predecessors for this scheduler.

10.1.25 EthSwtPortEgressSchedulerPredecessor

SWS Item	**[ECUC_EthSwt_00019]**		
Container Name	EthSwtPortEgressSchedulerPredecessor		
Parent Container	EthSwtPortEgressScheduler		
Description	Defines an ordered list of predecessors for this scheduler.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00020]**		
Parameter Name	EthSwtPortSchedulerPredecessorOrder		
Parent Container	EthSwtPortEgressSchedulerPredecessor		
Description	Defines the order of the scheduler predecessors. This value has to be understood as a relative value, i.e. the value shows only the relative ordering of the elements. The highest value has the highest priority and gaps are allowed (not dense based). The values need to be unique within one EthSwtPort Scheduler.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
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: ECU		

SWS Item	**[ECUC_EthSwt_00010]**		
Parameter Name	EthSwtPortEgressPredecessorRef		
Parent Container	EthSwtPortEgressSchedulerPredecessor		
Description	Choice reference to the scheduler predecessor.		
Multiplicity	1		
Type	Choice reference to [EthSwtPortEgressScheduler, EthSwtPortFifo, EthSwtPortQueue, EthSwtPortShaper]		
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		

No Included Containers

10.1.26 EthSwtPortFifo

SWS Item	**[ECUC_EthSwt_00011]** (Obsolete)		
Container Name	EthSwtPortFifo		
Parent Container	EthSwtPortEgress		
Description	Represents a Fifo in the egress port. **Tags:** atp.Status=obsolete		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00132]** (Obsolete)		
Parameter Name	EthSwtPortEgressFifoIdx		
Parent Container	EthSwtPortFifo		
Description	Specifies the instance ID of the fifo of the configured Ethernet switch egress port **Tags:** atp.Status=obsolete		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_EthSwt_00098]** (Obsolete)		
Parameter Name	EthSwtPortFifoMinimumLength		
Parent Container	EthSwtPortFifo		
Description	FIFO minimum length in Byte. This assignment is used to configure a guaranteed size of a configured FIFO. **Tags:** atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_EthSwt_00012]** (Obsolete)		
Parameter Name	EthSwtPortFifoTrafficClassAssignment		
Parent Container	EthSwtPortFifo		
Description	Defines which traffic classes are assigned to this Fifo. **Tags:** atp.Status=obsolete		
Multiplicity	0..8		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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: ECU		

No Included Containers

10.1.27 EthSwtPortQueue

SWS Item	**[ECUC_EthSwt_00182]**
Container Name	EthSwtPortQueue
Parent Container	EthSwtPortEgress
Description	Represents a Queue at the egress port. **Tags:** atp.Status=draft

▽

△

Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00183]**		
Parameter Name	EthSwtPortEgressQueueIdx		
Parent Container	EthSwtPortQueue		
Description	Specifies the instance ID of the queue of the configured Ethernet switch egress port. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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		

SWS Item	**[ECUC_EthSwt_00184]**		
Parameter Name	EthSwtPortQueueMinimumLength		
Parent Container	EthSwtPortQueue		
Description	Queue minimum length in Byte. This assignment is used to configure a guaranteed size of a configured Queue. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00185]**
Parameter Name	EthSwtPortQueueTrafficClassAssignment
Parent Container	EthSwtPortQueue
Description	Defines which traffic class is mapped to this queue. **Tags:** atp.Status=draft

▽

△

Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtPortEgressQueueTransmissionSelection	1	Represents the transmission selection of an egress port queue. **Tags:** atp.Status=draft

10.1.28 EthSwtPortEgressQueueTransmissionSelection

SWS Item	**[ECUC_EthSwt_00186]**
Container Name	EthSwtPortEgressQueueTransmissionSelection
Parent Container	EthSwtPortQueue
Description	Represents the transmission selection of an egress port queue. **Tags:** atp.Status=draft
Configuration Parameters	

SWS Item	**[ECUC_EthSwt_00191]**	
Parameter Name	EthSwtPortEgressQueueTransmissionSelectionAlgorithm	
Parent Container	EthSwtPortEgressQueueTransmissionSelection	
Description	Represents the transmission selection algorithm of an egress port queue. **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_ATS	Ethernet frames are selected from the egress queue for transmission according the asynchronous traffic shaping algorithm. **Tags:** atp.Status=draft
	ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_CBS	Ethernet frames are selected from the egress queue for transmission according the credit based shaping algorithm. **Tags:** atp.Status=draft
	ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_ETS	Ethernet frames are selected from the egress queue for transmission according the enhanced transmission selection algorithm. **Tags:** atp.Status=draft

▽

△

	ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_UNSHAPED	Ethernet frames are selected from the egress queue for transmission in an unshaped manner. Please note: IEEE802.1Q uses the term "strict priority". Term "UNSHAPED" is used to avoid confusion with strict priority in context of EthSwtPortEgressScheduler. **Tags:** atp.Status=draft	
Default value	ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_UNSHAPED		
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		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtPortEgressQueueTransmissionSelectionCBSConfig	0..1	Represents the configuration of a credit based shaper transmission selection algorithm of an egress port queue. This configuration is used if the EthSwtPortEgressQueueTransmissionSelectionAlgorithm is set to ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_CBS. **Tags:** atp.Status=draft

10.1.29 EthSwtPortEgressQueueTransmissionSelectionCBSConfig

SWS Item	**[ECUC_EthSwt_00187]**		
Container Name	EthSwtPortEgressQueueTransmissionSelectionCBSConfig		
Parent Container	EthSwtPortEgressQueueTransmissionSelection		
Description	Represents the configuration of a credit based shaper transmission selection algorithm of an egress port queue. This configuration is used if the EthSwtPortEgressQueueTransmissionSelectionAlgorithm is set to ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_CBS. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00188]**
Parameter Name	EthSwtPortEgressQueueCreditBasedShaperIdleSlope
Parent Container	EthSwtPortEgressQueueTransmissionSelectionCBSConfig

▽

△

Description	Defines the increase of credit in bits per second for the AVB shaper. Note: this parameter maps to IEEE802.1Q parameter "ieee8021FqtssAdminIdleSlopeMs" and "ieee8021FqtssAdminIdleSlopeLs". **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
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		

SWS Item	**[ECUC_EthSwt_00190]**		
Parameter Name	EthSwtPortEgressQueueCreditBasedShaperLowerBoundary		
Parent Container	EthSwtPortEgressQueueTransmissionSelectionCBSConfig		
Description	Defines the lower credit boundary for the Credit Based Shaper. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
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		

SWS Item	**[ECUC_EthSwt_00189]**		
Parameter Name	EthSwtPortEgressQueueCreditBasedShaperUpperBoundary		
Parent Container	EthSwtPortEgressQueueTransmissionSelectionCBSConfig		
Description	Defines the upper credit boundary for the Credit Based Shaper. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	

▽

△

	Post-build time	–	
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		

No Included Containers

10.1.30 EthSwtPortShaper

SWS Item	**[ECUC_EthSwt_00021]** (Obsolete)		
Container Name	EthSwtPortShaper		
Parent Container	EthSwtPortEgress		
Description	Represents a Shaper in the egress port. **Tags:** atp.Status=obsolete		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00042]** (Obsolete)		
Parameter Name	EthSwtPortShaperIdleSlope		
Parent Container	EthSwtPortShaper		
Description	Defines the increase of credit in bits per second for the AVB shaper. **Tags:** atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
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		

SWS Item	**[ECUC_EthSwt_00009]** (Obsolete)
Parameter Name	EthSwtPortEgressPredecessorFifoRef
Parent Container	EthSwtPortShaper
Description	Reference to the fifo which is the predecessor for this shaper. **Tags:** atp.Status=obsolete
Multiplicity	1
Type	Reference to EthSwtPortFifo

▽

△

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		

No Included Containers

10.1.31 EthSwtPortIngress

SWS Item	**[ECUC_EthSwt_00014]**
Container Name	EthSwtPortIngress
Parent Container	EthSwtPort
Description	Configuration of one Ethernet Switch Port ingress behavior.
Configuration Parameters	

SWS Item	**[ECUC_EthSwt_00096]**		
Parameter Name	EthSwtPortIngressDefaultPriority		
Parent Container	EthSwtPortIngress		
Description	Default priority for ingress.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default value	0		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: If EthSwtPortIngressDefaultPriority is configured (multiplicity set to 1) then EthSwtPortIngressDefaultVlan shall be configured. If EthSwtPortIngressDefaultVlan is configured EthSwtPortIngressDropUntagged shall be set to FALSE.		

SWS Item	**[ECUC_EthSwt_00095]**	
Parameter Name	EthSwtPortIngressDefaultVlan	
Parent Container	EthSwtPortIngress	
Description	Default VLAN for ingress.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 4094	

▽

△

Default value	1		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: If EthSwtPortIngressDefaultVlan is configured (multiplicity set to 1) then EthSwtPortIngressDefaultPriority shall be configured. If EthSwtPortIngressDefaultVlan is configured EthSwtPortIngressDropUntagged shall be set to FALSE.		

SWS Item	**[ECUC_EthSwt_00097]**		
Parameter Name	EthSwtPortIngressDropUntagged		
Parent Container	EthSwtPortIngress		
Description	Defines the ingress behavior for untagged frames.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
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 dependency: If EthSwtPortIngressDropUntagged is set to TRUE then EthSwtPortIngressDefaultVlan and EthSwtPortIngressDefaultPriority parameters shall not be configured.		

