

<b>Document Title</b>	General Requirements specific to Adaptive Platform
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	714

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Adaptive Platform
<b>Part of Standard Release</b>	18-03

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2018-03-29	18-03	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Text entry for Supporting Material for RS_AP_00111</li> <li>• Text entry for Supporting Material for RS_AP_00114 only refers now to ISO/IEC 14882</li> <li>• Description of RS_AP_00115 revised</li> <li>• Description of RS_AP_00116, RS_AP_00117, RS_AP_00118, RS_AP_00120, RS_AP_00121, RS_AP_00124, RS_AP_00125 revised (in general "all ara libraries" changed to "all functional clusters").</li> </ul>
2017-10-27	17-10	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Minor fixes</li> </ul>
2017-03-31	17-03	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	Scope of this document	4
1.1	Document Conventions . . . . .	5
2	Acronyms and Abbreviations	6
3	Requirements Tracing	7
4	Requirements specification	8
4.1	Architecture Requirements . . . . .	8
4.2	Non-functional Requirements . . . . .	8
	[RS_AP_00111] The AUTOSAR Adaptive platform shall support source code portability for AUTOSAR Adaptive applications. . . . .	8
4.3	Design Requirements . . . . .	8
	[RS_AP_00113] API specification shall comply with selected coding guidelines. . . . .	8
	[RS_AP_00114] C++ binding shall be compatible with C++11. . . . .	8
	[RS_AP_00115] Standardized scope/namespace definition . . . . .	9
	[RS_AP_00116] Header file name . . . . .	9
	[RS_AP_00117] Class and structure names . . . . .	9
	[RS_AP_00122] Type names . . . . .	10
	[RS_AP_00120] Method and Function names . . . . .	10
	[RS_AP_00121] Parameter names . . . . .	10
	[RS_AP_00124] Variable names . . . . .	10
	[RS_AP_00125] Enumerator and constant names . . . . .	11
	[RS_AP_00118] Exceptions . . . . .	11
	[RS_AP_00119] Return values / application errors . . . . .	11
5	References	12

## 1 Scope of this document

The goal of this document is to define a common set of basic requirements that apply to all SWS documents of the Adaptive Platform.

## 1.1 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS\_STDT\_00078], see Standardization Template, chapter Support for Traceability ([1]).

The verbal forms for the expression of obligation specified in [TPS\_STDT\_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([1]).

## 2 Acronyms and Abbreviations

There are no acronyms and abbreviations relevant within this document that are not included in the [2, AUTOSAR glossary].

### 3 Requirements Tracing

The following table references the requirements specified in [3] and links to the fulfillments of these.

Requirement	Description	Satisfied by
[RS_Main_00030]	AUTOSAR shall support development processes for safety related systems	<a href="#">[RS_AP_00113]</a>
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	<a href="#">[RS_AP_00111]</a> <a href="#">[RS_AP_00115]</a> <a href="#">[RS_AP_00116]</a> <a href="#">[RS_AP_00117]</a> <a href="#">[RS_AP_00118]</a> <a href="#">[RS_AP_00119]</a> <a href="#">[RS_AP_00120]</a> <a href="#">[RS_AP_00121]</a> <a href="#">[RS_AP_00122]</a> <a href="#">[RS_AP_00124]</a> <a href="#">[RS_AP_00125]</a>
[RS_Main_00500]	AUTOSAR shall provide naming conventions	<a href="#">[RS_AP_00113]</a> <a href="#">[RS_AP_00115]</a> <a href="#">[RS_AP_00116]</a> <a href="#">[RS_AP_00117]</a> <a href="#">[RS_AP_00120]</a> <a href="#">[RS_AP_00121]</a> <a href="#">[RS_AP_00122]</a> <a href="#">[RS_AP_00124]</a> <a href="#">[RS_AP_00125]</a>
[RS_Main_00513]	AUTOSAR shall support language bindings for different programming languages	<a href="#">[RS_AP_00114]</a>

## 4 Requirements specification

### 4.1 Architecture Requirements

### 4.2 Non-functional Requirements

**[RS\_AP\_00111] The AUTOSAR Adaptive platform shall support source code portability for AUTOSAR Adaptive applications. [**

<b>Type:</b>	draft
<b>Description:</b>	The AUTOSAR Adaptive platform shall support source code portability
<b>Rationale:</b>	Ensure reuse of existing IPs.
<b>Dependencies:</b>	–
<b>Use Case:</b>	Integration of applications on a platform, one of them delivered as source code.
<b>Supporting Material:</b>	Any implementation of the AUTOSAR Adaptive Platform shall allow successful compilation and linking of an Adaptive Application that uses ARA only as specified in the standard. No changes to the source code, and no conditional compilation constructs shall be necessary for this, if the application only uses constructs from the designated minimum C++ language version. The implementation may provide proprietary, non-ARA interfaces, as long as they are not contradicting with the AP standard. However, an implementation shall not add declarations or definitions that are not specified in an SWS to namespace ara or to a namespace within namespace ara unless otherwise specified.

]([RS\\_Main\\_00150](#))

### 4.3 Design Requirements

**[RS\_AP\_00113] API specification shall comply with selected coding guidelines. [**

<b>Type:</b>	draft
<b>Description:</b>	API specification shall comply with coding guideline "Guidelines for the use of the C++14 language in critical and safety-related systems" in order to adopt best practices for the safe and secure implementations.
<b>Rationale:</b>	Enhance readability and maintainability of specification.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	"Guidelines for the use of the C++14 language in critical and safety-related systems"

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00030](#))

**[RS\_AP\_00114] C++ binding shall be compatible with C++11. [**

<b>Type:</b>	draft
<b>Description:</b>	The AUTOSAR Adaptive platform shall be compatible with C++11.



