| Document Title | Foundation Release Overview |
|---|---|
| **Document Owner** | AUTOSAR |
| **Document Responsibility** | AUTOSAR |
| **Document Identification No** | 781 |

| | |
|---|---|
| **Document Status** | Final |
| **Part of AUTOSAR Standard** | Foundation |
| **Part of Standard Release** | 1.3.0 |
| **Release Life Cycle Status** | R1.3 is in Evolution, R1.3 supersedes R1.2 |

## Document Change History

| Date | Release | Changed by | Description |
|---|---|---|---|
| 2017-12-08 | 1.3.0 | AUTOSAR Release Management | Updated according to Release 1.3.0 |
| 2017-10-27 | 1.2.0 | AUTOSAR Release Management | Updated according to Release 1.2.0 |
| 2017-03-31 | 1.1.0 | AUTOSAR Release Management | Updated according to Release 1.1.0 |
| 2016-11-30 | 1.0.0 | AUTOSAR Release Management | Initial release |

# Table of Contents

# 1 Introduction

## 1.1 Scope of this document

This document provides an overview of the complement of AUTOSAR specifications of the AUTOSAR standard “Foundation” comprising the initial Release 1.3.0 and its latest Revision.

## 1.2 AUTOSAR standards

### 1.2.1 Introduction

AUTOSAR addresses with its standards a wide range of use cases in automotive software development. These use cases have different requirements and lead to different technical solutions.

Packaging its deliverables into different “standards”

- eases the access to AUTOSAR solutions for users
- allows AUTOSAR to scale with market needs

### 1.2.2 Definition

An AUTOSAR standard is a consistent set of AUTOSAR deliverables, which are released at the same time. AUTOSAR deliverables can, but are not limited to be of the following kinds:

- textual explanations
- textual specifications
- test specification
- source code
- other formal or semi-formal textual formats (e.g. ARXML, UML models, XML Schemata)

Each AUTOSAR Standard has its own release schedule. At the time of release, AUTOSAR ensures that the dependencies are fulfilled when a standard depends on another.

### 1.2.3 Overview on AUTOSAR’s standards

AUTOSAR delivers the following standards:

| Cluster / Standard | Abbreviation |
|---|---|
| Classic Platform | CP |
| Adaptive Platform | AP |
| Foundation | FO |

#### 1.2.3.1 Foundation

The purpose of the Foundation standard is to enforce interoperability between the AUTOSAR platforms.

Foundation contains common requirements and technical specifications (e.g. protocols) shared between the AUTOSAR platforms.

#### 1.2.3.2 Classic Platform

The Classic Platform is AUTOSAR's solution for embedded systems with hard real-time and safety constraints.

#### 1.2.3.3 Adaptive Platform

The Adaptive Platform is AUTOSAR's solution for high-performance computing ECUs to build safety-related systems for use cases such as highly automated and autonomous driving.

### 1.2.4 Dependencies between Standards

Each release of Classic, Adaptive Platform relies on a dedicated version of Foundation. The specific dependency is documented in the release overview of the respective standard.

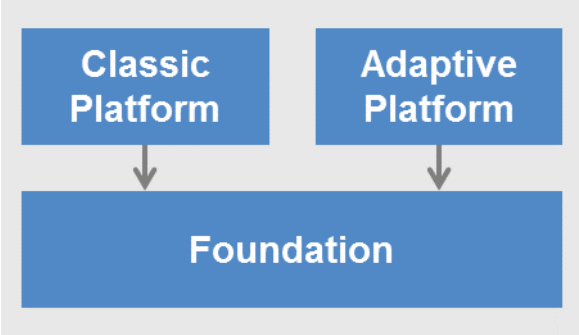


**Figure 1: Dependencies of AUTOSAR Standards**

### 1.2.5 Overview of AUTOSAR schema versions and corresponding valid AUTOSAR releases

| Schema Version | Classic Platform Release | Adaptive Platform Release |
|---|---|---|
| AUTOSAR_00042 | R 4.3.0 | 17-03 |
| AUTOSAR_00043 | R 4.3.0 | 17-10 |
| AUTOSAR_00044 | R 4.3.1 | 17-10 |

## 1.3 Content of chapters

This document is structured as follows:

- Chapter 2 provides a list of documentation references.
- Chapter 3 contains the overview of specifications comprising the AUTOSAR Foundation Release 1.3.0 in its latest Revision. This chapter is structured according to the clusters being in use in AUTOSAR Foundation Release 1.3.0.
- Chapter 4 provides a summary of changes e.g. in case a document has been migrated from another standard like the Classic Platform.
- Chapter 5 contains remarks about known technical deficiencies.
- Chapter 6 contains the detailed revision history of all released specifications.
- Chapter 7.1 provides a set of definitions aimed to increase the understanding of the content of this document and the AUTOSAR Foundation Release 1.3.0.