SWS Item	**[ECUC_EthSwt_00015]**		
Parameter Name	EthSwtPortIngressVlanModification		
Parent Container	EthSwtPortIngress		
Description	If this parameter is defined all messages which arrive at this ingress port will be tagged with this VLAN-ID. This tagging happens also if the arriving message already has a VLAN-ID, it will be overwritten by the defined one. If this parameter is not defined no changes to the VLAN-ID shall happen at this ingress port.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4095		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE

▽

△

	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_EthSwt_00023]**		
Parameter Name	EthSwtPortTrafficClassAssignment		
Parent Container	EthSwtPortIngress		
Description	If this parameter is defined all arriving messages at this ingress port shall be assigned this traffic class. If this parameter is not defined no general port based traffic class assignment is done.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtPortIngressScheduler	0..*	Represents a Scheduler configuration at an ingress port. **Tags:** atp.Status=draft
EthSwtPortOutboundVlanPriorityAssignment	0..8	Defines a priority mapping from a regenerated VLAN priority (EthSwtPortOutboundVlanPriorityAssignmentOutboundVlanPriority) to an outbound VLAN priority (EthSwtPortOutboundVlanPriorityAssignmentRegeneratedPriority). The EthSwtPortOutboundVlanPriorityAssignment is optional. The outbound priority mapping shall only be performed if EthSwtPortOutboundVlanPriorityAssignment is configured. In case an EthSwtPortOutboundVlanPriorityAssignment is defined it shall have 8 mappings, one for each priority. **Tags:** atp.Status=draft
EthSwtPortPolicer	0..32760	Definition of Rate Policing parameters. **Tags:** atp.Status=obsolete
EthSwtPortPriorityRegeneration	8	Defines a priority regeneration where the EthSwtPortPriorityRegenerationIngressPCP is replaced by EthSwtPortPriorityRegenerationRegeneratedPriority. The EthSwtPortPriorityRegeneration is mandatory and shall always be available. An EthSwtPortPriorityRegeneration shall have 8 mappings, one for each priority. Rational: an Ethernet switch always performs a priority regeneration.

▽

△

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtPortPriorityTrafficClassAssignment	0..8	Defines a priority based traffic class assignment. All messages with a specific priority (EthSwtPortPriorityTrafficClassAssignmentRegeneratedPriority) arriving at this ingress port or, if enabled regenerated priorities (EthSwtPortPriorityRegeneration), shall be assigned to a traffic class (EthSwtPortPriorityTrafficClassAssignmentTrafficClass).

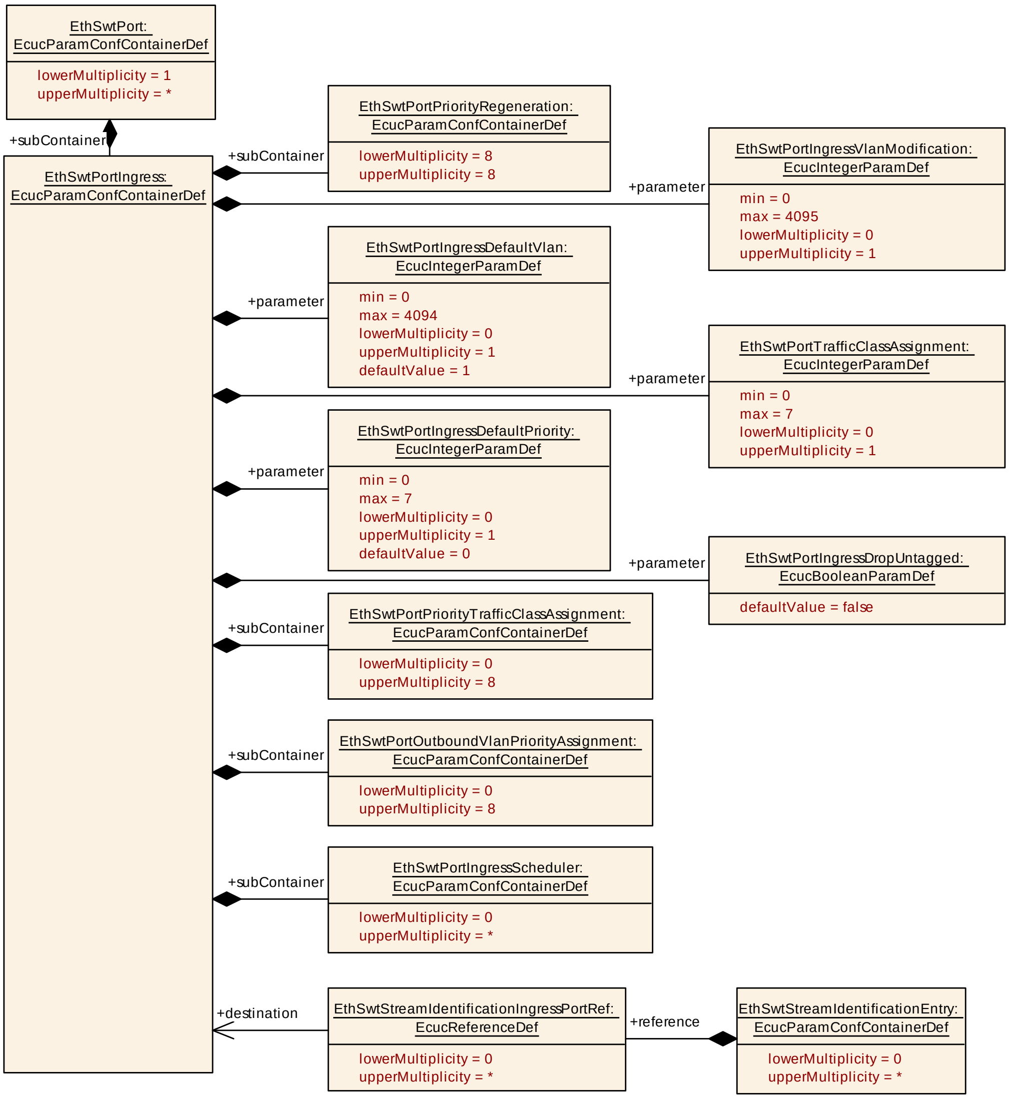


Figure 10.15: EthSwtPortIngress (1/2)

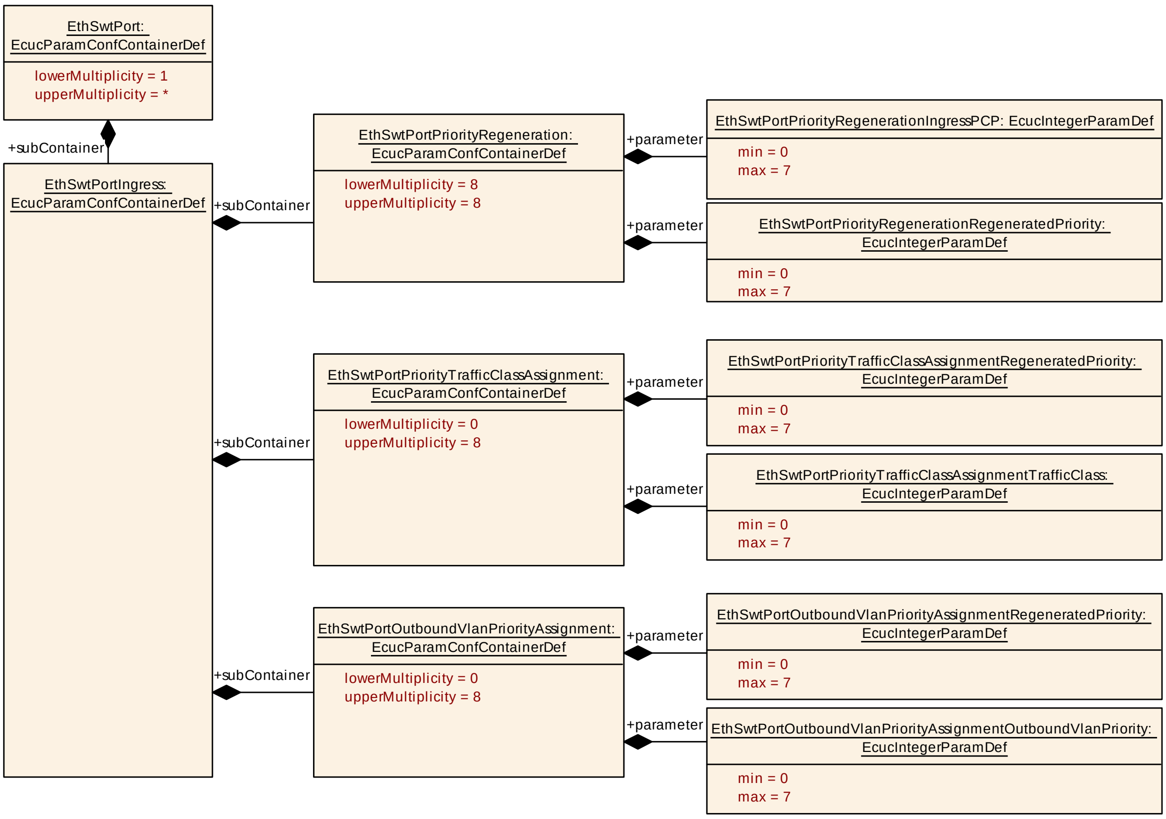

EthSwtPort: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
EthSwtPortIngress: EcucParamConfContainerDef
+subContainer
EthSwtPortPriorityRegeneration: EcucParamConfContainerDef
lowerMultiplicity = 8
upperMultiplicity = 8
+parameter
EthSwtPortPriorityRegenerationIngressPCP: EcucIntegerParamDef
min = 0
max = 7
+parameter
EthSwtPortPriorityRegenerationRegeneratedPriority: EcucIntegerParamDef
min = 0
max = 7
+subContainer
EthSwtPortPriorityTrafficClassAssignment: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 8
+parameter
EthSwtPortPriorityTrafficClassAssignmentRegeneratedPriority: EcucIntegerParamDef
min = 0
max = 7
+parameter
EthSwtPortPriorityTrafficClassAssignmentTrafficClass: EcucIntegerParamDef
min = 0
max = 7
+subContainer
EthSwtPortOutboundVlanPriorityAssignment: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 8
+parameter
EthSwtPortOutboundVlanPriorityAssignmentRegeneratedPriority: EcucIntegerParamDef
min = 0
max = 7
+parameter
EthSwtPortOutboundVlanPriorityAssignmentOutboundVlanPriority: EcucIntegerParamDef
min = 0
max = 7

