Document Title	Requirements on Operating System Interface
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	718

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R24-11

Document Change History			
Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	• Minor changes, document clean up
2023-11-23	R23-11	AUTOSAR Release Management	• Changed [RS_OSI_00201], [RS_OSI_00202], [RS_OSI_00203], [RS_OSI_00206] to point to appropriate AUTOSAR_FO_RS_Safety requirements. • Added [RS_OSI_00210], [RS_OSI_00211] about tracing support.
2022-11-24	R22-11	AUTOSAR Release Management	• Changed [RS_OSI_00103] to point to [RS_Main_00060]
2021-11-25	R21-11	AUTOSAR Release Management	• Added [RS_OSI_00209]
2020-11-30	R20-11	AUTOSAR Release Management	• Uptrace to AUTOSAR_FO_RS_Safety document • Clarified Execution Management description • Moved time-triggered execution to AUTOSAR_AP_RS_ExecutionManagement
2019-11-28	R19-11	AUTOSAR Release Management	• Updated document structure • Changed Document Status from Final to published

▽

△

2019-03-29	19-03	AUTOSAR Release Management	• Added: use case for [RS_OSI_00201] and [RS_OSI_00202]
2018-10-31	18-10	AUTOSAR Release Management	• Removed: RS_OSI_00102 and RS_OSI_00105 • Added: [RS_OSI_00207], [RS_OSI_00208].
2018-03-29	18-03	AUTOSAR Release Management	• Removed: RS_OSI_00101, RS_OSI_00200 and RS_OSI_00205. • Added: [RS_OSI_00103].
2017-10-27	17-10	AUTOSAR Release Management	• Minor changes, document clean up
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.

Contents

1 Scope of Document

This document specifies the requirements of AUTOSAR Adaptive Platform on the Operating System that is part of the Foundation in the AUTOSAR Adaptive Platform.

2 Conventions to be used

2.1 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]).

2.2 Requirements Guidelines

2.2.1 Requirements status

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

- [RS_OSI_00204]
- [RS_OSI_00208]
- [RS_OSI_00209]

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

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to this Document.

Abbreviation / Acronym:	Description:
Operating System	Software responsible for managing `Processes` on a `Machine` and for providing an interface to hardware resources.
Operating System Interface (OSI)	The `Functional Cluster` within the `AUTOSAR Adaptive Platform` which specifies the `Operating System` interface used by `Adaptive Applications` and by the `AUTOSAR Adaptive Platform` itself.

Table 3.1: Technical Terms

The following technical terms used in this document are defined in the corresponding document mentioned in the table below.

Abbreviation / Acronym:	Description:
Execution Management	see [2] Requirements on Execution Management
Adaptive Application	see [3] AUTOSAR Glossary
ARTI	see [3] AUTOSAR Glossary
AUTOSAR Adaptive Platform	see [3] AUTOSAR Glossary
Functional Cluster	see [3] AUTOSAR Glossary
Machine	see [3] AUTOSAR Glossary
Process	see [3] AUTOSAR Glossary
Task	In case of POSIX a Task is called thread or pthread. see [3] AUTOSAR Glossary

Table 3.2: Reference to Technical Terms

4 Requirements Specification

This chapter describes all requirements for the selection of the Operating System for the AUTOSAR Adaptive Platform.

4.1 Functional Overview

The Operating System is responsible for run-time resource management (including time) for all applications on and within the AUTOSAR Adaptive Platform. This includes not only the Adaptive Applications that run on top of ARA provided by AUTOSAR Adaptive Platform, but also the Functional Clusters that constitute the platform, which are also implemented as applications. The Operating System functions in cooperation with Execution Management which is responsible for platform initialization and the start-up / shut-down of applications.

Note that this Operating System Interface requirement specification contains two different categories. The first category contains the requirements that are directly needed by the Adaptive Applications. The other category contains the ones that are needed by the AUTOSAR Adaptive Platform to realize implementation of Functional Clusters, especially the required mechanisms are difficult or inefficient to be implemented by other software entity than the Operating System. The most notable such Functional Cluster requiring the various Operating System mechanisms is Execution Management.

4.2 Functional Requirements

4.2.1 Assumption of Use

This section describes execution environment of the AUTOSAR Adaptive Platform that is directly or indirectly used to run the Adaptive Applications. This execution environment is not requirements to the AUTOSAR Adaptive Platform in the strict sense, but rather assumptions on properties of Adaptive Application running on the AUTOSAR Adaptive Platform. These assumptions are used to motivate the further requirements, and provide hints for Adaptive Application developers to check if their use cases are covered in this specification document and which specific requirements are derived from those use cases.