# 2 Related documentation

1) AUTOSAR specifications in general
2) Change Documentation
3) Glossary

# 3 Specification overview

The published specifications are divided up into the following clusters:

- Release Documentation
- General
- Diagnostics
- Methodology and Templates
- Communication Management
- Health Monitoring
- Protocols

The assignment of the specifications to those clusters is shown below.

| Long Name | File Name | Life cycle changes |
|---|---|---|
| **Cluster: Cluster: Release Documentation** | | |
| Foundation Release Overview | AUTOSAR_TR_Foundation ReleaseOverview | |
| AUTOSAR Foundation Specification Hashes | AUTOSAR_TR_Foundation SpecificationHashes | |
| Foundation Change Documentation | AUTOSAR_TR_Foundation ChangeDocumentation | |
| **Cluster: General** | | |
| Main Requirements | AUTOSAR_RS_Main | |
| Glossary | AUTOSAR_TR_Glossary | |
| Project Objectives | AUTOSAR_RS_Project Objectives | |
| Foundation UML Model | AUTOSAR_MOD_Foundation UMLModel | |
| **Cluster: Diagnostics** | | |
| Requirements on Diagnostic | AUTOSAR_SRS_Diagnostic | |
| Requirements on Log and Trace | AUTOSAR_RS_LogAndTrace | |
| **Cluster: Methodology and Templates** | | |
| Requirements on Methodology | AUTOSAR_RS_Methodology | |
| **Cluster: Communication Management** | | |
| Requirements on E2E | AUTOSAR_RS_E2E | |
| **Cluster: Health Monitoring** | | |
| Requirements on Health Monitoring | AUTOSAR_RS_Health Monitoring | |
| **Cluster: Protocols** | | |
| SOME/IP Protocol Specification | AUTOSAR_PRS_SOMEIP Protocol | |
| Log and Trace Protocol Specification | AUTOSAR_PRS_LogAndTrace Protocol | |
| Requirements on SOME/IP Protocol | AUTOSAR_RS_SOMEIP Protocol | |
| Requirements on SOME/IP Service Discovery Protocol | AUTOSAR_RS_SOMEIP ServiceDiscoveryProtocol | |

| Long Name | File Name | Life cycle changes |
|---|---|---|
| SOME/IP Service Discovery Protocol Specification | AUTOSAR_PRS_SOMEIP ServiceDiscoveryProtocol | |
| Remote Event Communication Protocol Specification | AUTOSAR_PRS_RemoteEvent CommunicationProtocol | |
| E2E Protocol Specification | AUTOSAR_PRS_E2EProtocol | |

# 4 Summary of changes

This chapter contains a summary of changes which were implemented. This can have the following sources:

- Regular maintenance of document
- Documents have been migrated from the Classic Platform R4.3.1 or Adaptive Platform to the Foundation
- New documents have been created and are first time released.
- Documents went through a major rework

## 4.1 Release 1.3.0

The purpose of the Foundation standard is to enforce interoperability between the AUTOSAR platforms.

With the current release, this goal has been pursued once more. Foundation contains common requirements and technical specifications (e.g. protocols) shared between the AUTOSAR platforms.

### 4.1.1 Concepts

No concepts have been introduced with FO R1.3.0.

### 4.1.2 Specifications

#### 4.1.2.1 New Specifications

- No new specifications in R1.3.0

#### 4.1.2.2 Migrated Specifications

With this release, the following specifications were moved from Adaptive Platform to the Foundation standard:

- No migration in R1.3.0

With this release, the following specifications were moved from Classic Platform to the Foundation standard:

- No migration in R4.3.1

##### 4.1.2.2.1 Obsolete Specifications

The following specification is set to status “obsolete” in this release:

- No specifications were set to “obsolete”.

### 4.1.3 Release Documentation

There were no major changes regarding the Release Documentation.

# 5 Remarks to known technical deficiencies

The technical deficiencies per specification are – if applicable – mentioned inside the respective specification in a chapter called “Known Limitations” which is located after the table of contents.

There are the following technical deficiencies to be mentioned which are not related to a specific specification:

- none

## 5.1 Known technical deficiencies per document

- **E2E Protocol Specification (UID 849, PRS):** E2E communication protection is limited to periodic or semi-periodic data communication paradigm, where the receiver (subscriber) has an expectancy on the regular reception of data and in case of communication loss/timeout or error, performs an error handling.
  Data communication is called sender/receiver in Classic Platform, and it is called event communication in Adaptive Platform.
  Note that the word event is a bit confusing as a periodic communication is required. This means, a protection of client-server (methods) as well as non-periodic data communication (e.g. transmission only on occurrence of a specific event) are not supported by E2E communication protection.