Figure 10.16: EthSwtPortIngress (2/2)

10.1.32 EthSwtPortIngressScheduler

SWS Item	**[ECUC_EthSwt_00139]**		
Container Name	EthSwtPortIngressScheduler		
Parent Container	EthSwtPortIngress		
Description	Represents a Scheduler configuration at an ingress port. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

No Included Containers

10.1.33 EthSwtPortOutboundVlanPriorityAssignment

SWS Item	**[ECUC_EthSwt_00138]**		
Container Name	EthSwtPortOutboundVlanPriorityAssignment		
Parent Container	EthSwtPortIngress		
Description	Defines a priority mapping from a regenerated VLAN priority (EthSwtPortOutboundVlanPriorityAssignmentOutboundVlanPriority) to an outbound VLAN priority (EthSwtPortOutboundVlanPriorityAssignmentRegeneratedPriority). The EthSwtPortOutboundVlanPriorityAssignment is optional. The outbound priority mapping shall only be performed if EthSwtPortOutboundVlanPriorityAssignment is configured. In case an EthSwtPortOutboundVlanPriorityAssignment is defined it shall have 8 mappings, one for each priority. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00193]**		
Parameter Name	EthSwtPortOutboundVlanPriorityAssignmentOutboundVlanPriority		
Parent Container	EthSwtPortOutboundVlanPriorityAssignment		
Description	Message priority the outgoing message will be tagged with. Note: This parameter maps IEEE802.1Q parameter "ieee8021BridgePortOutboundAccessPriority". **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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		

SWS Item	**[ECUC_EthSwt_00192]**	
Parameter Name	EthSwtPortOutboundVlanPriorityAssignmentRegeneratedPriority	
Parent Container	EthSwtPortOutboundVlanPriorityAssignment	
Description	Regenerated priority of the message. Note: this parameter maps to IEEE802.1Q parameter "eee8021BridgeRegenUserPriority" **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 7	
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		

No Included Containers

10.1.34 EthSwtPortPolicer

SWS Item	**[ECUC_EthSwt_00074]** (Obsolete)		
Container Name	EthSwtPortPolicer		
Parent Container	EthSwtPortIngress		
Description	Definition of Rate Policing parameters. **Tags:** atp.Status=obsolete		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00075]** (Obsolete)		
Parameter Name	EthSwtPortRatePolicedByteCount		
Parent Container	EthSwtPortPolicer		
Description	Amount of Byte Counts (excluding Header information) which can be received in a configured EthSwtPortRatePolicedTimeInterval. **Tags:** atp.Status=obsolete		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 18446744073709551615		
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		

SWS Item	**[ECUC_EthSwt_00077]** (Obsolete)
Parameter Name	EthSwtPortRatePolicedPriority
Parent Container	EthSwtPortPolicer
Description	Defines the priority which this rate policy shall be limited on. If no priority is given this rate policy is not considering priority. **Tags:** atp.Status=obsolete
Multiplicity	0..1

▽

△

Type	EcucIntegerParamDef		
Range	0 .. 7		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: If no priority is configured the rate policing only applies to the configured EthSwtRateVlanMembershipRef.		

SWS Item	**[ECUC_EthSwt_00076]** (Obsolete)		
Parameter Name	EthSwtPortRatePolicedTimeInterval		
Parent Container	EthSwtPortPolicer		
Description	Time interval in seconds where a configured EthSwtPortRatePolicedByteCount can be received without a rate limitation. **Tags:** atp.Status=obsolete		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
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		

SWS Item	**[ECUC_EthSwt_00078]** (Obsolete)		
Parameter Name	EthSwtPortRateViolationAction		
Parent Container	EthSwtPortPolicer		
Description	Action to be taken when the rate policy criteria defined for this EthSwtPortPolicer are met. **Tags:** atp.Status=obsolete		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BLOCK_SOURCE	All incoming traffic from the violating Source based on the MAC-Address is blocked. **Tags:** atp.Status=obsolete	
	DROP_FRAME	The received frame which led to the violation of the rate policy is dropped. **Tags:** atp.Status=obsolete	
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		

SWS Item	**[ECUC_EthSwt_00081]** (Obsolete)		
Parameter Name	EthSwtRateVlanMembershipRef		
Parent Container	EthSwtPortPolicer		
Description	References the Vlans this rate policy shall apply to. If no EthSwtRateVlanMembershipRef is configured the rate policing applies only on the configured EthSwtPortRatePolicedPriority. **Tags:** atp.Status=obsolete		
Multiplicity	0..4095		
Type	Reference to EthSwtVlanMembership		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

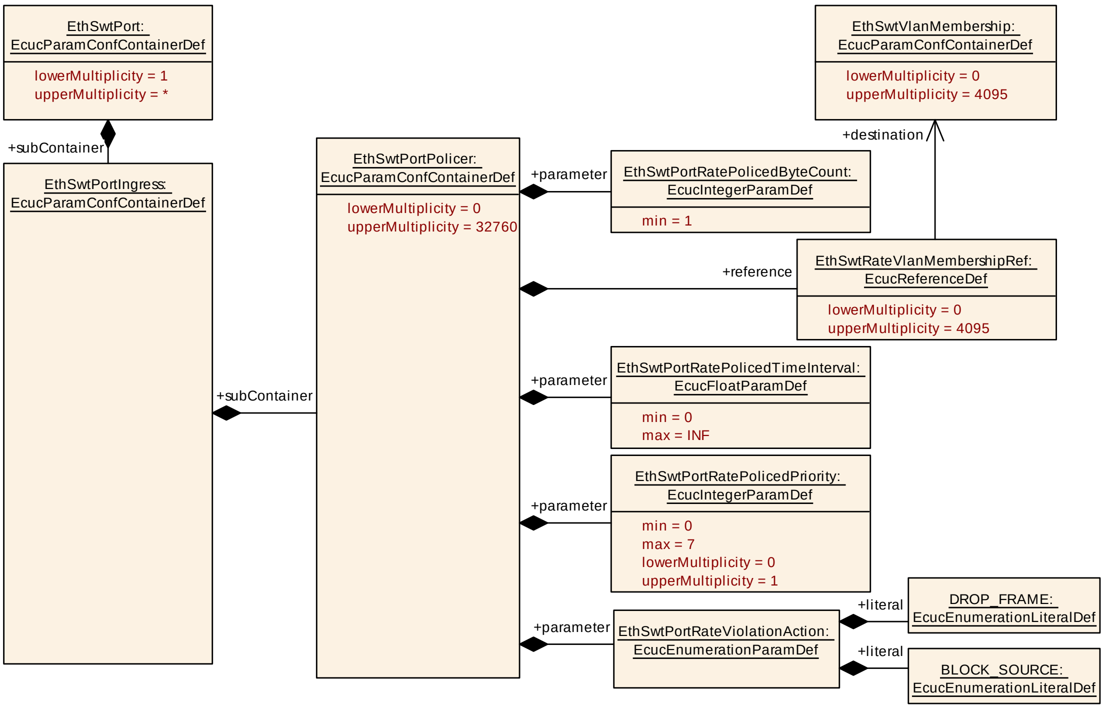

EthSwtPort: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
EthSwtPortIngress: EcucParamConfContainerDef
+subContainer
EthSwtPortPolicer: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 32760
+parameter
EthSwtPortRatePolicedByteCount: EcucIntegerParamDef
min = 1
EthSwtVlanMembership: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 4095
+destination
+reference
EthSwtRateVlanMembershipRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 4095
+parameter
EthSwtPortRatePolicedTimeInterval: EcucFloatParamDef
min = 0
max = INF
+parameter
EthSwtPortRatePolicedPriority: EcucIntegerParamDef
min = 0
max = 7
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtPortRateViolationAction: EcucEnumerationParamDef
+literal
DROP_FRAME: EcucEnumerationLiteralDef
+literal
BLOCK_SOURCE: EcucEnumerationLiteralDef

Figure 10.17: EthSwtPortPolicer

10.1.35 EthSwtPortPriorityRegeneration

SWS Item	**[ECUC_EthSwt_00057]**		
Container Name	EthSwtPortPriorityRegeneration		
Parent Container	EthSwtPortIngress		
Description	Defines a priority regeneration where the EthSwtPortPriorityRegenerationIngressPCP is replaced by EthSwtPortPriorityRegenerationRegeneratedPriority. The EthSwtPortPriorityRegeneration is mandatory and shall always be available. An EthSwtPortPriorityRegeneration shall have 8 mappings, one for each priority. Rational: an Ethernet switch always performs a priority regeneration.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00058]**		
Parameter Name	EthSwtPortPriorityRegenerationIngressPCP		
Parent Container	EthSwtPortPriorityRegeneration		
Description	PCP (VLAN-priority) in the incoming message.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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: ECU		

SWS Item	**[ECUC_EthSwt_00059]**		
Parameter Name	EthSwtPortPriorityRegenerationRegeneratedPriority		
Parent Container	EthSwtPortPriorityRegeneration		
Description	Message priority the incoming message will be tagged with.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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: ECU		

No Included Containers

10.1.36 EthSwtPortPriorityTrafficClassAssignment

SWS Item	[ECUC_EthSwt_00027]		
Container Name	EthSwtPortPriorityTrafficClassAssignment		
Parent Container	EthSwtPortIngress		
Description	Defines a priority based traffic class assignment. All messages with a specific priority (EthSwtPortPriorityTrafficClassAssignmentRegeneratedPriority) arriving at this ingress port or, if enabled regenerated priorities (EthSwtPortPriorityRegeneration), shall be assigned to a traffic class (EthSwtPortPriorityTrafficClassAssignmentTrafficClass).		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	[ECUC_EthSwt_00028]		
Parameter Name	EthSwtPortPriorityTrafficClassAssignmentRegeneratedPriority		
Parent Container	EthSwtPortPriorityTrafficClassAssignment		
Description	Regenerated priority of the message.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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: ECU		

SWS Item	[ECUC_EthSwt_00029]		
Parameter Name	EthSwtPortPriorityTrafficClassAssignmentTrafficClass		
Parent Container	EthSwtPortPriorityTrafficClassAssignment		
Description	Traffic Class value.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
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: ECU		

No Included Containers

10.1.37 EthSwtSpi

SWS Item	**[ECUC_EthSwt_00030]**		
Container Name	EthSwtSpi		
Parent Container	EthSwtConfig		
Description	Configuration of one Ethernet Switch SPI access (if SPI is used).		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtSpiSequence	1..*	Container gives EthSwt driver information about one SPI sequence. One SPI sequence used by EthSwt driver is in exclusive use for it. No other driver is allowed to access this sequence. EthSwt driver may use one sequence to access n Eth Swt hardware chips of the same type or n sequences are used to access one single EthSwt hardware chip. If a EthSwt hardware has no SPI interface, there is no instance of this container.

