

<b>Document Title</b>	Requirements of State Management
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	909

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Adaptive Platform
<b>Part of Standard Release</b>	18-10

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2018-10-31	18-10	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 Document	4
1.1	Document Conventions . . . . .	4
2	Acronyms and Abbreviations	4
3	Requirements Specification	5
3.1	Functional Overview . . . . .	5
3.2	Functional Requirements . . . . .	5
3.2.1	State Management . . . . .	5
3.2.2	Support for Diagnostics . . . . .	6
3.2.3	Virtualization support . . . . .	7
3.2.4	Calibration and variant support . . . . .	8
3.2.5	Dynamic communication paths . . . . .	8
3.2.6	Efficient resource usage . . . . .	9
4	Requirements Tracing	9
5	References	10

# 1 Scope of Document

This document specifies requirements on [State Management](#). [State Management](#) implements interfaces of `State Manager` on the AUTOSAR Adaptive Platform, because [State Management](#) is highly project specific and therefore to be implemented by the project itself.

## 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

The glossary below includes acronyms and abbreviations relevant to the `StateManagement` module that are not included in the AUTOSAR glossary[2].

Terms:	Description:
Process	A process is a loaded instance of an <code>Executable</code> to be executed on a <code>Machine</code> .
Execution Management	The element of the <code>Adaptive Platform</code> responsible for the ordered startup and shutdown of the <code>Adaptive Platform</code> and the <code>Applications</code> .
State Management	The element defining modes of operation for <code>Adaptive Platform</code> . It allows flexible definition of functions which are active on the platform at any given time.
Function Group State	The element of <a href="#">State Management</a> that characterizes the current status of a set of (functionally coherent) user-level <code>Applications</code> . The set of <code>Function Groups</code> and their <code>Function Group States</code> is machine specific and are configured in the <code>Machine Manifest</code> [3].
Machine State	The element of the <a href="#">State Management</a> . See <a href="#">Function Group State</a> .
Operational State	The element of <a href="#">State Management</a> that characterizes the current internal state of the <code>State Manager</code> . The <code>Operational State</code> is machine specific and are depends on multiple events from somewhere within the system.
Component	Element of a <code>Process</code> . <code>Processes</code> are comprised of one or more SW-entities that provide a particular function or group of related functions, called <code>Component</code> . <b>Please note that the term 'Component' is not yet fixed for this scope.</b>

Component State	The element of <i>State Management</i> that characterizes the current state of <i>Components</i> within an Adaptive Application. The Component State is Adaptive Application specific and therefore it has to be described in the respective Execution Manifest. Every Adaptive Application provides at least an "On" and an "Off" State.
-----------------	---

### 3 Requirements Specification

This chapter describes all requirements driving the work to define the *StateManagement*.

#### 3.1 Functional Overview

This document specifies the requirements regarding the realization of the *StateManagement* on Adaptive Platform. Only the interfaces and abstract functionality will be defined, because *StateManagement* is highly project specific.

#### 3.2 Functional Requirements

##### 3.2.1 State Management

**[RS\_SM\_00001] *State Management* shall support Function Group state change requests.** [

<b>Type:</b>	draft
<b>Description:</b>	<i>State Management</i> shall allow an Application to request a change in <i>Machine State</i> or <i>Function Group State</i> .
<b>Rationale:</b>	To support the starting and stopping of Applications based on declared state dependencies, <i>State Management</i> shall include an interface to request <i>Machine State</i> and/or <i>Function Group State</i> changes by external Applications. In response to state change requests, <i>State Management</i> triggers requests to the <i>Execution Management</i> [4] that creates or terminates <i>Processes</i> based on the configured mode dependencies of the respective applications.
<b>Dependencies:</b>	-
<b>Use Case:</b>	Provide a mechanism to define modes of operation of the Machine.
<b>Supporting Material:</b>	-

] ([RS\\_Main\\_00460](#))

[RS\_SM\_00002] **State Management shall support Component State change requests.** [

<b>Type:</b>	draft
<b>Description:</b>	<b>State Management</b> shall support to change the state of individual software <b>Components</b>
<b>Rationale:</b>	<b>Processes</b> are comprised of one or more software <b>Components</b> . <b>Component states</b> allow <b>State Management</b> to interact with those <b>Components</b> and change their behavior based on a external and internal events. <b>State Management</b> is also able to orchestrate between the states of different software components. <b>Component State</b> changes do not require to restart the hosting <b>Process</b> .
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00460)

### 3.2.2 Support for Diagnostics

[RS\_SM\_00100] **State Management shall support ECU reset** [

<b>Type:</b>	draft
<b>Description:</b>	<b>State Management</b> shall support to reset the ECU.
<b>Rationale:</b>	<b>Diagnostic Application</b> [5] shall support ECUReset according to ISO 14229-1 [6]. <b>State Management</b> shall handle and coordinate the requests from <b>Diagnostic Application</b> .
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00260)

[RS\_SM\_00101] **State Management shall support diagnostic reset cause** [

<b>Type:</b>	draft
<b>Description:</b>	<b>State Management</b> shall support the provision of the last reset cause to <b>Functional Clusters</b> , e.g. <b>Diagnostics</b> .