The Operating System section defines requirements on the Operating System that Processes can consider fulfilled in order to achieve their function.

4.2.2 General Requirements

This section describes APIs that should be exposed to `Processes` on the `AUTOSAR Adaptive Platform` that closely relates to the `Operating System` functionality. Some libraries in the system may be considered by a Developer to be part of the `Operating System` without strictly-speaking belonging to it, for example the C++ runtime libraries. For the purpose of this section, the `Operating System` kernel and its libraries are considered to be a single entity.

[RS_OSI_00100] POSIX PSE51 Compliance

Upstream requirements: RS_Main_00002, RS_Main_00050, RS_Main_00150, RS_Main_00420

Description:	The foundation of the `Operating System` provided to `Adaptive Application Processes` shall be POSIX-compliant as defined by PSE51.
Rationale:	The defined functionality of the POSIX profile PSE51 defined by IEEE1003.13 [4] is provided by various off-the-shelf `Operating Systems`. The PSE51 profile is intended for embedded systems, with a single multi-threaded `Process`, no file system, no user and group support and only selected options from more general IEEE1003.1 [5], which is the well-known POSIX standard. PSE51 offers functions for basic synchronized I/O, high-resolution timer, signals, semaphores, shared memory and threads. As the envisioned `Adaptive Application` software components will not require to fork new `Processes` themselves, and only need limited direct access to files. Hence for `Adaptive Applications` the PSE51 profile is thought to be sufficient.
Dependencies:	–
Use Case:	Application portability.
Supporting Material:	IEEE1003.13 [4] and IEEE1003.1 [5]

[RS_OSI_00103] The Operating System Interface shall support C++.

Upstream requirements: RS_Main_00060

Description:	The `Operating System` interface shall support C++11.
Rationale:	`Processes` are written in C++ and interfaces are expected to conform to C++11. Note that the POSIX API consists of C functions which can be invoked from a C++ program.
Dependencies:	–
Use Case:	–
Supporting Material:	–

[RS_OSI_00104] The Operating System Interface shall support the reaction on Process-external stimuli from devices.

Upstream requirements: RS_Main_00050, RS_Main_00060, RS_Main_00460

⌈

Description:	The `Operating System` shall enable `Processes` to react on external stimuli from devices.
Rationale:	`Adaptive Application` will react on reception of functional data, signals and timers from the platform. Certain computations shall be executed in reaction on these application-external stimuli.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋

[RS_OSI_00105] The Operating System Interface shall support the start of Execution Management.

Upstream requirements: RS_Main_00049, RS_Main_00460

⌈

Description:	The `Operating System` shall provide means to start the `Execution Management Functional Cluster` as first `Process`.
Rationale:	`Execution Management` is responsible for startup and shutdown of all `Processes` of the `AUTOSAR Adaptive Platform`.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋

4.2.3 Operating System Requirements

[RS_OSI_00201] The Operating System shall provide mechanisms for system memory budgeting.

Upstream requirements: RS_Main_00002, RS_Main_00010, RS_Main_00011, RS_Main_00012, RS_Main_00106, RS_Main_00150, RS_Main_00514, RS_SAF_10008

⌈

Description:	The `Operating System` shall provide mechanisms to configure memory budgeting for each `Process` or for groups of `Processes`.
Rationale:	In order to ensure resource availability in the context of a multi-`Process` system, the system integrator/architect may require a set of tools to configure memory budgeting for each `Process` or for groups of `Processes`.
Dependencies:	–
Use Case:	security - protection against DoS attacks - resource starvation types.
Supporting Material:	–

⌋

[RS_OSI_00202] The Operating System shall provide mechanisms for CPU time budgeting.

Upstream requirements: RS_Main_00002, RS_Main_00010, RS_Main_00011, RS_Main_00012, RS_Main_00106, RS_Main_00150, RS_Main_00514, RS_SAF_10008

⌈

Description:	The `Operating System` shall provide mechanisms to configure resource budgeting in terms of CPU time for each `Process` or group of `Processes`.
Rationale:	In order to ensure schedulability in the context of a multi-`Process` system, the system integrator/architect may require a set of tools to configure CPU time allocated for each `Process` or for groups of `Processes`.
Dependencies:	–
Use Case:	security - protection against DoS attacks - resource starvation types.
Supporting Material:	–

⌋

[RS_OSI_00203] The Operating System should provide mechanisms for binding Processes to CPU cores.