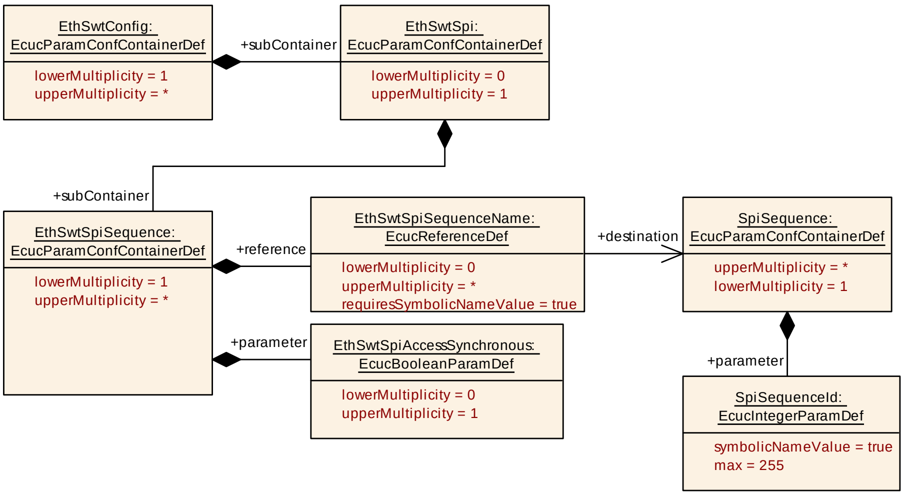


Figure 10.18: EthSwtSpi

10.1.38 EthSwtSpiSequence

SWS Item	**[ECUC_EthSwt_00034]**
Container Name	EthSwtSpiSequence
Parent Container	EthSwtSpi

▽

△

Description	Container gives EthSwt driver information about one SPI sequence. One SPI sequence used by EthSwt driver is in exclusive use for it. No other driver is allowed to access this sequence. EthSwt driver may use one sequence to access n EthSwt hardware chips of the same type or n sequences are used to access one single EthSwt hardware chip. If a EthSwt hardware has no SPI interface, there is no instance of this container.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00036]**		
Parameter Name	EthSwtSpiAccessSynchronous		
Parent Container	EthSwtSpiSequence		
Description	This parameter is used to define whether the access to the Spi sequence is synchronous or asynchronous. true: SPI access is synchronous. false: SPI access is asynchronous.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_EthSwt_00035]**		
Parameter Name	EthSwtSpiSequenceName		
Parent Container	EthSwtSpiSequence		
Description	Reference to a Spi sequence configuration container.		
Multiplicity	0..*		
Type	Symbolic name reference to SpiSequence		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.1.39 EthSwtStreamIdentificationTable

SWS Item	**[ECUC_EthSwt_00208]**		
Container Name	EthSwtStreamIdentificationTable		
Parent Container	EthSwtConfig		
Description	Configuration of a stream identification table. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtStreamIdentificationEntry	0..*	Configuration of a stream identification. **Tags:** atp.Status=draft

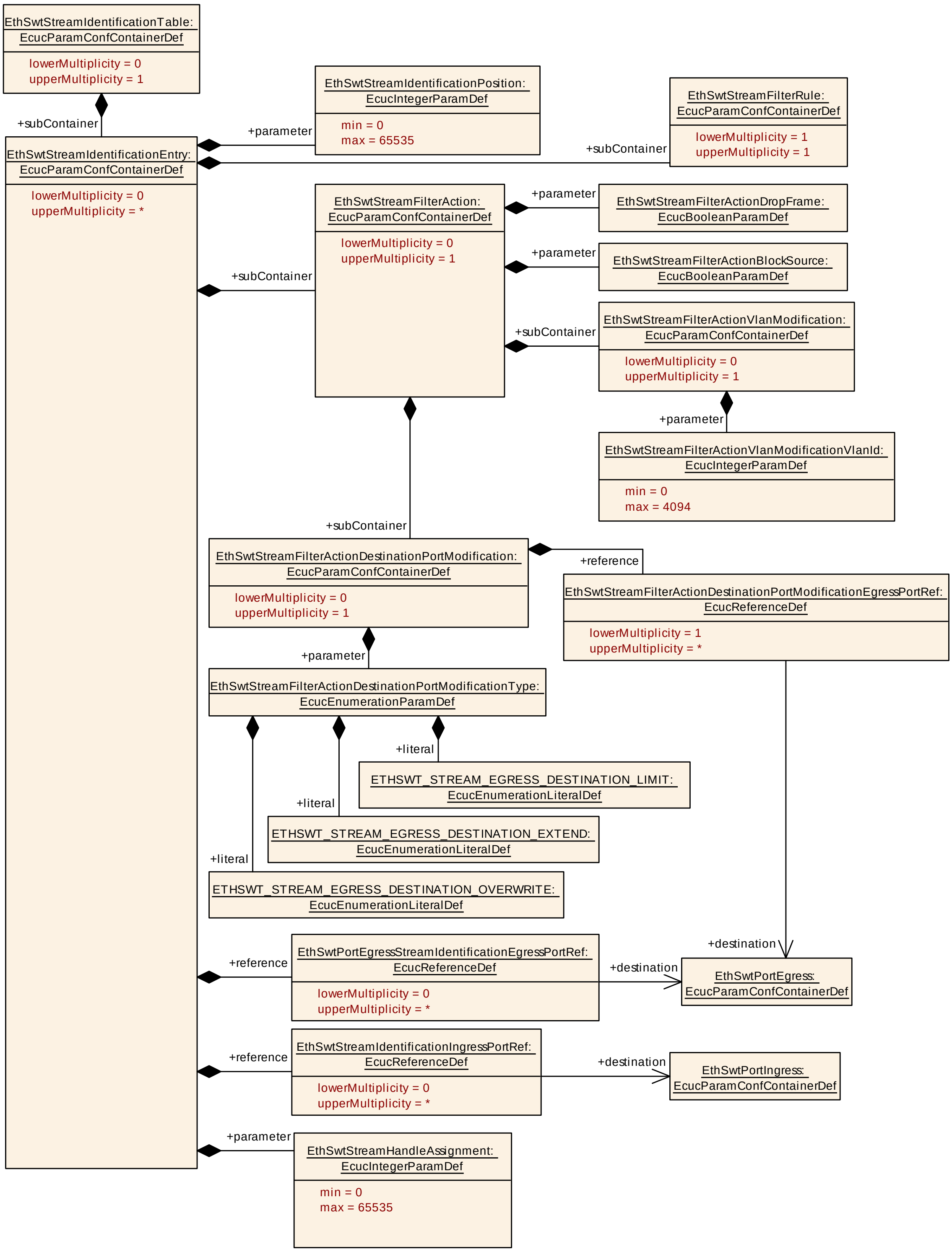

EthSwtStreamIdentificationTable:
EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+subContainer
EthSwtStreamIdentificationEntry:
EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
EthSwtStreamIdentificationPosition:
EcucIntegerParamDef
min = 0
max = 65535
+subContainer
EthSwtStreamFilterRule:
EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = 1
+subContainer
EthSwtStreamFilterAction:
EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtStreamFilterActionDropFrame:
EcucBooleanParamDef
+parameter
EthSwtStreamFilterActionBlockSource:
EcucBooleanParamDef
+subContainer
EthSwtStreamFilterActionVlanModification:
EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthSwtStreamFilterActionVlanModificationVlanId:
EcucIntegerParamDef
min = 0
max = 4094
+subContainer
EthSwtStreamFilterActionDestinationPortModification:
EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+reference
EthSwtStreamFilterActionDestinationPortModificationEgressPortRef:
EcucReferenceDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthSwtStreamFilterActionDestinationPortModificationType:
EcucEnumerationParamDef
+literal
ETHSWT_STREAM_EGRESS_DESTINATION_LIMIT:
EcucEnumerationLiteralDef
+literal
ETHSWT_STREAM_EGRESS_DESTINATION_EXTEND:
EcucEnumerationLiteralDef
+literal
ETHSWT_STREAM_EGRESS_DESTINATION_OVERWRITE:
EcucEnumerationLiteralDef
+destination
+reference
EthSwtPortEgressStreamIdentificationEgressPortRef:
EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = *
+destination
EthSwtPortEgress:
EcucParamConfContainerDef
+reference
EthSwtStreamIdentificationIngressPortRef:
EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = *
+destination
EthSwtPortIngress:
EcucParamConfContainerDef
+parameter
EthSwtStreamHandleAssignment:
EcucIntegerParamDef
min = 0
max = 65535

Figure 10.19: EthSwtStreamIdentificationTable

10.1.40 EthSwtStreamIdentificationEntry

SWS Item	**[ECUC_EthSwt_00140]**		
Container Name	EthSwtStreamIdentificationEntry		
Parent Container	EthSwtStreamIdentificationTable		
Description	Configuration of a stream identification. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00211]**		
Parameter Name	EthSwtStreamHandleAssignment		
Parent Container	EthSwtStreamIdentificationEntry		
Description	Assigment of this stream identification to an stream filter entry. **Tags:** atp.Status=draft		
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		