<b>Rationale:</b>	(Diagnostic) Applications need to determine the cause of the last reset, e.g. to support UDS service ECUReset. The information on what triggered a reset is required so that as after a reset Diagnostics can determine whether it was a planned (controlled, requested) reset or unplanned and have to react accordingly.
<b>Dependencies:</b>	–
<b>Use Case:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00260)

### 3.2.3 Virtualization support

[RS\_SM\_00200] State Management shall provide an interface between State Management instances. [

<b>Type:</b>	draft
<b>Description:</b>	State Management shall provide an interface between State Management instances used in a hierarchically manner.
<b>Rationale:</b>	In a virtualized environment several instances of State Management will be active. Instances with lower priority have to be controlled by instances with a higher priority
<b>Dependencies:</b>	
<b>Use Case:</b>	The components are possibly provided by different vendors, working on different microcontrollers or virtual machines. On each controller or (virtual) machine a separate instance of State Management might be used and it should be possible to operate these instances in a hierarchically manner.
<b>Supporting Material:</b>	

](RS\_Main\_00511)

[RS\_SM\_00201] State Management shall provide the interface over ara::com. [

<b>Type:</b>	draft
<b>Description:</b>	State Management shall provide the hierarchically interface over at least ara::com.
<b>Rationale:</b>	
<b>Dependencies:</b>	





<b>Use Case:</b>	The <a href="#">State Management</a> is possibly provided by different vendors, working on different microcontrollers or virtual machines. On each controller or (virtual) machine a separate instance of <a href="#">State Management</a> might be used and it should be possible to operate these instances in a hierarchically manner. Note: Providing the <code>ara::com</code> is mandatory for each implementation <a href="#">State Management</a> , but it is also possible to add more efficient implementations locally.
<b>Supporting Material:</b>	

]([RS\\_Main\\_00511](#))

### 3.2.4 Calibration and variant support

[[RS\\_SM\\_00300](#)] [State Management](#) shall support variant handling based on calibration data. [

<b>Type:</b>	draft
<b>Description:</b>	<a href="#">State Management</a> shall evaluate calibration data. <a href="#">State Management</a> should (or not) set <a href="#">Function Groups</a> to specified <a href="#">Function Group State</a> depending on read configuration data.
<b>Rationale:</b>	
<b>Dependencies:</b>	
<b>Use Case:</b>	For different car lines, countries or regions different <a href="#">Function Groups</a> will be allowed to be started. <a href="#">State Management</a> evaluates this information from calibration data to enable only the wanted <a href="#">Function Groups</a> .
<b>Supporting Material:</b>	

]([RS\\_Main\\_00261](#), [RS\\_Main\\_00360](#))

### 3.2.5 Dynamic communication paths

[[RS\\_SM\\_00400](#)] [State Management](#) shall establish communication paths dynamically. [

<b>Type:</b>	draft
<b>Description:</b>	<a href="#">State Management</a> shall be able to evaluate which communication channels are needed by a <a href="#">Function Group</a> . Opening and closing of these channels shall be done by requesting them from <a href="#">Network Management</a> .







<b>Rationale:</b>	
<b>Dependencies:</b>	
<b>Use Case:</b>	Function Group will have a need to use communication with other ones. Therefore <i>State Management</i> evaluates this information from configuration and requests <i>NetworkManagement</i> to establish or shutdown the corresponding communication channel.
<b>Supporting Material:</b>	

]([RS\\_Main\\_01005](#))

### 3.2.6 Efficient resource usage

[[RS\\_SM\\_00500](#)] *State Management* shall support efficient resource usage. [

<b>Type:</b>	draft
<b>Description:</b>	<i>State Management</i> shall be able to use the system resources efficiently during system startup.
<b>Rationale:</b>	<i>State Management</i> shall implement means to parallelize and order startup of <i>Function Groups</i> in such way as to minimize system startup time.
<b>Dependencies:</b>	
<b>Use Case:</b>	
<b>Supporting Material:</b>	

]([RS\\_Main\\_00200](#))

## 4 Requirements Tracing

The following table references the features specified in [7] and links to the fulfillments of these.

Feature	Description	Satisfied by
[ <a href="#">RS_Main_00200</a> ]	AUTOSAR specifications shall allow resource efficient implementations	[ <a href="#">RS_SM_00500</a> ]
[ <a href="#">RS_Main_00260</a> ]	AUTOSAR shall provide diagnostics means during runtime, for production and services purposes	[ <a href="#">RS_SM_00100</a> ] [ <a href="#">RS_SM_00101</a> ]
[ <a href="#">RS_Main_00261</a> ]	AUTOSAR shall provide means for calibration	[ <a href="#">RS_SM_00300</a> ]
[ <a href="#">RS_Main_00360</a> ]	AUTOSAR shall support variant management	[ <a href="#">RS_SM_00300</a> ]
[ <a href="#">RS_Main_00460</a> ]	AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level	[ <a href="#">RS_SM_00001</a> ] [ <a href="#">RS_SM_00002</a> ]
[ <a href="#">RS_Main_00511</a> ]	AUTOSAR shall support virtualization	[ <a href="#">RS_SM_00200</a> ] [ <a href="#">RS_SM_00201</a> ]

[RS_Main_01005]	AUTOSAR shall establish communication paths dynamically	[RS_SM_00400]
-----------------	---	---------------

## 5 References

- [1] Standardization Template  
AUTOSAR\_TPS\_StandardizationTemplate
- [2] Glossary  
AUTOSAR\_TR\_Glossary
- [3] Specification of Manifest  
AUTOSAR\_TPS\_ManifestSpecification
- [4] Specification of Execution Management  
AUTOSAR\_SWS\_ExecutionManagement
- [5] Specification of Diagnostics  
AUTOSAR\_SWS\_Diagnostics
- [6] Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2013-03)  
<http://www.iso.org>
- [7] Main Requirements  
AUTOSAR\_RS\_Main