

Document Title	Adaptive Platform Demonstrator Release Notes
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	865

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R22-11

Document Change History			
Date	Release	Changed by	Description
2022-11-24	R22-11	AUTOSAR Release Management	<ul style="list-style-type: none"> R22-11 APD Pre-Release
2022-04-22	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> R21-11 APD Release
2021-11-25	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> R21-11 APD Pre-Release
2021-04-16	R20-11	AUTOSAR Release Management	<ul style="list-style-type: none"> R20-11 APD Release
2020-11-30	R20-11	AUTOSAR Release Management	<ul style="list-style-type: none"> R20-11 APD Pre-Release
2019-11-28	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> Release R19-11
2019-03-29	19-03	AUTOSAR Release Management	<ul style="list-style-type: none"> Release R19-03
2018-12-07	18-10-1	AUTOSAR Release Management	<ul style="list-style-type: none"> Release R18-10-1
2018-11-02	18-10	AUTOSAR Release Management	<ul style="list-style-type: none"> Release R18-10

2018-05-02	18-03	AUTOSAR Release Management	<ul style="list-style-type: none">• Release R18-03
2017-12-11	17-10	AUTOSAR Release Management	<ul style="list-style-type: none">• Release R17-10

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	Introduction	6
2	Related documentation	7
2.1	Input documents & related standards and norms	7
3	R22-11 APD Pre-Release	8
3.1	Overview	8
3.2	Licenses	9
3.3	Known issues	9
4	R21-11 APD Release	11
4.1	Overview	11
4.2	Licenses	12
4.3	Implemented changes	12
4.4	Known issues	12
5	R21-11 APD Pre-Release	14
5.1	Overview	14
5.2	Licenses	15
5.3	Known issues	15
6	R20-11 APD Release	17
6.1	Overview	17
6.2	Licenses	18
6.3	Implemented changes	18
6.4	Known issues	18
7	R20-11 APD Pre-Release	20
7.1	Overview	20
7.2	Licenses	21
7.3	Known issues	21
8	R19-11	23
8.1	Overview	23
8.2	Known issues	24
9	R19-03	26
9.1	Overview	26
9.2	Known issues	27
10	R18-10-1	29
10.1	Implemented changes	29
10.2	Known issues	29
11	R18-10	30

11.1	Overview	30
11.2	Known issues	31
12	R18-03	32
12.1	Overview	32
12.2	Known issues	33
13	R17-10	34
13.1	Overview	34
13.2	Known issues	35
A	Appendix	36
A.1	Licenses	36
A.1.1	Usage of W3C XML schema	36

1 Introduction

This document details the release history of the Adaptive Platform Demonstrator. It is an adjunct to [\[1\]](#).

2 Related documentation

2.1 Input documents & related standards and norms

- [1] Adaptive Platform Release Overview
AUTOSAR_TR_AdaptivePlatformReleaseOverview
- [2] Source of Adaptive Platform Demonstrator
AUTOSAR_SRC_AdaptivePlatformDemonstrator
- [3] Testdata of Adaptive Platform Demonstrator
AUTOSAR_SRC_AdaptivePlatformDemonstratorTestdata
- [4] Adaptive Platform Demonstrator Foss Results
AUTOSAR_TR_APDemonstratorFossResults

3 R22-11 APD Pre-Release

3.1 Overview

This chapter provides an overview of the AUTOSAR “R22-11 Adaptive Platform Demonstrator Pre-Release”. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R22-11, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code: AUTOSAR_SRC_AdaptivePlatformDemonstrator.zip [2]
- Additional data files: AUTOSAR_SRC_AdaptivePlatformDemonstratorTestdata.zip [3]
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, or any of the other sample applications contained within the release archive.
- Explanatory material that details and documents the software design of the included Functional Clusters, see table 3.1.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
924	AUTOSAR_EXP_FCDesignPlatformHealthManagement
1016	AUTOSAR_EXP_FCDesignStateManagement
905	AUTOSAR_EXP_FCDesignTimeSynchronization
894	AUTOSAR_EXP_FCDesignUpdateAndConfigurationManagement

Table 3.1: FCDesign documents contained in this release

For this release, the Demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

3.2 Licenses

See [A.1.1](#).

3.3 Known issues

Please note that most known Functional Clusters' deficiencies are given in their respective FCDesign documents.

The known issues listed here are either affecting the overall formation of the Demonstrator, or have no other suitable location into which to put them.

- Some of the `ara::core` types' implementations violate their specifications because they allow implicit conversion from and/or to the `std::` types on which they are modeled. For instance, `ara::core::Vector` is currently implemented as a type alias for `std::vector` and can therefore be used interchangeably with it – this is not standardized though and should not be assumed by user code.
- The classes `ara::core::Future` and `ara::core::Promise` contain nested type aliases, which are not specified. `ara::core::Promise` additionally also contains the unspecified function `Promise::set_exception`.
- The asynchronous stackable continuations feature of `ara::core::Future` (using `Future::then`) is implemented in a way that makes it neither asynchronous, nor a continuation, nor stackable.
- The implementations of the classes `ara::core::Optional` and `ara::core::StringView` perform heap allocations, which is not allowed by the specification. This issue also makes the affected member functions unsuitable for use in constant expressions.
- There is no implementation of `ara::core::Variant`.
- Some Functional Clusters' implementations do not accurately reflect the state of their respective specifications. In particular, the implementation of the DIAG Functional Cluster is still based on the R18-10 standard.
- The values serialized into JSON files by `ara::per` lose their type information; this means that `ara::per` cannot know which data type is actually meant for an entry. Consequently, the implementation now allows conversion into any data type with `KVS::GetValue()`, and never returns `PerErrc::kDataTypeMismatch`.
- Some Adaptive Applications that use `ara::com` with SOME/IP bindings throw an exception when terminating.
- The SystemTest application wrongly marks the test item STS_E2E_00001 as “passed”, even though it should always fail due to missing implementations in the E2E subsystem. This happens due to the fact that SystemTest cannot determine