SWS Item	**[ECUC_EthSwt_00142]**		
Parameter Name	EthSwtStreamIdentificationPosition		
Parent Container	EthSwtStreamIdentificationEntry		
Description	Specifies the position as unique ID within an ordered list of EthSwtStreamIdentification Entrys. The ordered list shall start with 0 and continue as linear list with no gaps. Note: The list is processed in ascending order. The instance of EthSwtStream IdentificationEntry with position 0 is processed first. **Tags:** atp.Status=draft		
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: ECU		

SWS Item	**[ECUC_EthSwt_00153]**		
Parameter Name	EthSwtPortEgressStreamIdentificationEgressPortRef		
Parent Container	EthSwtStreamIdentificationEntry		
Description	Reference to the egress ports this stream identification applies to. **Tags:** atp.Status=draft		
Multiplicity	0..*		
Type	Reference to EthSwtPortEgress		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00152]**		
Parameter Name	EthSwtStreamIdentificationIngressPortRef		
Parent Container	EthSwtStreamIdentificationEntry		
Description	Reference to the ingress ports this stream identification applies to. **Tags:** atp.Status=draft		
Multiplicity	0..*		
Type	Reference to EthSwtPortIngress		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtStreamFilterAction	0..1	Configuration of a stream filter action. **Tags:** atp.Status=draft
EthSwtStreamFilterRule	1	Configuration of a filter rule. **Tags:** atp.Status=draft

10.1.41 EthSwtStreamFilterAction

SWS Item	**[ECUC_EthSwt_00143]**		
Container Name	EthSwtStreamFilterAction		
Parent Container	EthSwtStreamIdentificationEntry		
Description	Configuration of a stream filter action. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00145]**		
Parameter Name	EthSwtStreamFilterActionBlockSource		
Parent Container	EthSwtStreamFilterAction		
Description	Enables Blocking all frames from the MAC address. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
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		

SWS Item	**[ECUC_EthSwt_00144]**		
Parameter Name	EthSwtStreamFilterActionDropFrame		
Parent Container	EthSwtStreamFilterAction		
Description	Enables Drop Frame action. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
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		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtStreamFilterActionDestinationPortModification	0..1	Defines the action to modify the destination port(s) determined by the frame forwarding process for an particular Ethernet frame. Either the egress destination of an Ethernet frame is extended or overwritten. **Tags:** atp.Status=draft

▽

△

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtStreamFilterActionVlanModification	0..1	Defines the action to modify the VLAN-ID within a VLAN-tag of an Ethernet frame. **Tags:** atp.Status=draft

10.1.42 EthSwtStreamFilterActionDestinationPortModification

SWS Item	**[ECUC_EthSwt_00148]**		
Container Name	EthSwtStreamFilterActionDestinationPortModification		
Parent Container	EthSwtStreamFilterAction		
Description	Defines the action to modify the destination port(s) determined by the frame forwarding process for an particular Ethernet frame. Either the egress destination of an Ethernet frame is extended or overwritten. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00150]**		
Parameter Name	EthSwtStreamFilterActionDestinationPortModificationType		
Parent Container	EthSwtStreamFilterActionDestinationPortModification		
Description	Defines the method to modify the egress destination. Either overwrite or extend the egress destination. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ETHSWT_STREAM_EGRESS_DESTINATION_EXTEND	extend the egress destination of an Ethernet frame. **Tags:** atp.Status=draft	
	ETHSWT_STREAM_EGRESS_DESTINATION_LIMIT	limit the egress destination of an Ethernet frame. **Tags:** atp.Status=draft	
	ETHSWT_STREAM_EGRESS_DESTINATION_OVERWRITE	overwrite the egress destination of an Ethernet frame. **Tags:** atp.Status=draft	
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		

SWS Item	**[ECUC_EthSwt_00149]**		
Parameter Name	EthSwtStreamFilterActionDestinationPortModificationEgressPortRef		
Parent Container	EthSwtStreamFilterActionDestinationPortModification		
Description	Defines a set of destination ports (egress ports) used for the modification of the egress destination of an Ethernet frame. **Tags:** atp.Status=draft		
Multiplicity	1..*		
Type	Reference to EthSwtPortEgress		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.1.43 EthSwtStreamFilterActionVlanModification

SWS Item	**[ECUC_EthSwt_00146]**		
Container Name	EthSwtStreamFilterActionVlanModification		
Parent Container	EthSwtStreamFilterAction		
Description	Defines the action to modify the VLAN-ID within a VLAN-tag of an Ethernet frame. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00147]**		
Parameter Name	EthSwtStreamFilterActionVlanModificationVlanId		
Parent Container	EthSwtStreamFilterActionVlanModification		
Description	Defines the VLAN-ID to modify the existing VLAN-ID within the VLAN-tag of an Ethernet frame. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4094		
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		

No Included Containers

10.1.44 EthSwtStreamFilterRule

SWS Item	**[ECUC_EthSwt_00141]**
Container Name	EthSwtStreamFilterRule
Parent Container	EthSwtStreamIdentificationEntry
Description	Configuration of a filter rule. **Tags:** atp.Status=draft
Configuration Parameters	

SWS Item	**[ECUC_EthSwt_00170]**		
Parameter Name	EthSwtStreamFilterEtherType		
Parent Container	EthSwtStreamFilterRule		
Description	Definition of the filter Ether Type. **Tags:** atp.Status=draft		
Multiplicity	0..*		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00181]**
Parameter Name	EthSwtStreamFilterIEEE1722StreamId
Parent Container	EthSwtStreamFilterRule
Description	Definition of the filter IEEE1722 Stream Id. Specifies a 64 bit Stream Id. **Tags:** atp.Status=draft
Multiplicity	0..*
Type	EcucStringParamDef
Default value	–
Regular Expression	–
Post-Build Variant Multiplicity	false

▽

△

Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
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		

SWS Item	**[ECUC_EthSwt_00168]**		
Parameter Name	EthSwtStreamFilterVlanId		
Parent Container	EthSwtStreamFilterRule		
Description	Definition of the filter VLAN-ID. **Tags:** atp.Status=draft		
Multiplicity	0..*		
Type	EcucIntegerParamDef		
Range	0 .. 4094		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00169]**		
Parameter Name	EthSwtStreamFilterVlanPriority		
Parent Container	EthSwtStreamFilterRule		
Description	Definition of the filter VLAN Priority. **Tags:** atp.Status=draft		
Multiplicity	0..*		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtStreamFilterIPDestAddress	0..*	Configuration of one IP destination filter. **Tags:** atp.Status=draft
EthSwtStreamFilterIPSrcAddress	0..*	Configuration of one IP source filter. **Tags:** atp.Status=draft
EthSwtStreamFilterMACDestAddress	0..*	Configuration of one MAC destination filter. **Tags:** atp.Status=draft
EthSwtStreamFilterMACSrcAddress	0..*	Configuration of one MAC source filter. **Tags:** atp.Status=draft
EthSwtStreamFilterTcpDestPort	0..*	Configuration of a TCP destination port filter. **Tags:** atp.Status=draft
EthSwtStreamFilterTcpSrcPort	0..*	Configuration of a TCP source port filter. **Tags:** atp.Status=draft
EthSwtStreamFilterUdpDestPort	0..*	Configuration of a UDP destination port filter. **Tags:** atp.Status=draft
EthSwtStreamFilterUdpSrcPort	0..*	Configuration of a UDP source port filter. **Tags:** atp.Status=draft

