Document Title	Requirements on Persistency
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	857

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R19-11

Document Change History

Date	Release	Changed by	Description
2019-11-28	R19-11	AUTOSAR Release Management	• Statistic data • Initialization/De-initialization • Changed Document Status from Final to published
2019-03-29	19-03	AUTOSAR Release Management	• Updated introduction • Information on wear leveling • Finalization was removed
2018-10-31	18-10	AUTOSAR Release Management	• Restructured document • Added resource handling • Added UCM related requirements
2018-03-29	18-03	AUTOSAR Release Management	• Added requirement on configuration • Removed requirement on authorization
2017-10-27	17-10	AUTOSAR Release Management	• Requirements on Secure Storage • Requirements on Safe Storage
2017-03-31	17-03	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.

Table of Contents

1 Scope of this document

This document specifies the requirements of Adaptive Applications to the functional cluster Persistency of the AUTOSAR Adaptive Platform. The motivation is to provide a standardized and portable way to store and write data persistently.

2 Conventions to be used

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template [1], chapter Support for Traceability.

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template [1], chapter Support for Traceability.

3 Acronyms and Abbreviations

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

4 Requirements Specification

4.1 Functional overview

The AUTOSAR Adaptive Platform Persistency provides services for Adaptive Applications and other clusters of the AUTOSAR Adaptive Platform. The AUTOSAR Adaptive Platform Persistency is responsible for all aspects which regard the storage/retrieval of data, and therefore it has to deal with:

- Persistently storing data over boot and ignition cycles.
- Accessing data which has been stored persistently.
- Using a unique identifier to access data.
- Reading and writing data from/to files.
- Encryption of persistent data.
- Error detection and correction of stored data.
- Monitoring storage space.

4.2 Configuration of Persistent Data

[RS_PER_00010]{DRAFT} **The layout of persistent data shall be configurable** ⌈

Type:	draft
Description:	Persistency shall support configuration of provided key-value data bases and file arrays.
Rationale:	Generation of interfaces
Dependencies:	–
Use Case:	An Adaptive Application or a Functional Cluster needs access to persistent data and expects a dedicated interface for each set of data.
Supporting Material:	–

⌋*(RS_Main_00440)*

4.3 Storage of Persistent Data

[RS_PER_00001]{DRAFT} **Persistency shall support storage of persistent data** ⌈

Type:	draft
Description:	Persistency shall support persistent storage of data on a platform instance over restart cycles of the Adaptive Application or the platform.
Rationale:	Persistent storage of data
Dependencies:	–
Use Case:	Storage of data which shall be available after an Adaptive Application has been shut down and restarted. In case of direct storage to flash memory or other storage hardware that has a limited number of write cycles, the implementation of Persistency needs to take care of wear leveling.
Supporting Material:	–

⌋*(RS_Main_00440)*

[RS_PER_00018]{DRAFT} **Persistency shall support central initialization and shutdown** ⌈

Type:	draft
Description:	By using a central mechanism, an application developer shall be able to control the life cycle of Persistency together with other functional clusters with direct interfaces to the application.
Rationale:	Preparation and cleanup of the in-memory structures of Persistency is required to ensure a reliable life cycle handling.

▽

△

Dependencies:	–
Use Case:	The manifest configuration of Persistency needs to be parsed before any storage can be accessed, and before shutdown, all data of Persistency should be persisted and the structures cleaned up.
Supporting Material:	–

⌋*(RS_Main_00011)*

[RS_PER_00002]{DRAFT} **Persistency shall support to retrieve data that has been persistently stored on a platform instance** ⌈

Type:	draft
Description:	Persistency shall provide the functionality to load data which is persistently stored.
Rationale:	Load of persistently stored data
Dependencies:	–
Use Case:	An Adaptive Application or Functional Cluster which stores persistent data needs to restore it after a restart of the Adaptive Application or the platform.
Supporting Material:	–

⌋*(RS_Main_00440)*

[RS_PER_00003]{DRAFT} **Persistency shall support identification of data using a unique identifier** ⌈

Type:	draft
Description:	Data shall be stored in way that it can be accessed from an Adaptive Application or a Functional Cluster by using a unique identifier e.g. identify a value by a key.
Rationale:	Load of persistently stored data
Dependencies:	–
Use Case:	Storage of a variety of different data objects that can be accessed individually for loading.
Supporting Material:	–

⌋*(RS_Main_00440)*

[RS_PER_00004]{DRAFT} **Persistency shall support access to file-like structures** ⌈

Type:	draft
Description:	Persistency shall provide a standardized way to access file-like structures. Adaptive Applications and Functional Clusters shall be able to read and write data from file-like structures.
Rationale:	Filesystem abstraction
Dependencies:	–
Use Case:	Persistent data can be represented in multiple ways, e.g. human-readable format or binary. Every format of data needs to be accessible by the Persistency cluster.
Supporting Material:	–

⌋*(RS_Main_00440)*

4.4 Secure Storage of Persistent Data

[RS_PER_00005]{DRAFT} **Persistency shall support encryption/decryption of persistent data** ⌈