Upstream requirements: RS_Main_00002, RS_Main_00010, RS_Main_00011, RS_Main_00012, RS_Main_00050, RS_Main_00106, RS_Main_00514, RS_SAF_10008

⌈

Description:	The `Operating System` should provide mechanisms for binding individual `Process` or groups of `Processes` to CPU cores.
Rationale:	In order to ensure correct `Task` schedulability, the system integrator may require a set of tools to configure the CPU affinity of `Processes`. In a multi-core system, it may be relevant to ensure some `Processes` can only run on some CPU cores, to allow other less- or differently-restricted `Processes` to concurrently progress.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋

[RS_OSI_00204] The Operating System shall support authorized operating system object access for the software entities which are allowed to do so.

Status: DRAFT

Upstream requirements: RS_Main_00010, RS_Main_00170, RS_Main_00514

⌈

Description:	The `Operating System` shall provide access rights and permissions mechanisms to achieve secure data access and data exchange.
Rationale:	The `Operating System` consists of a collection of hardware and software objects, e.g. pipes, files. Safety or/and Security related requirements may be imposed to grant special access rights and permissions in order to avoid unauthorized access to communication channels or to ensure exclusive access to the `Process`-specific data stored persistently.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋

[RS_OSI_00206] The Operating System shall provide multi-Process support for isolation of applications.

Upstream requirements: RS_Main_00010, RS_Main_00049, RS_Main_00106, RS_SAF_10008, RS_SAF_10037

⌈

Description:	The `Operating System` shall provide mechanisms to let multiple `Processes` run isolated from each other.
Rationale:	Each `Process` may have a different robustness, safety and security level. As a consequence, an incorrect memory access from one `Process` execution shall not result in a corruption of memory in another `Process`, unless the data area is explicitly shared. In addition, a `Process` may not access or read data from another `Process` without explicit data sharing.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋

[RS_OSI_00207] The Operating System shall provide the capability to share code and data in an implicit manner.

Upstream requirements: RS_Main_00410, RS_Main_00150, RS_Main_00503

⌈

Description:	The `Operating System` shall provide mechanisms to run the same code and associated data copy in multiple `Processes` by using a shared object mechanism, where data is either read-only or a `Process`-private copy.
Rationale:	To allow more efficient memory usage both in runtime memory and non-volatile storage, as well as making platform updates faster, applications may be linked against shared code and data objects. Using shared objects is not intended however to be mandatory for use by the `AUTOSAR Adaptive Platform`, as security, safety or other constraints may require not to use this feature.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋

[RS_OSI_00208] The Operating System shall allow Processes to access only required functionality.

Status:　　　　　　　　DRAFT

Upstream requirements:　RS_Main_00010, RS_Main_00514

⌈

Description:	The `Operating System` shall ensure that either a `Process` does not access `Operating System` functionality via a system call at all, or the system call has to be authorized.
Rationale:	A `Process` with full access to the Platform could cause significant damage.
Dependencies:	–
Use Case:	–
Supporting Material:	This requirement was visible before 18-10 Specification Release as RS_SEC_05009.

⌋

[RS_OSI_00209] The Operating System shall support error handling including unrecoverable errors.

Status:　　　　　　　　DRAFT

Upstream requirements:　RS_Main_00010, RS_Main_00514

⌈

Description:	The `Operating System` shall support error handling including unrecoverable errors.
Rationale:	The `Operating System` may face conditions where has no mechanism to recover the system. These situations are typically expected to result from a misconfigured system and therefore a suitable response might be to halt startup so that the misconfiguration can be resolved outside of the `Operating System` context.
Dependencies:	–
Use Case:	The `Operating System` can not start `Execution Management` and hence the platform as a whole cannot be started - it is not possible to recover from this situation hence The `Operating System` shall halt startup.
Supporting Material:	This requirement was visible before 18-10 Specification Release as RS_SEC_05009.