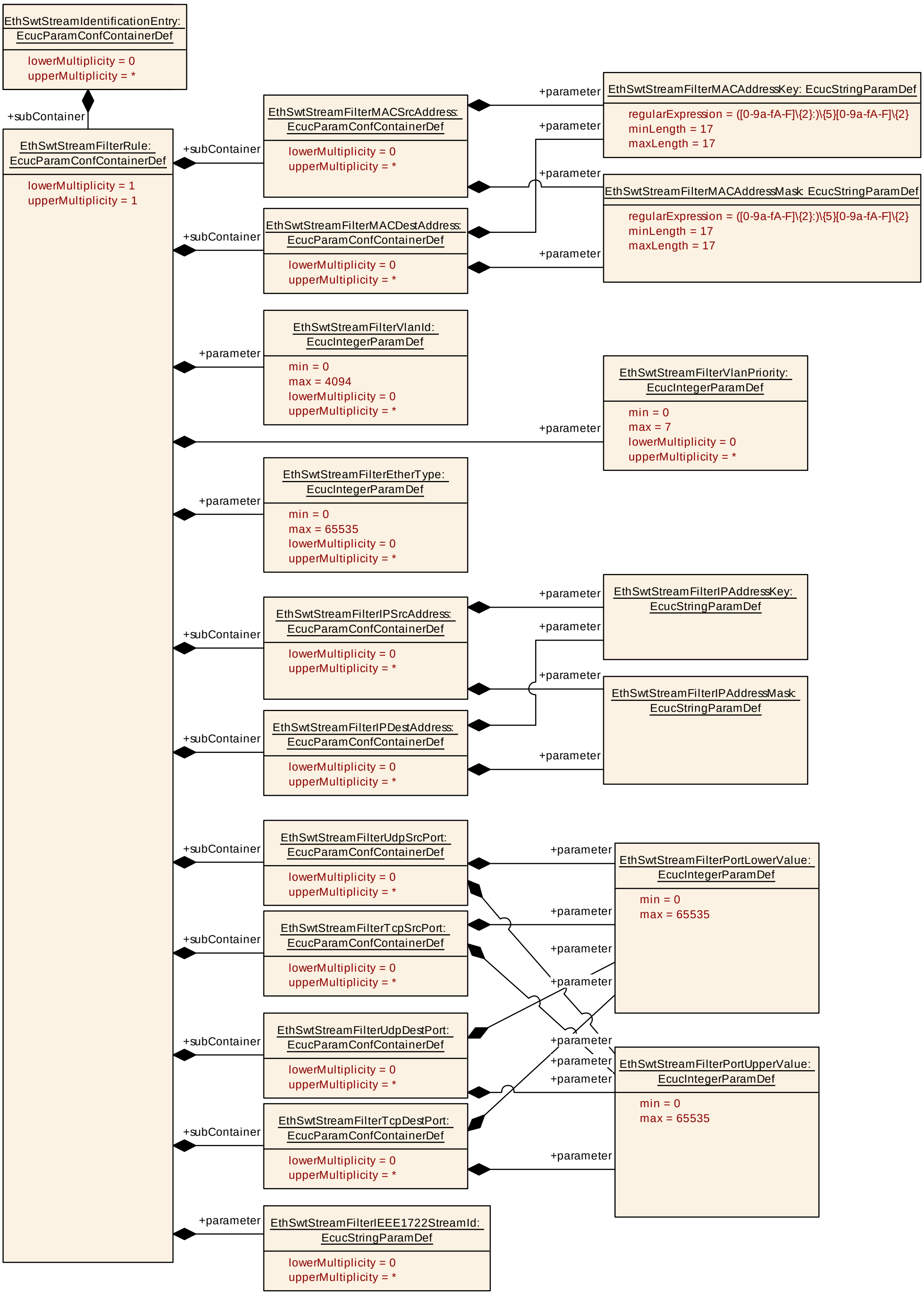

EthSwtStreamIdentificationEntry: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+subContainer
EthSwtStreamFilterRule: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = 1
+subContainer
EthSwtStreamFilterMACSrcAddress: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
+parameter
+parameter
+parameter
EthSwtStreamFilterMACAddressKey: EcucStringParamDef
regularExpression = ([0-9a-fA-F]\{2}:)\{5}[0-9a-fA-F]\{2}
minLength = 17
maxLength = 17
EthSwtStreamFilterMACAddressMask: EcucStringParamDef
regularExpression = ([0-9a-fA-F]\{2}:)\{5}[0-9a-fA-F]\{2}
minLength = 17
maxLength = 17
+subContainer
EthSwtStreamFilterMACDestAddress: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
EthSwtStreamFilterVlanId: EcucIntegerParamDef
min = 0
max = 4094
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
EthSwtStreamFilterVlanPriority: EcucIntegerParamDef
min = 0
max = 7
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
EthSwtStreamFilterEtherType: EcucIntegerParamDef
min = 0
max = 65535
lowerMultiplicity = 0
upperMultiplicity = *
+subContainer
EthSwtStreamFilterIPSrcAddress: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
+parameter
+parameter
+parameter
EthSwtStreamFilterIPAddressKey: EcucStringParamDef
EthSwtStreamFilterIPAddressMask: EcucStringParamDef
+subContainer
EthSwtStreamFilterIPDestAddress: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+subContainer
EthSwtStreamFilterUdpSrcPort: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
EthSwtStreamFilterPortLowerValue: EcucIntegerParamDef
min = 0
max = 65535
+subContainer
EthSwtStreamFilterTcpSrcPort: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
+parameter
+parameter
+subContainer
EthSwtStreamFilterUdpDestPort: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
+parameter
+parameter
EthSwtStreamFilterPortUpperValue: EcucIntegerParamDef
min = 0
max = 65535
+subContainer
EthSwtStreamFilterTcpDestPort: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
+parameter
EthSwtStreamFilterIEEE1722StreamId: EcucStringParamDef
lowerMultiplicity = 0
upperMultiplicity = *

Figure 10.20: EthSwtStreamFilterRule

10.1.45 EthSwtStreamFilterIPDestAddress

SWS Item	**[ECUC_EthSwt_00172]**		
Container Name	EthSwtStreamFilterIPDestAddress		
Parent Container	EthSwtStreamFilterRule		
Description	Configuration of one IP destination filter. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00173]**		
Parameter Name	EthSwtStreamFilterIPAddressKey		
Parent Container	EthSwtStreamFilterIPDestAddress		
Description	IP address key pattern. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

SWS Item	**[ECUC_EthSwt_00174]**		
Parameter Name	EthSwtStreamFilterIPAddressMask		
Parent Container	EthSwtStreamFilterIPDestAddress		
Description	IP address mask pattern. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

No Included Containers

10.1.46 EthSwtStreamFilterIPSrcAddress

SWS Item	**[ECUC_EthSwt_00171]**		
Container Name	EthSwtStreamFilterIPSrcAddress		
Parent Container	EthSwtStreamFilterRule		
Description	Configuration of one IP source filter. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00173]**		
Parameter Name	EthSwtStreamFilterIPAddressKey		
Parent Container	EthSwtStreamFilterIPSrcAddress		
Description	IP address key pattern. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

SWS Item	**[ECUC_EthSwt_00174]**		
Parameter Name	EthSwtStreamFilterIPAddressMask		
Parent Container	EthSwtStreamFilterIPSrcAddress		
Description	IP address mask pattern. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

No Included Containers

10.1.47 EthSwtStreamFilterMACDestAddress

SWS Item	**[ECUC_EthSwt_00165]**		
Container Name	EthSwtStreamFilterMACDestAddress		
Parent Container	EthSwtStreamFilterRule		
Description	Configuration of one MAC destination filter. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00166]**		
Parameter Name	EthSwtStreamFilterMACAddressKey		
Parent Container	EthSwtStreamFilterMACDestAddress		
Description	Specifies the 48-bit physical address (MAC address) key value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Length	17-17		
Regular Expression	([0-9a-fA-F]\{2}:)\{5}[0-9a-fA-F]\{2}		
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		

SWS Item	**[ECUC_EthSwt_00167]**		
Parameter Name	EthSwtStreamFilterMACAddressMask		
Parent Container	EthSwtStreamFilterMACDestAddress		
Description	Specifies the 48-bit physical address (MAC address) mask value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Length	17-17		
Regular Expression	([0-9a-fA-F]\{2}:)\{5}[0-9a-fA-F]\{2}		
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		

No Included Containers

10.1.48 EthSwtStreamFilterMACSrcAddress

SWS Item	**[ECUC_EthSwt_00164]**		
Container Name	EthSwtStreamFilterMACSrcAddress		
Parent Container	EthSwtStreamFilterRule		
Description	Configuration of one MAC source filter. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00166]**		
Parameter Name	EthSwtStreamFilterMACAddressKey		
Parent Container	EthSwtStreamFilterMACSrcAddress		
Description	Specifies the 48-bit physical address (MAC address) key value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Length	17-17		
Regular Expression	([0-9a-fA-F]\{2}:)\{5}[0-9a-fA-F]\{2}		
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		

SWS Item	**[ECUC_EthSwt_00167]**		
Parameter Name	EthSwtStreamFilterMACAddressMask		
Parent Container	EthSwtStreamFilterMACSrcAddress		
Description	Specifies the 48-bit physical address (MAC address) mask value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Length	17-17		
Regular Expression	([0-9a-fA-F]\{2}:)\{5}[0-9a-fA-F]\{2}		
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		

No Included Containers

10.1.49 EthSwtStreamFilterTcpDestPort

SWS Item	**[ECUC_EthSwt_00178]**		
Container Name	EthSwtStreamFilterTcpDestPort		
Parent Container	EthSwtStreamFilterRule		
Description	Configuration of a TCP destination port filter. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00179]**		
Parameter Name	EthSwtStreamFilterPortLowerValue		
Parent Container	EthSwtStreamFilterTcpDestPort		
Description	Definition of the filter port lower value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00180]**		
Parameter Name	EthSwtStreamFilterPortUpperValue		
Parent Container	EthSwtStreamFilterTcpDestPort		
Description	Definition of the filter port upper value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.1.50 EthSwtStreamFilterTcpSrcPort

SWS Item	**[ECUC_EthSwt_00176]**		
Container Name	EthSwtStreamFilterTcpSrcPort		
Parent Container	EthSwtStreamFilterRule		
Description	Configuration of a TCP source port filter. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00179]**		
Parameter Name	EthSwtStreamFilterPortLowerValue		
Parent Container	EthSwtStreamFilterTcpSrcPort		
Description	Definition of the filter port lower value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00180]**		
Parameter Name	EthSwtStreamFilterPortUpperValue		
Parent Container	EthSwtStreamFilterTcpSrcPort		
Description	Definition of the filter port upper value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.1.51 EthSwtStreamFilterUdpDestPort

SWS Item	**[ECUC_EthSwt_00177]**		
Container Name	EthSwtStreamFilterUdpDestPort		
Parent Container	EthSwtStreamFilterRule		
Description	Configuration of a UDP destination port filter. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00179]**		
Parameter Name	EthSwtStreamFilterPortLowerValue		
Parent Container	EthSwtStreamFilterUdpDestPort		
Description	Definition of the filter port lower value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00180]**		
Parameter Name	EthSwtStreamFilterPortUpperValue		
Parent Container	EthSwtStreamFilterUdpDestPort		
Description	Definition of the filter port upper value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.1.52 EthSwtStreamFilterUdpSrcPort

SWS Item	**[ECUC_EthSwt_00175]**		
Container Name	EthSwtStreamFilterUdpSrcPort		
Parent Container	EthSwtStreamFilterRule		
Description	Configuration of a UDP source port filter. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00179]**		
Parameter Name	EthSwtStreamFilterPortLowerValue		
Parent Container	EthSwtStreamFilterUdpSrcPort		
Description	Definition of the filter port lower value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthSwt_00180]**		
Parameter Name	EthSwtStreamFilterPortUpperValue		
Parent Container	EthSwtStreamFilterUdpSrcPort		
Description	Definition of the filter port upper value. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.1.53 EthSwtVlanMembership

SWS Item	**[ECUC_EthSwt_00199]**		
Container Name	EthSwtVlanMembership		
Parent Container	EthSwtConfig		
Description	Determines the membership of this Ethernet switch and the referenced ports to the virtual network, i.e. frames with this VID can be received and transmitted via the referenced ports.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00202]**		
Parameter Name	EthSwtVlanMembershipId		
Parent Container	EthSwtVlanMembership		
Description	Determines the VID of the virtual network this port belongs to.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4094		
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: ECU		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthSwtVlanMembershipPortRefEntry	0..255	Determines the VLAN membership of one referenced ports to the virtual network and the according forwarding type (NOT_SENT, SENT_UNTAGGED, SENT_TAGGED).