# 6 Revision history

## 6.1 Release 1.3.0

Revision 0 of Release 1.3.0 has been released on the 8th of December 2017. The following deliverables had major changes.

| Name | Specification history entry |
|---|---|
| Project Objectives | • No content changes |
| Requirements on Diagnostic | • Minor corrections<br>• Update of requirements tracing |
| Glossary | • No content changes |
| Main Requirements | • New requirement for logging |
| Requirements on Log and Trace | • No content changes |
| Requirements on Methodology | • Migration of the Classic Platform requirements document to standard “Foundation” finalized<br>• Enhanced quality of requirements<br>• New requirement added which applies to both platforms<br>• New requirement for Classic Platform only added |
| SOME/IP Protocol Specification | • No content changes |
| Log and Trace Protocol Specification | • No content changes |
| Requirements on SOME/IP Protocol | • No content changes |
| Requirements on SOME/IP Service Discovery Protocol | • No content changes |
| SOME/IP Service Discovery Protocol Specification | • Minor changes |
| Remote Event Communication Protocol Specification | • No content changes |
| Requirements on E2E | • No content changes |
| E2E Protocol Specification | • No content changes |
| Requirements on Health Monitoring | • No content changes |

More specifications might have been changed, which are not listed here. Those specifications have then only “minor corrections, clarifications or editorial changes; for details please refer to the Change Documentation” [3].

# 7 Appendix

## 7.1 Definitions

As far as not explained in this chapter, a collection of AUTOSAR definitions is provided in 3).

### 7.1.1 Release number

AUTOSAR applies a two-digit numbering scheme Rx.y to identify Releases. Its primary purpose is to identify a Release as a major (upgrade, can contain non-backward-compatible extensions) or as minor (update, backward compatible extensions) Release. Referring to previous Releases (e.g. R2.0), incrementing the first digit “x” does identify a Release as major, whereas incrementing “y” will mark a Release as only minor by nature.

### 7.1.2 Revision number

The Revision Number was first time introduced with Release 2.1 and extends the Release Numbering scheme as explained in section 7.1.1. Combined with the Release Number, the Revision Number shall:

1) Precisely identify the actual content (set of specifications) of a given Release.
2) As depicted in every specification, precisely identify a given specification (with its unique name and three-digit version ID) as being part of the Release.

Item 1) addresses the fact that the set of specifications comprising a Release (in the meaning of a baseline) is rarely established once at a certain point in time (“Big Bang”), but rather evolves and/or varies over a certain timeframe. The maximum duration, which is limited by the timeframe, a Release is declared as “valid” by the AUTOSAR Partnership (see section 7.1.3).

Hence with Item 1), a major prerequisite will be put in place to enable the Standard Maintenance as planned by the AUTOSAR Partnership. In general, the primary objective is to avoid the provision of an additional – previously not planned – Release in case only one or a few specifications were to be modified as part of the Standard Maintenance. Conversely, without the application of a Revision Number, if the AUTOSAR partnership wants to avoid the provision of (an) additional intermediate Release(s), one would have to defer the introduction of any changes until the next planned Release – even in case of changes urgently needed by the applicants of the AUTOSAR Standard.

Item 2) is complementary to Item 1) in that for every specification a unique identifier is provided upon which Revision a) a specification was either 1$^{st}$ time added to/removed from a Release or b) a specification was modified as being part of one and the same Release, as long the latter is valid and therefore subject to Standard Maintenance.

Hence with item 2), the combination of Release and Revision Number in a specification can be interpreted either as a) “specification was (1st time) added to the Release x.y Rev n” or b) as “specification was modified as part of Release x.y Rev m”, with m > n.

Conversely, the Revision number will only change for specifications subject to addition or modification of a valid Release (baseline). After their 1st time addition to the Release (baseline), it will not change for specifications which are not modified.

In the light of the above provided background, as an additional remark, the Revision Number will only be applied for each specification’s Release version, i.e. it will not be applied to working versions.

### 7.1.3 Release life cycle of a major release

Each major release goes through four consecutive steps within its lifecycle:

1. Development: Between start of life cycle and the initial release (e.g. R4.0.1)
2. Evolution: Following the initial release with zero, one or several minor releases and/or revisions (e.g. R4.0.2, R4.1.1)
3. Maintenance: No new contents is added to a major release but only maintenance of the existing content with zero, one or several revisions (e.g. R3.2.2) is provided
4. Issue Notice: No more revisions but zero, one or several issue notices, i.e. updates of the list of known issues until end of life cycle.

### 7.1.4 Specification item and requirement life cycle states

**The life cycle state of a specification item** is found after the specification item ID surrounded by curly brackets. The states are:

- **Valid**: This indicates that the related entity is a valid part of the document. This is the default.
- **Draft:** This indicates that the related entity is newly introduced but still experimental. This information is published but is subject to change without backward compatibility guarantee.
- **Obsolete:** This indicates that the related entity is obsolete and will be removed in the next release.

If there is no life cycle state information stated then the state is Valid.

**The life cycle state of a requirement** is found in the attribute „type“. The states are the same as the specification item states.

### 7.1.5 History information in AUTOSAR

The following diagram shows where which changes are documented.