⌋

[RS_OSI_00210] The Operating System shall provide a mechanism to observe scheduling of CPU time to Task, Process

Upstream requirements: RS_Main_01026

⌈

Description:	The `Operating System` shall provide a mechanism to observe scheduling of CPU time to `Tasks`, `Processes`, etc.
Rationale:	It is mandatory to detect shortage in CPU resources and identify causing `Tasks` and `Processes`. For this the CPU load has to be observable. To identify tasks and `Processes` that influences this CPU load the start, stop and switching of `Tasks` and `Processes` shall be observable, typically through additional `ARTI`- hooks.
Dependencies:	–
Use Case:	In order to support `Operating System` awareness of tracing solutions, they require an interface to receive this information from the `Operating System`.

⌋

[RS_OSI_00211] The Operating System shall provide a mechanism to export low-level scheduling and trace information to applications.

Upstream requirements: RS_Main_01026

⌈

Description:	The `Operating System` shall provide a mechanism to export low-level scheduling and trace information to applications.
Rationale:	Similar to observing the scheduling of CPU time and its exporting through hooks, the same type of information can be provided to ara::log, if exported by the `Operating System`.
Dependencies:	–
Use Case:	Provide additional `Operating System` information in ara::log.

⌋

4.3 Non-Functional Requirements

No content.

5 Requirements Tracing

The following table references the features specified in [6] and links to the fulfillments of these. Please note that if column "Satisfied by" is empty for a specific requirement this means that this requirement is not fulfilled by this document. Likewise, an entry of [RS_OSI_NA] indicates that the source requirement has been evaluated as "not applicable" to the Operating System Interface.

Requirement	Description	Satisfied by
[RS_Main_00002]	AUTOSAR shall provide a software platform for high performance computing platforms	[RS_OSI_00100] [RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203]
[RS_Main_00010]	Safety Mechanisms	[RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203] [RS_OSI_00204] [RS_OSI_00206] [RS_OSI_00208] [RS_OSI_00209]
[RS_Main_00011]	Mechanisms for Reliable Systems	[RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203]
[RS_Main_00012]	Highly Available Systems Support	[RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203]
[RS_Main_00049]	AUTOSAR shall provide an Execution Management for running multiple applications	[RS_OSI_00105] [RS_OSI_00206]
[RS_Main_00050]	AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows	[RS_OSI_00100] [RS_OSI_00104] [RS_OSI_00203]
[RS_Main_00060]	Standardized Application Communication Interface	[RS_OSI_00103] [RS_OSI_00104]
[RS_Main_00106]	AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation	[RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203] [RS_OSI_00206]
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	[RS_OSI_00100] [RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00207]
[RS_Main_00170]	AUTOSAR shall provide secure access to ECU data and services	[RS_OSI_00204]
[RS_Main_00410]	AUTOSAR shall provide specifications for routines commonly used by Application Software to support sharing and optimization	[RS_OSI_00207]
[RS_Main_00420]	AUTOSAR shall use established software standards and consolidate de-facto standards for basic software functionality	[RS_OSI_00100]
[RS_Main_00460]	AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level	[RS_OSI_00104] [RS_OSI_00105]
[RS_Main_00503]	AUTOSAR shall support change of communication and application software at runtime.	[RS_OSI_00207]
[RS_Main_00514]	System Security Support	[RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203] [RS_OSI_00204] [RS_OSI_00208] [RS_OSI_00209]
[RS_Main_01026]	AUTOSAR shall support tracing and profiling on the target and onboard	[RS_OSI_00210] [RS_OSI_00211]

▽

△

Requirement	Description	Satisfied by
[RS_SAF_10008]	AUTOSAR shall provide mechanisms to support safe resource management for the AUTOSAR Adaptive Platform functional-clusters, applications and services and AUTOSAR Classic Platform basic software modules and software components.	[RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203] [RS_OSI_00206]
[RS_SAF_10037]	AUTOSAR shall provide mechanisms to prevent unintended alteration of data.	[RS_OSI_00206]

Table 5.1: Requirements Tracing

5.1   Not applicable requirements

[RS_OSI_NA]

Status:	DRAFT
Upstream requirements:	RS_Main_00026, RS_Main_00030, RS_Main_00080, RS_Main_00180, RS_Main_00190, RS_Main_00230, RS_Main_00250, RS_Main_00260, RS_Main_00261, RS_Main_00270, RS_Main_00280, RS_Main_00300, RS_Main_00301, RS_Main_00310, RS_Main_00320, RS_Main_00340, RS_Main_00350, RS_Main_00360, RS_Main_00440, RS_Main_00445, RS_Main_00490, RS_Main_00491, RS_Main_00500, RS_Main_00501, RS_Main_00507, RS_Main_00510, RS_Main_00511, RS_Main_00650, RS_Main_00653, RS_Main_01001, RS_Main_01002, RS_Main_01003, RS_Main_01004, RS_Main_01005, RS_Main_01007, RS_Main_01008, RS_Main_01025