10.1.54 EthSwtVlanMembershipPortRefEntry

SWS Item	**[ECUC_EthSwt_00203]**		
Container Name	EthSwtVlanMembershipPortRefEntry		
Parent Container	EthSwtVlanMembership		
Description	Determines the VLAN membership of one referenced ports to the virtual network and the according forwarding type (NOT_SENT, SENT_UNTAGGED, SENT_TAGGED).		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**[ECUC_EthSwt_00026]**		
Parameter Name	EthSwtVlanForwardingType		
Parent Container	EthSwtVlanMembershipPortRefEntry		
Description	Defines how the message with a specific VLAN-ID at the referenced port shall be handled.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ETHSWT_NOT_SENT	The message with the specific VLAN Id shall not be sent at the referenced port.	
	ETHSWT_SENT_TAGGED	The message with the specific VLAN Id shall be sent with its VLAN Id at the referenced port.	
	ETHSWT_SENT_UNTAGGED	The message with the specific VLAN Id shall be sent untagged at the referenced port.	
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: ECU		

SWS Item	**[ECUC_EthSwt_00204]**		
Parameter Name	EthSwtVlanMembershipPortRef		
Parent Container	EthSwtVlanMembershipPortRefEntry		
Description	Reference to one port the VLAN shall be assigned to.		
Multiplicity	1		
Type	Reference to EthSwtPort		
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: ECU		

No Included Containers

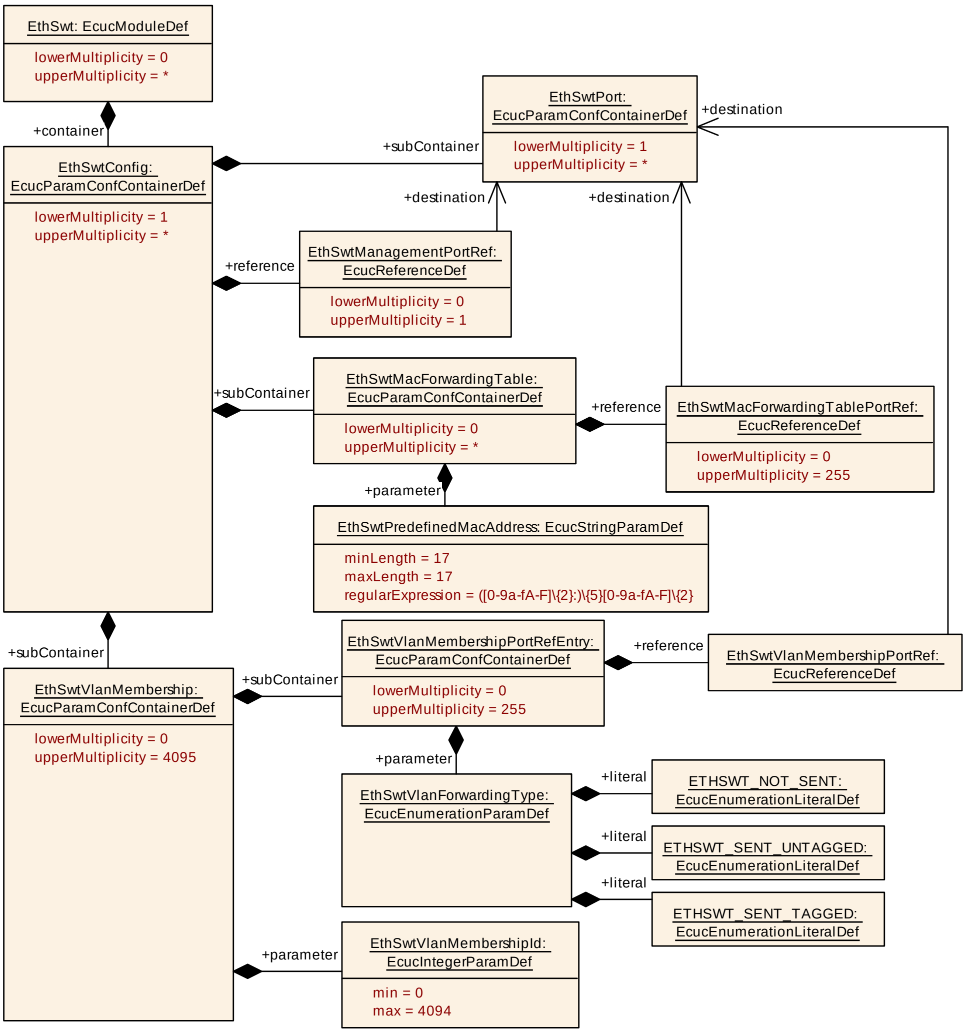

EthSwt: EcucModuleDef
lowerMultiplicity = 0
upperMultiplicity = *
+container
EthSwtConfig: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
EthSwtPort: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+destination
+destination
+destination
+reference
EthSwtManagementPortRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
+subContainer
EthSwtMacForwardingTable: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+reference
EthSwtMacForwardingTablePortRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 255
+parameter
EthSwtPredefinedMacAddress: EcucStringParamDef
minLength = 17
maxLength = 17
regularExpression = ([0-9a-fA-F]\{2}:)\{5}[0-9a-fA-F]\{2}
+subContainer
EthSwtVlanMembershipPortRefEntry: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 255
+reference
EthSwtVlanMembershipPortRef: EcucReferenceDef
+subContainer
EthSwtVlanMembership: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 4095
+parameter
EthSwtVlanForwardingType: EcucEnumerationParamDef
+literal
ETHSWT_NOT_SENT: EcucEnumerationLiteralDef
+literal
ETHSWT_SENT_UNTAGGED: EcucEnumerationLiteralDef
+literal
ETHSWT_SENT_TAGGED: EcucEnumerationLiteralDef
+parameter
EthSwtVlanMembershipId: EcucIntegerParamDef
min = 0
max = 4094

Figure 10.21: EthSwtVlanMembershipPortRefEntry

10.2 Constraints

[SWS_EthSwt_CONSTR_00413] ⌈The module will operate as an independent instance in each of the partitions (see EthSwtEcucPartitionRef), means the called API will only target the partition it is called in.⌋*()*

[SWS_EthSwt_CONSTR_00411] ⌈The ECUC partitions referenced by EthSwtConfigEcucPartitionRef shall be a subset of the ECUC partitions referenced by EthSwtEcucPartitionRef.⌋*()*

[SWS_EthSwt_CONSTR_00412] ⌈EthSwtConfig, EthCtrlConfig and EthTrcvConfig of one communication channel shall all reference the same ECUC partition.⌋*()*

[SWS_EthSwt_CONSTR_00438] ⌈If `EthSwtEcucPartitionRef` references one or more ECUC partitions, `EthSwtConfigEcucPartitionRef` shall have a multiplicity of one and reference one of these ECUC partitions as well.⌋*()*

A Change History

Please note that the lists in this chapter also include constraints and specification items that have been removed from the specification in a later version. These constraints and specification items do not appear as hyperlinks in the document.

A.1 Traceable item history of this document according to AUTOSAR Release R22-11

A.1.1 Added Specification Items in R22-11

Number	Heading
[SWS_EthSwt_00455]	
[SWS_EthSwt_00460]	
[SWS_EthSwt_00461]	
[SWS_EthSwt_00462]	
[SWS_EthSwt_00463]	
[SWS_EthSwt_00465]	
[SWS_EthSwt_00466]	
[SWS_EthSwt_00467]	
[SWS_EthSwt_00469]	
[SWS_EthSwt_00471]	
[SWS_EthSwt_00472]	
[SWS_EthSwt_00475]	
[SWS_EthSwt_00476]	
[SWS_EthSwt_00477]	
[SWS_EthSwt_00478]	
[SWS_EthSwt_00479]	
[SWS_EthSwt_00480]	
[SWS_EthSwt_00481]	
[SWS_EthSwt_00482]	
[SWS_EthSwt_00483]	
[SWS_EthSwt_00484]	
[SWS_EthSwt_00486]	
[SWS_EthSwt_00487]	
[SWS_EthSwt_00490]	
[SWS_EthSwt_00491]	
[SWS_EthSwt_00492]	
[SWS_EthSwt_00493]	

▽

△

Number	Heading
[SWS_EthSwt_00494]	
[SWS_EthSwt_91124]	
[SWS_EthSwt_91125]	
[SWS_EthSwt_91126]	
[SWS_EthSwt_91127]	
[SWS_EthSwt_91128]	
[SWS_EthSwt_91129]	
[SWS_EthSwt_91130]	
[SWS_EthSwt_91131]	
[SWS_EthSwt_91132]	
[SWS_EthSwt_91133]	
[SWS_EthSwt_91134]	
[SWS_EthSwt_91135]	
[SWS_EthSwt_91136]	
[SWS_EthSwt_91137]	
[SWS_EthSwt_91138]	
[SWS_EthSwt_91139]	
[SWS_EthSwt_CONSTR_00450]	
[SWS_EthSwt_CONSTR_00451]	
[SWS_EthSwt_CONSTR_00452]	
[SWS_EthSwt_CONSTR_00453]	
[SWS_EthSwt_CONSTR_00454]	
[SWS_EthSwt_CONSTR_00456]	
[SWS_EthSwt_CONSTR_00457]	
[SWS_EthSwt_CONSTR_00458]	
[SWS_EthSwt_CONSTR_00459]	
[SWS_EthSwt_CONSTR_00464]	
[SWS_EthSwt_CONSTR_00468]	
[SWS_EthSwt_CONSTR_00470]	

