Document Title	Requirements on Update and Configuration Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	887

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	18-03

Document Change History

Date	Release	Changed by	Description
2017-03-29	18-03	AUTOSAR Release Management	• Requirements on Software Updates • Requirements on Data Transfer • Requirements on Version Reporting • Requirements on Validation
2017-10-27	17-10	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 Document

This document specifies requirements of the AUTOSAR Adaptive Platform on the Update and Configuration Management (UCM). The motivation of UCM is to provide a standardized way to install, update and uninstall software on the adaptive platform safely and securely.

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

All acronyms and abbreviations relevant for this document are included in the AUTOSAR Glossary [2].

4 Constraints and assumptions

This chapter lists known limitations of UCM in terms of unimplemented requirements. The intent is to provide an indication how the Adaptive Platform will evolve in future releases.

Software Cluster Manifest referred in [RS_UCM_00016] is not specified in 18-03 release.

The following requirements are described within this document but not otherwise considered in this release:

- [RS_UCM_00027]

The functionality described above is subject to modification and will be considered for inclusion in a future release of this document.

5 Functional Overview

One of the declared goals of Adaptive AUTOSAR is the ability to flexibly update the software and its configuration through local (“tester-based”) or remote (“over-the-air”) updates. UCM provides services for updating the software and its configuration running on Adaptive Platform. UCM is responsible for updates of Adaptive Platform Applications and changes to the Adaptive Platform itself, including all functional clusters and the underlying OS. Therefore this document includes requirements on the following functionality:

- Installation of new software:
  - Install software
  - Install persistent data for the software
- Update of already installed software:
  - Update software
  - Update persistent data for the software
- Uninstallation of installed software:
  - Remove software
  - Remove data created by the software
- Providing information of installed software:
  - Names of installed software
  - Versions of installed software
- Security and safety of the update process
  - Signature verification
  - Software dependency check
  - Recovery of software after failure during update process

6 Requirements Specification

6.1 Versions reporting

[RS_UCM_00002] UCM shall support reporting version information for an Adaptive Platform ⌈

Type:	draft
Description:	The UCM shall provide functionality to retrieve version information describing the software installed on Adaptive Platform.
Rationale:	Adaptive Platform shall support keeping software up-to-date through vehicles lifecycle.
Dependencies:	–
Use Case:	Retrieve version information of the installed software to determine which software needs to be installed, updated or removed.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00010] UCM shall support reporting of Software Packages downloaded for Adaptive Platform ⌈

Type:	draft
Description:	The UCM shall provide functionality to retrieve information describing the software downloaded, but not activated on Adaptive Platform
Rationale:	Adaptive Platform shall support downloading data and continuing it after re-establishing a lost connection.
Dependencies:	–
Use Case:	Retrieve list of downloaded packages to know which data still needs to be downloaded and which has been already downloaded and can now be installed.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00011] UCM shall support reporting software versions which have been installed and will be activated when new versions are activated ⌈

Type:	draft
Description:	UCM shall support reporting software versions which have been installed and will be activated when new versions are activated
Rationale:	Adaptive Platform shall support retrieving information which software is already updated and will be activated in the switch to new versions.
Dependencies:	–
Use Case:	Retrieve list of installed software in order to check that all intended software have been installed and is ready for activation.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

6.2 Data management

[RS_UCM_00013] UCM shall check that it has enough resources to receive, process and store the Software Package and associated data ⌈

Type:	draft
Description:	UCM shall assure that it has enough resources to receive, process and store the Software Package and associated data. Resources may be for example memory, CPU time or network sockets. The resource check only covers the installation process. It is the responsibility of the integrator creating the Software Package to ensure enough resources for actual execution are available on the Adaptive Platform (e.g. RAM, CPU time).
Rationale:	Adaptive Platform shall protect itself from resource starvation.
Dependencies:	–
Use Case:	Deny further transfers if no more resources for additional clients are available.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00014] UCM shall check that correct amount of data has been transferred for the Software Package ⌈

Type:	draft
Description:	UCM shall verify if the size of the received data is the same as the allocated amount stated in the download request in order to ensure the download was successful
Rationale:	UCM shall make sure complete Software Package was transferred.
Dependencies:	[RS_UCM_00013]
Use Case:	UCM shall assure that same amount of data has been transferred as was reserved in the downloaded request.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00015] UCM shall remove all unneeded data after Software Package processing has finished ⌈

Type:	draft
Description:	UCM shall remove all unneeded data such as logs, possible backups after Software Package processing has finished
Rationale:	UCM shall make sure all unneeded resources are freed after Software Package processing has finished.
Dependencies:	–
Use Case:	At the end of update process UCM shall remove log files, stored backups for Software Package and other temporary items created during update process.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