⌈These requirements are not applicable as they are not within the scope of this release or this document.⌋

6 References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] Requirements on Execution Management
AUTOSAR_AP_RS_ExecutionManagement

[3] Glossary
AUTOSAR_FO_TR_Glossary

[4] IEEE Standard for Information Technology- Standardized Application Environment
Profile (AEP)-POSIX Realtime and Embedded Application Support
https://standards.ieee.org/findstds/standard/1003.13-2003.html

[5] Standard for Information Technology–Portable Operating System Interface
(POSIX(R)) Base Specifications, Issue 7
http://pubs.opengroup.org/onlinepubs/9699919799/

[6] Main Requirements
AUTOSAR_FO_RS_Main

A Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

A.1 Traceable item history of this document according to AUTOSAR Release R24-11

A.1.1 Added Requirements in R24-11

A.1.2 Changed Requirements in R24-11

Number	Heading
[RS_OSI_00100]	POSIX PSE51 Compliance
[RS_OSI_00103]	The Operating System Interface shall support C++.
[RS_OSI_00104]	The Operating System Interface shall support the reaction on Process-external stimuli from devices.
[RS_OSI_00105]	The Operating System Interface shall support the start of Execution Management.
[RS_OSI_00201]	The Operating System shall provide mechanisms for system memory budgeting.
[RS_OSI_00202]	The Operating System shall provide mechanisms for CPU time budgeting.
[RS_OSI_00203]	The Operating System should provide mechanisms for binding Processes to CPU cores.
[RS_OSI_00204]	The Operating System shall support authorized operating system object access for the software entities which are allowed to do so.
[RS_OSI_00206]	The Operating System shall provide multi-Process support for isolation of applications.
[RS_OSI_00207]	The Operating System shall provide the capability to share code and data in an implicit manner.
[RS_OSI_00208]	The Operating System shall allow Processes to access only required functionality.
[RS_OSI_00209]	The Operating System shall support error handling including unrecoverable errors.
[RS_OSI_00210]	The Operating System shall provide a mechanism to observe scheduling of CPU time to Task, Process

▽

△

Number	Heading
[RS_OSI_00211]	The Operating System shall provide a mechanism to export low-level scheduling and trace information to applications.

Table A.1: Changed Requirements in R24-11

A.1.3 Deleted Requirements in R24-11

A.2 Traceable item history of this document according to AUTOSAR Release R23-11

A.2.1 Added Requirements in R23-11

Number	Heading
[RS_OSI_00210]	The `Operating System` shall provide a mechanism to observe scheduling of CPU time to tasks, processes.
[RS_OSI_00211]	The `Operating System` shall provide a mechanism to export low-level scheduling and trace information to applications.

Table A.2: Added Requirements in R23-11

A.2.2 Changed Requirements in R23-11

Number	Heading
[RS_OSI_00201]	The Operating System shall provide mechanisms for system memory budgeting.
[RS_OSI_00202]	The Operating System shall provide mechanisms for CPU time budgeting.
[RS_OSI_00203]	The Operating System should provide mechanisms for binding processes to CPU cores.
[RS_OSI_00206]	The Operating System shall provide multi-process support for isolation of applications.

Table A.3: Changed Requirements in R23-11

A.2.3 Deleted Requirements in R23-11

A.3 Traceable item history of this document according to AUTOSAR Release R22-11

A.3.1 Added Advisories in R22-11

A.3.2 Changed Advisories in R22-11

A.3.3 Deleted Advisories in R22-11

A.3.4 Added Constraints in R22-11

A.3.5 Changed Constraints in R22-11

A.3.6 Deleted Constraints in R22-11

A.3.7 Added Requirements in R22-11

A.3.8 Changed Requirements in R22-11

Number	Heading
[RS_OSI_00103]	The Operating System Interface shall support C++.
[RS_OSI_00104]	The Operating System Interface shall support the reaction on process-external stimuli from devices.
[RS_OSI_NA]	

Table A.4: Changed Requirements in R22-11

A.3.9 Deleted Requirements in R22-11