

<b>Document Title</b>	Explanation of Classic Platform Software Architectural Decisions
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	1028

<b>Document Status</b>	published
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	R21-11

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2021-11-25	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Initial release</li> </ul>

## 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.

## Table of Contents

1	Introduction	4
1.1	Objectives	4
1.2	Scope	4
2	Definition of terms and acronyms	5
3	Related Documentation	6
4	Overview	7
5	Architectural Decisions	8
5.1	Ordering of structure elements	8
5.2	Header File Cleanup	8

# 1 Introduction

This explanatory document provides additional information on architectural decisions made for the AUTOSAR Classic Platform standard.

## 1.1 Objectives

The main objective of this document is to provide a documentation of architectural decisions made for the AUTOSAR Classic Platform to make such decisions comprehensible and reviewable in the future and ultimately get a more maintainable standard.

## 1.2 Scope

This document covers decisions made for the software architecture of the AUTOSAR Classic Platform. The main audience of this document are architects of the AUTOSAR Classic Platform as well as members of other working groups.

## 2 Definition of terms and acronyms

There are currently no terms and acronyms which have a local scope and therefore are not contained in the AUTOSAR glossary [1].

### 3 Related Documentation

This document provides an overview of the architectural decisions that have been made for the AUTOSAR Classic Platform and their rationale. A high-level overview of the current AUTOSAR Classic Platform architecture is provided in [[2](#), EXP\_LayeredSoftwareArchitecture].

## 4 Overview

This chapter provides an overview of the organization and structure of decisions listed in this document. All decisions are organized in a table as described below.

<b>Date of approval</b>	Date when the decision has been approved by CF-CCB.
<b>Decision</b>	The decision itself. Impact or direct consequences (for example, changes to interfaces) of the decision are not documented in the normal case. Changes are requested during the roll-out process after the decision has been made.
<b>Rationale</b>	A rationale for the decision.
<b>Category</b>	Category of the decision.
<b>Application affected</b>	States if the decision has a direct impact on existing applications.
<b>Assumptions</b>	Lists the assumptions that have been made before making the decision itself. These assumptions are documented in order to be able to review decisions in the future and check if some assumptions probably no longer hold.
<b>Constraints</b>	Provides an overview of the constraints that were identified to have an impact on possible solutions. The constraints are also documented in order to be reference points for future reviews of the decision.
<b>Alternatives</b>	Lists the alternatives that were considered and a rationale why they are worse than the decision that has been made.
<b>Remarks</b>	Lists remarks on the decisions.
<b>Related requirements</b>	Lists requirements related to the decision.

## 5 Architectural Decisions

This chapter lists architectural decisions that have been made for the AUTOSAR Classic Platform.

### 5.1 Ordering of structure elements

<b>Date of approval</b>	2021
<b>Decision</b>	The order of structure elements as defined by the SWS is considered as part of the standard. Implementation specific optimizations, e.g. a re-ordering of structure elements by size to avoid alignment gaps, are therefore not standard compliant.
<b>Rationale</b>	Object code interoperability could be jeopardized by deviating structure type definitions.
<b>Category</b>	None
<b>Application affected</b>	Yes
<b>Assumptions</b>	Structure elements are usually accessed via name, which means that the order shouldn't matter. There are however valid use-cases like the initialization of structures without designated initializers (e.g. <code>my_struct x = {0, 42}</code> ) where no element names are involved at all.
<b>Constraints</b>	None
<b>Alternatives</b>	The order of structure elements in the SWS is not prescribed by the standard. An implementation is free to do any desired re-ordering.
<b>Remarks</b>	In resource optimized implementations, structure elements are usually ordered by size to avoid alignment gaps. This helps to increase efficiency and reduces memory consumption. Nevertheless some structures defined in AUTOSAR do not follow this rule.
<b>Related requirements</b>	None

### 5.2 Header File Cleanup

<b>Date of approval</b>	2017 (before ArcDecision process was introduced)
<b>Decision</b>	<p>There shall be only 3 types of headers:</p> <ol style="list-style-type: none"> <li>1. The module header (e.g. <code>NvM.h</code>, <code>CanIf.h</code>, <code>EcuM.h</code>, ...)</li> <li>2. The private header between two modules (e.g. <code>BswM_Sd.h</code>, <code>Adc_SchM.h</code>, <code>Dcm_Externals.h</code>, ...)</li> <li>3. The shared header (e.g. <code>PlatformTypes.h</code>, <code>StandardTypes.h</code>, <code>Can_GeneralTypes.h</code>, <code>ComStackTypes.h</code>, ...)</li> </ol>





△

	<p style="text-align: center;">△</p> <p>Any additional headers are no longer necessary and are dropped/removed from the SWS. This means that they are no longer standardized. An implementation is however free to have such headers for its own purpose.</p> <p><b>Rules:</b></p> <ul style="list-style-type: none"> <li>• All header files are self-contained</li> <li>• A module which uses types of another BSW in its own interface must consider moving such types into a shared header (Exception: types of service interfaces which are generated by the RTE and are available via <code>Rte_&lt;Mip&gt;.h</code>)</li> <li>• A library cannot have private headers by definition</li> <li>• Shared headers only consist of types and enums (No function prototypes...)</li> <li>• Shared headers do never depend on other module or private headers</li> <li>• For callouts to integration code or CDDs: The prototypes are available via <code>&lt;Mip&gt;_Externals.h</code></li> </ul> <p><b>Consequences:</b></p> <ul style="list-style-type: none"> <li>• The tables for types and APIs (C interface) shall have a line "Available via" to indicate the name of the header which exports the type/function</li> </ul>
<b>Rationale</b>	This is sufficient for an external view to answer the question which header is needed by a user.
<b>Category</b>	None
<b>Application affected</b>	No
<b>Assumptions</b>	None
<b>Constraints</b>	None
<b>Alternatives</b>	Keep the current BSW implementation focused header file concept.
<b>Remarks</b>	None
<b>Related requirements</b>	None

## References

- [1] Glossary  
AUTOSAR\_TR\_Glossary
- [2] Layered Software Architecture  
AUTOSAR\_EXP\_LayeredSoftwareArchitecture