6.2.1 Data transfer

[RS_UCM_00025] UCM shall support efficient streaming of Software Package data ⌈

Type:	draft
Description:	UCM shall support efficient streaming of Software Package data
Rationale:	Adaptive Platform shall support updates originating from another Adaptive Platform. Therefore no local, e.g. file-based transfer shall be used.
Dependencies:	–
Use Case:	Receive Software Package(s) from a DSA.
Supporting Material:	–

⌋*()*

[RS_UCM_00019] UCM shall support simultaneous transfer from multiple clients ⌈

Type:	draft
Description:	UCM shall support simultaneous transfer from multiple clients
Rationale:	Adaptive Platform shall support multiple update sources. Therefore several clients shall be able to transfer Software Packages to UCM simultaneously.
Dependencies:	[RS_UCM_00025]
Use Case:	Receive Software Package(s) from a DSA.
Supporting Material:	–

⌋*()*

6.3 Software updates

[RS_UCM_00001] UCM shall support installing new software on Adaptive Platform ⌈

Type:	draft
Description:	The UCM shall provide functionality to install new software on the Adaptive Platform.
Rationale:	Adaptive Platform shall support keeping software up-to-date and to introduce new features through out vehicles life cycle.
Dependencies:	–
Use Case:	Introduce new functionality with new Adaptive Application or replace old functionality by replacing old Adaptive Application with new Adaptive Application.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00003] UCM shall support updating installed software on Adaptive Platform ⌈

Type:	draft
Description:	UCM shall provide functionality to update already installed software.
Rationale:	Adaptive Platform shall support keeping software up-to-date through vehicles life cycle.
Dependencies:	–
Use Case:	Update an application to achieve updated or fixed functionality.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00020] UCM shall support cancel of an update or install operation ⌈

Type:	draft
Description:	UCM shall support cancel of an update or install operation
Rationale:	Adaptive Platform shall support cancel of an update on user request.
Dependencies:	–
Use Case:	Update has timed out.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00026] UCM shall process installation of new Software Packages, updates and removal of Software Packages sequentially ⌈

Type:	draft
Description:	UCM shall process installation of new software, updates and removal of software sequentially, i.e. not in parallel.
Rationale:	Adaptive Platform shall support safe installation of Software Packages, therefore it shall not be possible to apply several modifications to installed software simultaneously.
Dependencies:	–
Use Case:	Software Packages are transferred by multiple clients. One client starts installing, the other has to wait until installation is finished to install packages transferred by it.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00017] UCM shall support installing and updating the persistent data storage for an Adaptive Application ⌈

Type:	draft
Description:	The UCM shall process the persistent data in the Software Package so that it is available through Persistency
Rationale:	To support changing the Adaptive Application configurations.
Dependencies:	–
Use Case:	Install an Software Package to change the configuration for an Adaptive Platform Application
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00004] UCM shall support uninstalling software on Adaptive Platform ⌈

Type:	draft
Description:	The UCM shall provide functionality to uninstall already installed software.
Rationale:	Adaptive Platform shall support removing unwanted or malfunctioning software.
Dependencies:	–
Use Case:	Uninstall an application which will be replaced with other installed application or which is not needed anymore.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00005] UCM shall make sure that persistent data owned by uninstalled software is deleted ⌈

Type:	draft
Description:	The UCM shall make sure that also persistent data is deleted with uninstalled software.
Rationale:	Make sure the uninstall process does not leave unused data.
Dependencies:	–
Use Case:	Make sure install and uninstall cycle do not leave unused data and does not cause memory issues.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00021] UCM shall support atomic activation of installed or updated packages ⌈

Type:	draft
Description:	UCM shall support atomic activation of installed or updated packages
Rationale:	UCM shall apply the modifications to NVM in an inactive memory area and atomically switch when the modifications are finished.
Dependencies:	–
Use Case:	After a failed remote update the Adaptive Platform recovers to the previous system state.
Supporting Material:	–

⌋*()*

[RS_UCM_00008] UCM shall support a recovery mechanism in case of failed update process ⌈

Type:	draft
Description:	UCM shall assure that, in case of failed update process, the system will recover to the state it was before the update process started.
Rationale:	A failed update shall not result in a loss of desired functionality of the Adaptive Platform.

Dependencies:	[RS_UCM_00021]
Use Case:	After a failed remote update the Adaptive Platform recovers to the previous system state.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503, RS_Main_00011)*

[RS_UCM_00018] UCM shall announce when an application has been installed, updated or uninstalled ⌈

Type:	draft
Description:	UCM shall make information available for other Functional Clusters that software configuration has changed
Rationale:	Several Functional Clusters need to be constantly aware of current Software configuration
Use Case:	UCM shall provide information of installed or updated application for Functional Clusters which subscribed to this information. E.g. Execution Manager to manage start of the application or for Health Manager to monitor the functionality of the application.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

[RS_UCM_00027] UCM shall be able to safely recover from unexpected interruption. ⌈