Type:	draft
Description:	Persistency shall provide a standardized way to encrypt/decrypt persistent data.
Rationale:	Support of data encryption
Dependencies:	–
Use Case:	Storage of persistent data that shall be encrypted for security reasons.
Supporting Material:	–

⌋*(RS_Main_00514)*

4.5 Safe Storage of Persistent Data

[RS_PER_00008]{DRAFT} **Persistency shall support detection of data corruption in persistent memory** ⌈

Type:	draft
Description:	Persistency shall support detection of data corruption in persistently stored data. The corruption may be caused by systematic or random failures.
Rationale:	Support detection of data corruption
Dependencies:	–

▽

△

Use Case:	Notification to an Adaptive Application or Functional Cluster in case of corrupted data in persistent memory. The detection of data corruption is also necessary to support data recovery mechanisms.
Supporting Material:	–

⌋*(RS_Main_00011)*

[RS_PER_00009]{DRAFT} **Persistency shall support data recovery mechanisms if persistent data was corrupted** ⌈

Type:	draft
Description:	Persistency shall support a recovery mechanism if corruption of persistently stored data was detected.
Rationale:	Support of data recovery mechanism
Dependencies:	–
Use Case:	If corruption of persistent data was detected it shall be possible for safety reasons to recover corrupted data (e.g. by usage of redundant data or error correcting codes).
Supporting Material:	–

⌋*(RS_Main_00011)*

4.6 Installation, Update, Roll-back, and Removal of Persistent Data

[RS_PER_00012]{DRAFT} **Persistency shall support installation of persistent data** ⌈

Type:	draft
Description:	Persistency shall allow for installation of pre-configured values in key-value databases and pre-configured files in a file-proxy.
Rationale:	It shall be possible to install an application with a preset.
Dependencies:	–
Use Case:	Providing initial or fixed content for key-value databases and file-proxies.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_PER_00013]{DRAFT} **Persistency shall support update of persistent data** ⌈

Type:	draft
Description:	Persistency shall allow for an update of values in key-value databases and of files in a file-proxy.
Rationale:	It shall be possible to update an application and set a new preset.
Dependencies:	–
Use Case:	Providing updated content for key-value databases and file-proxies.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_PER_00014]{DRAFT} **Persistency shall support roll-back of persistent data** ⌈

Type:	draft
Description:	Persistency shall allow for a roll-back of values in key-value databases and files in a file-proxy to the state before an update.
Rationale:	It shall be possible to roll back an application and return persisted data to its previous state.
Dependencies:	–
Use Case:	Reverting the content of key-value databases and file-proxies.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_PER_00015]{DRAFT} **Persistency shall support removal of persistent data** ⌈

Type:	draft
Description:	Persistency shall allow for removal of all key-value databases and file-proxies.
Rationale:	It shall be possible to uninstall an application together with its persisted data.
Dependencies:	–
Use Case:	Removing all persisted content of key-value databases and file-proxies.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

4.7 Resource handling for Persistent Data

[RS_PER_00011]{DRAFT} **Persistency shall be able to ensure and limit the amount of storage used by persisted data** ⌈

Type:	draft
Description:	Persistency shall support monitoring of the storage space allocated by persistently stored data. It shall ensure that a configurable amount of storage space is always available for stored data, and that the stored data never surpasses a configurable limit.
Rationale:	Avoid situations where applications cannot run reliably because they cannot access the required amount of storage, or because another application uses too much storage.
Dependencies:	–
Use Case:	Ensuring reliability of the access to the persistently stored data of a single process, and ensuring overall reliability of applications regarding access to persistently stored data.
Supporting Material:	–

⌋*(RS_Main_00011)*

[RS_PER_00017]{DRAFT} **Persistency shall be able to report the amount of currently used storage** ⌈

Type:	draft
Description:	Persistency shall support querying the amount of storage currently allocated by persisted data.
Rationale:	It shall be possible to acquire information about persistent storage.
Dependencies:	–
Use Case:	Polling of the current size of persisted data using a diagnostic service.
Supporting Material:	–

⌋*(RS_Main_00440)*

5 Requirements Tracing

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

Requirement	Description	Satisfied by
[RS_Main_00011]	AUTOSAR shall support the development of reliable systems	[RS_PER_00008] [RS_PER_00009] [RS_PER_00011] [RS_PER_00018]
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	[RS_PER_00012] [RS_PER_00013] [RS_PER_00014] [RS_PER_00015]

[RS_Main_00440]	AUTOSAR shall standardize access to non-volatile memory	[RS_PER_00001] [RS_PER_00002] [RS_PER_00003] [RS_PER_00004] [RS_PER_00010] [RS_PER_00017]
[RS_Main_00503]	AUTOSAR shall support change of communication and application software at runtime.	[RS_PER_00012] [RS_PER_00013] [RS_PER_00014] [RS_PER_00015]
[RS_Main_00514]	AUTOSAR shall support the development of secure systems	[RS_PER_00005]

6 References

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_TR_Glossary

[3] Main Requirements
AUTOSAR_RS_Main