<b>Rationale:</b>	The AUTOSAR Adaptive platform standard is designed to be compatible with C++11 due to high availability of C++11 compiler for embedded devices. Nevertheless projects are free to use newer C++ version like C++14. Adaptive platform vendors can restrict their package to a newer C++ version (e.g. to support newer build systems).
<b>Dependencies:</b>	RS_Main_00513
<b>Use Case:</b>	To master the complexity of the application development, the AUTOSAR Adaptive platform shall support object-oriented programming. C++ is the programming language, which supports object-oriented programming and is best suited for memory-constrained and real-time applications.
<b>Supporting Material:</b>	ISO/IEC 14882

]([RS\\_Main\\_00513](#))

### [RS\_AP\_00115] Standardized scope/namespace definition [

<b>Type:</b>	draft
<b>Description:</b>	The namespace of adaptive platform in global scope shall be "ara". Within "ara" namespace each functional cluster shall have exactly one own namespace with its shortname (defined in [4]). No other namespaces below "ara" are permitted. All names shall be all lower-case. Underscores can be used.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00150](#))

### [RS\_AP\_00116] Header file name [

<b>Type:</b>	draft
<b>Description:</b>	All functional clusters shall provide a self-contained header file for each public class (except scoped enum and exceptions). The header file name shall be derived from the class name as recommended by Google C++ Style Guide. All header file names shall have the extensions .h.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	Google C++ Style Guide [ <a href="https://google.github.io/styleguide/cppguide.html">https://google.github.io/styleguide/cppguide.html</a> ]

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00150](#))

### [RS\_AP\_00117] Class and structure names [

<b>Type:</b>	draft
<b>Description:</b>	For all functional clusters the name of their public classes and structures shall be standardized in PascalCase. Further underscores shall not be used. Capitalized acronyms shall be used as single words.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–

<b>Supporting Material:</b>	<a href="https://en.wikipedia.org/wiki/PascalCase">https://en.wikipedia.org/wiki/PascalCase</a>
-----------------------------	---

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00150](#))

#### [RS\_AP\_00122] Type names [

<b>Type:</b>	draft
<b>Description:</b>	For all functional clusters the name of their types shall be standardized in PascalCase. Further underscores shall not be used. Capitalized acronyms shall be used as single words.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	<a href="https://en.wikipedia.org/wiki/PascalCase">https://en.wikipedia.org/wiki/PascalCase</a>

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00150](#))

#### [RS\_AP\_00120] Method and Function names [

<b>Type:</b>	draft
<b>Description:</b>	For all functional clusters the name of their public methods and functions shall be standardized in PascalCase. Further underscores shall not be used. Capitalized acronyms shall be used as single words.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	<a href="https://en.wikipedia.org/wiki/PascalCase">https://en.wikipedia.org/wiki/PascalCase</a>

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00150](#))

#### [RS\_AP\_00121] Parameter names [

<b>Type:</b>	draft
<b>Description:</b>	For all functional clusters the name of parameters in methods shall be standardized in lower camel case. Further underscores shall not be used. Capitalized acronyms shall be used as single words.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	<a href="https://en.wikipedia.org/wiki/CamelCase">https://en.wikipedia.org/wiki/CamelCase</a>

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00150](#))

#### [RS\_AP\_00124] Variable names [

<b>Type:</b>	draft
<b>Description:</b>	For all functional clusters the name of their variables (like Common Variable names, Class Data Members and Struct Data Members) shall be standardized in lower camel case. Further underscores shall not be used. Capitalized acronyms shall be used as single words.

<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	<a href="https://en.wikipedia.org/wiki/CamelCase">https://en.wikipedia.org/wiki/CamelCase</a>

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00150](#))

#### [RS\_AP\_00125] Enumerator and constant names [

<b>Type:</b>	draft
<b>Description:</b>	For all functional clusters the name of enumerations shall be standardized in PascalCase. The individual enumerators and constants shall be written with a leading "k" followed by PascalCase. Further underscores shall not be used. Capitalized acronyms shall be used as single words.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	<a href="https://en.wikipedia.org/wiki/PascalCase">https://en.wikipedia.org/wiki/PascalCase</a>

]([RS\\_Main\\_00500](#), [RS\\_Main\\_00150](#))

#### [RS\_AP\_00118] Exceptions [

<b>Type:</b>	draft
<b>Description:</b>	For all functional clusters the 'checked exceptions' shall be standardized and documented: when they are thrown and what is the rationale. The type names for exceptions, that are fully specified in AUTOSAR SWS specifications (i.e. not derived from the configuration), shall include the suffix "Exception". Adaptive Platform Vendors can add 'unchecked exceptions' on their own if necessary.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	For adaptive services it is done via 'extended application error'.
<b>Use Case:</b>	–
<b>Supporting Material:</b>	

]([RS\\_Main\\_00150](#))

#### [RS\_AP\_00119] Return values / application errors [

<b>Type:</b>	draft
<b>Description:</b>	For return values and application errors the condition(s) when they are returned shall be specified. Furthermore for each return value the list of valid output parameter(s) shall be specified.
<b>Rationale:</b>	Harmonized look and feel.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00150](#))

## 5 References

- [1] Standardization Template  
AUTOSAR\_TPS\_StandardizationTemplate
- [2] Glossary  
AUTOSAR\_TR\_Glossary
- [3] Main Requirements  
AUTOSAR\_RS\_Main
- [4] Functional Cluster Shortnames  
AUTOSAR\_TR\_FunctionalClusterShortnames