Type:	draft
Description:	At startup UCM shall be able to identify if some action was interrupted and exited in an uncontrolled way and needs to be reverted or finished to return the software into safe state
Rationale:	UCM shall make sure that software should not be started up into inconsistent and unsafe state
Use Case:	After unexpected reset or crash UCM shall identify that there was an interruption while an action was on going and UCM shall handle this by reverting or by finishing the unfinished action.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503)*

6.4 Logging, progress and status

[RS_UCM_00022] UCM shall support logging of the update or installation process ⌈

Type:	draft
Description:	UCM shall support logging of the update or installation process

Rationale:	UCM shall support the logging of update or installation process to enable debugging of a failed update or installation
Dependencies:	–
Use Case:	After a failed remote or local update the user accesses log information.
Supporting Material:	–

⌋*()*

[RS_UCM_00023] UCM shall provide an interface to read progress of the update ⌈

Type:	draft
Description:	UCM shall provide an interface to read progress of the update
Rationale:	Make the information available to user
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋*()*

[RS_UCM_00024] UCM shall provide an interface to read internal status of UCM ⌈

Type:	draft
Description:	UCM shall provide an interface to read internal status of UCM
Rationale:	UCM shall provide its internal state to clients which can then react on changing states.
Dependencies:	–
Use Case:	One client is installing packages, another client has to wait until the state allows the next operation.
Supporting Material:	–

⌋*()*

6.5 Validation

[RS_UCM_00006] UCM shall check Software Package authentication during processing using signature verification ⌈

Type:	draft
Description:	UCM shall check whether a package to be installed onto Adaptive Platform is authenticated by means of signature verification.
Rationale:	Adaptive platform shall make sure that origin of the Software Package can be trusted.
Dependencies:	–
Use Case:	An untrusted origin can be an attempt to hack the Adaptive Platform.
Supporting Material:	–

⌋(*RS_Main_00150*, *RS_Main_00503*)

[RS_UCM_00012] UCM shall check the consistency of Software Package during processing ⌈

Type:	draft
Description:	UCM shall check the consistency of the Software Package to be installed onto Adaptive Platform.
Rationale:	Adaptive platform shall make sure that the Software Package can be installed safely and securely.
Dependencies:	–
Use Case:	UCM shall check that provided Software Package data and content matches the description given by the Software Cluster Manifest file and that the Software Package has not been altered during the transfer process.
Supporting Material:	–

⌋(*RS_Main_00150*, *RS_Main_00503*)

[RS_UCM_00007] UCM shall check that software dependencies are fulfilled ⌈

Type:	draft
Description:	UCM shall check dependencies configured for a Software Package.
Rationale:	Adaptive platform shall make sure that the dependencies of the considered package are fulfilled.
Dependencies:	–
Use Case:	Check that installed software is able to run with all needed dependencies available on Adaptive Platform.
Supporting Material:	–

⌋(*RS_Main_00150*, *RS_Main_00503*)

[RS_UCM_00016] UCM shall check installed software is consistent with provided Software Cluster Manifest ⌈

Type:	draft
Description:	UCM shall check that all the requested Software Packages have been installed as it has been described in Software Cluster Manifest
Rationale:	UCM shall check that Software Packages content matches with provided Software Cluster Manifest
Dependencies:	–
Use Case:	Consistency check for activation for update which covers several Software Packages.
Supporting Material:	–

⌋(*RS_Main_00150*, *RS_Main_00503*)

7 Requirements Tracing

The following table references the features 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_UCM_00008]
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	[RS_UCM_00001] [RS_UCM_00002] [RS_UCM_00003] [RS_UCM_00004] [RS_UCM_00005] [RS_UCM_00006] [RS_UCM_00007] [RS_UCM_00008] [RS_UCM_00010] [RS_UCM_00011] [RS_UCM_00012] [RS_UCM_00013] [RS_UCM_00014] [RS_UCM_00015] [RS_UCM_00016] [RS_UCM_00017] [RS_UCM_00018] [RS_UCM_00020] [RS_UCM_00026] [RS_UCM_00027]
[RS_Main_00503]	AUTOSAR shall provide a Software Platform that supports adaptation of communication topology after production	[RS_UCM_00001] [RS_UCM_00002] [RS_UCM_00003] [RS_UCM_00004] [RS_UCM_00005] [RS_UCM_00006] [RS_UCM_00007] [RS_UCM_00008] [RS_UCM_00010] [RS_UCM_00011] [RS_UCM_00012] [RS_UCM_00013] [RS_UCM_00014] [RS_UCM_00015] [RS_UCM_00016] [RS_UCM_00017] [RS_UCM_00018] [RS_UCM_00020] [RS_UCM_00026] [RS_UCM_00027]

8 References

[1] System Template
AUTOSAR_TPS_SystemTemplate

[2] Glossary
AUTOSAR_TR_Glossary

[3] Main Requirements
AUTOSAR_RS_Main