▽

△

Number	Heading
[SWS_EthSwt_CONSTR_00473]	
[SWS_EthSwt_CONSTR_00474]	
[SWS_EthSwt_CONSTR_00485]	
[SWS_EthSwt_CONSTR_00488]	
[SWS_EthSwt_CONSTR_00489]	
[SWS_EthSwt_CONSTR_00495]	
[SWS_EthSwt_CONSTR_00496]	

Table A.1: Added Specification Items in R22-11

A.1.2 Changed Specification Items in R22-11

Number	Heading
[SWS_EthSwt_00001]	
[SWS_EthSwt_00002]	
[SWS_EthSwt_00006]	
[SWS_EthSwt_00009]	
[SWS_EthSwt_00018]	
[SWS_EthSwt_00019]	
[SWS_EthSwt_00023]	
[SWS_EthSwt_00025]	
[SWS_EthSwt_00026]	
[SWS_EthSwt_00031]	
[SWS_EthSwt_00032]	
[SWS_EthSwt_00037]	
[SWS_EthSwt_00038]	
[SWS_EthSwt_00044]	
[SWS_EthSwt_00045]	
[SWS_EthSwt_00051]	
[SWS_EthSwt_00052]	
[SWS_EthSwt_00058]	
[SWS_EthSwt_00060]	

▽

△

Number	Heading
[SWS_EthSwt_00061]	
[SWS_EthSwt_00086]	
[SWS_EthSwt_00087]	
[SWS_EthSwt_00091]	
[SWS_EthSwt_00092]	
[SWS_EthSwt_00098]	
[SWS_EthSwt_00106]	
[SWS_EthSwt_00111]	
[SWS_EthSwt_00114]	
[SWS_EthSwt_00117]	
[SWS_EthSwt_00118]	
[SWS_EthSwt_00123]	
[SWS_EthSwt_00125]	
[SWS_EthSwt_00126]	
[SWS_EthSwt_00127]	
[SWS_EthSwt_00128]	
[SWS_EthSwt_00132]	
[SWS_EthSwt_00133]	
[SWS_EthSwt_00134]	
[SWS_EthSwt_00135]	
[SWS_EthSwt_00136]	
[SWS_EthSwt_00154]	
[SWS_EthSwt_00156]	
[SWS_EthSwt_00157]	
[SWS_EthSwt_00162]	
[SWS_EthSwt_00164]	
[SWS_EthSwt_00165]	
[SWS_EthSwt_00172]	
[SWS_EthSwt_00173]	
[SWS_EthSwt_00178]	
[SWS_EthSwt_00179]	
[SWS_EthSwt_00180]	
[SWS_EthSwt_00181]	
[SWS_EthSwt_00182]	
[SWS_EthSwt_00183]	
[SWS_EthSwt_00187]	
[SWS_EthSwt_00188]	
[SWS_EthSwt_00193]	
[SWS_EthSwt_00194]	

▽

△

Number	Heading
[SWS_EthSwt_00196]	
[SWS_EthSwt_00197]	
[SWS_EthSwt_00198]	
[SWS_EthSwt_00199]	
[SWS_EthSwt_00203]	
[SWS_EthSwt_00204]	
[SWS_EthSwt_00206]	
[SWS_EthSwt_00211]	
[SWS_EthSwt_00212]	
[SWS_EthSwt_00216]	
[SWS_EthSwt_00217]	
[SWS_EthSwt_00221]	
[SWS_EthSwt_00222]	
[SWS_EthSwt_00226]	
[SWS_EthSwt_00227]	
[SWS_EthSwt_00228]	
[SWS_EthSwt_00231]	
[SWS_EthSwt_00233]	
[SWS_EthSwt_00234]	
[SWS_EthSwt_00235]	
[SWS_EthSwt_00240]	
[SWS_EthSwt_00241]	
[SWS_EthSwt_00242]	
[SWS_EthSwt_00243]	
[SWS_EthSwt_00245]	
[SWS_EthSwt_00372]	
[SWS_EthSwt_00373]	
[SWS_EthSwt_00378]	
[SWS_EthSwt_00398]	
[SWS_EthSwt_00416]	
[SWS_EthSwt_00417]	
[SWS_EthSwt_00418]	
[SWS_EthSwt_00419]	
[SWS_EthSwt_00420]	
[SWS_EthSwt_00430]	
[SWS_EthSwt_00431]	
[SWS_EthSwt_00434]	
[SWS_EthSwt_00440]	
[SWS_EthSwt_91000]	

▽

△

Number	Heading
[SWS_EthSwt_91001]	
[SWS_EthSwt_91002]	
[SWS_EthSwt_91003]	
[SWS_EthSwt_91004]	
[SWS_EthSwt_91005]	
[SWS_EthSwt_91006]	
[SWS_EthSwt_91007]	
[SWS_EthSwt_91008]	
[SWS_EthSwt_91009]	
[SWS_EthSwt_91010]	
[SWS_EthSwt_91011]	
[SWS_EthSwt_91012]	
[SWS_EthSwt_91013]	
[SWS_EthSwt_91014]	
[SWS_EthSwt_91015]	
[SWS_EthSwt_91016]	
[SWS_EthSwt_91017]	
[SWS_EthSwt_91018]	
[SWS_EthSwt_91019]	
[SWS_EthSwt_91020]	
[SWS_EthSwt_91021]	
[SWS_EthSwt_91022]	
[SWS_EthSwt_91023]	
[SWS_EthSwt_91024]	
[SWS_EthSwt_91025]	
[SWS_EthSwt_91028]	
[SWS_EthSwt_91029]	
[SWS_EthSwt_91030]	
[SWS_EthSwt_91031]	
[SWS_EthSwt_91032]	
[SWS_EthSwt_91033]	
[SWS_EthSwt_91034]	
[SWS_EthSwt_91035]	
[SWS_EthSwt_91036]	
[SWS_EthSwt_91037]	
[SWS_EthSwt_91038]	
[SWS_EthSwt_91039]	
[SWS_EthSwt_91040]	
[SWS_EthSwt_91050]	

▽

△

Number	Heading
[SWS_EthSwt_91104]	
[SWS_EthSwt_91123]	

Table A.2: Changed Specification Items in R22-11

A.1.3 Deleted Specification Items in R22-11

none

A.1.4 Added Constraints in R22-11

none

A.1.5 Changed Constraints in R22-11

none

A.1.6 Deleted Constraints in R22-11

none

A.2 Traceable item history of this document according to AUTOSAR Release R23-11

A.2.1 Added Specification Items in R23-11

Number	Heading
[SWS_EthSwt_00163]	
[SWS_EthSwt_00450]	
[SWS_EthSwt_00451]	
[SWS_EthSwt_00456]	
[SWS_EthSwt_00459]	
[SWS_EthSwt_00500]	
[SWS_EthSwt_00501]	

▽

△

Number	Heading
[SWS_EthSwt_00502]	
[SWS_EthSwt_00503]	
[SWS_EthSwt_00504]	
[SWS_EthSwt_00601]	
[SWS_EthSwt_00602]	
[SWS_EthSwt_00604]	
[SWS_EthSwt_00605]	
[SWS_EthSwt_00606]	
[SWS_EthSwt_00607]	
[SWS_EthSwt_00608]	
[SWS_EthSwt_00609]	
[SWS_EthSwt_00610]	
[SWS_EthSwt_00611]	
[SWS_EthSwt_00612]	
[SWS_EthSwt_00613]	
[SWS_EthSwt_00614]	
[SWS_EthSwt_91041]	Definition of API function EthSwt_SetStreamHandleIdxConfiguration
[SWS_EthSwt_91042]	Definition of API function EthSwt_GetStreamHandleIdxStatistics
[SWS_EthSwt_91043]	Definition of API function EthSwt_ExtractStreamHandleIdx

Table A.3: Added Specification Items in R23-11

A.2.2 Changed Specification Items in R23-11

Number	Heading
[SWS_EthSwt_00002]	Definition of imported datatypes of module EthSwt
[SWS_EthSwt_00016]	
[SWS_EthSwt_00133]	
[SWS_EthSwt_00179]	
[SWS_EthSwt_00455]	
[SWS_EthSwt_00465]	
[SWS_EthSwt_00467]	
[SWS_EthSwt_00469]	
[SWS_EthSwt_00471]	
[SWS_EthSwt_00472]	
[SWS_EthSwt_00475]	
[SWS_EthSwt_00476]	

▽

△

Number	Heading
[SWS_EthSwt_00478]	
[SWS_EthSwt_00493]	

Table A.4: Changed Specification Items in R23-11

A.2.3 Deleted Specification Items in R23-11

Number	Heading
[SWS_EthSwt_00136]	
[SWS_EthSwt_00162]	
[SWS_EthSwt_00181]	
[SWS_EthSwt_00466]	
[SWS_EthSwt_00490]	

Table A.5: Deleted Specification Items in R23-11

A.2.4 Added Constraints in R23-11

Number	Heading
[SWS_EthSwt_CONSTR_00602]	
[SWS_EthSwt_CONSTR_00603]	

Table A.6: Added Constraints in R23-11

A.2.5 Changed Constraints in R23-11

Number	Heading
[SWS_EthSwt_CONSTR_00453]	

▽

△

Number	Heading
[SWS_EthSwt_CONSTR_00454]	
[SWS_EthSwt_CONSTR_00464]	
[SWS_EthSwt_CONSTR_00468]	

Table A.7: Changed Constraints in R23-11

A.2.6 Deleted Constraints in R23-11

Number	Heading
[SWS_EthSwt_CONSTR_00450]	
[SWS_EthSwt_CONSTR_00451]	
[SWS_EthSwt_CONSTR_00456]	
[SWS_EthSwt_CONSTR_00458]	
[SWS_EthSwt_CONSTR_00459]	
[SWS_EthSwt_CONSTR_00473]	

▽

△

Number	Heading
[SWS_EthSwt_CONSTR_00474]	
[SWS_EthSwt_CONSTR_00488]	
[SWS_EthSwt_CONSTR_00496]	

Table A.8: Deleted Constraints in R23-11