whether E2E is in fact available. Hence it assumes that E2E is available and treats any successful data transmission as proof that data can be sent with E2E protection.

- The SystemTest application reports a number of other test failures.

4 R21-11 APD Release

4.1 Overview

This chapter provides an overview of the AUTOSAR “R21-11 Adaptive Platform Demonstrator Release”. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R21-11, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code: AUTOSAR_SRC_AdaptivePlatformDemonstrator.zip [2]
- Additional data files: AUTOSAR_SRC_AdaptivePlatformDemonstratorTestdata.zip [3]
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, or any of the other sample applications contained within the release archive.
- Results of a scan for Open Source licenses in all 3rd-party packages pulled in by the Yocto framework when using the release archive for building a minimal bootable image: AUTOSAR_TR_APDemonstratorFossResults.zip [4]

Please note that this scan report is for informational purposes only, and AUTOSAR does not provide any guarantee about its correctness or completeness.

- Explanatory material that details and documents the software design of the included Functional Clusters, see table 4.1.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
924	AUTOSAR_EXP_FCDesignPlatformHealthManagement
1016	AUTOSAR_EXP_FCDesignStateManagement
905	AUTOSAR_EXP_FCDesignTimeSynchronization
894	AUTOSAR_EXP_FCDesignUpdateAndConfigurationManagement

Table 4.1: FCDesign documents contained in this release

For this release, the Demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

4.2 Licenses

See [A.1.1](#).

4.3 Implemented changes

Changes made include the following:

- Fixed some issues with `ara::com` DDS bindings
- Added the AUTOSAR XML Schema `AUTOSAR_00050.xsd` to `ara-gen`
- Extended `ara-gen` to generate artifacts according to AUTOSAR XML Schema `AUTOSAR_00050.xsd`
- Updated all components' ARXML files to conform against the AUTOSAR XML Schema `AUTOSAR_00050.xsd`

4.4 Known issues

Please note that most known Functional Clusters' deficiencies are given in their respective FCDesign documents.

The known issues listed here are either affecting the overall formation of the Demonstrator, or have no other suitable location into which to put them.

- Trying to use the SDK to build the radar/fusion applications with DDS bindings for `ara::com` will fail with an error message. In order to work around this issue, you need to define an environment variable with e.g. `export TAO_IDL_PREPROCESSOR=/usr/bin/g++` prior to building.
- Some of the `ara::core` types' implementations violate their specifications because they allow implicit conversion from and/or to the `std::` types on which they are modeled. For instance, `ara::core::Vector` is currently implemented as a type alias for `std::vector` and can therefore be used interchangeably with it – this is not standardized though and should not be assumed by user code.
- The classes `ara::core::Future` and `ara::core::Promise` contain nested type aliases, which are not specified. `ara::core::Promise` additionally also contains the unspecified function `Promise::set_exception`.

- The asynchronous stackable continuations feature of `ara::core::Future` (using `Future::then`) is implemented in a way that makes it neither asynchronous, nor a continuation, nor stackable.
- The implementations of the classes `ara::core::Optional` and `ara::core::StringView` perform heap allocations, which is not allowed by the specification. This issue also makes the affected member functions unsuitable for use in constant expressions.
- There is no implementation of `ara::core::Variant`.
- Some Functional Clusters' implementations do not accurately reflect the state of their respective specifications. In particular, the implementation of the DIAG Functional Cluster is still based on the R18-10 standard.
- The values serialized into JSON files by `ara::per` lose their type information; this means that `ara::per` cannot know which data type is actually meant for an entry. Consequently, the implementation now allows conversion into any data type with `KVS::GetValue()`, and never returns `PerErrc::kDataTypeMismatch`.
- Some Adaptive Applications that use `ara::com` with SOME/IP bindings throw an exception when terminating.
- The SystemTest application wrongly marks the test item STS_E2E_00001 as "passed", even though it should always fail due to missing implementations in the E2E subsystem. This happens due to the fact that SystemTest cannot determine whether E2E is in fact available. Hence it assumes that E2E is available and treats any successful data transmission as proof that data can be sent with E2E protection.
- The SystemTest application reports a number of other test failures.

5 R21-11 APD Pre-Release

5.1 Overview

This chapter provides an overview of the AUTOSAR “R21-11 Adaptive Platform Demonstrator Pre-Release”. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R21-11, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code: AUTOSAR_SRC_AdaptivePlatformDemonstrator.zip [2]
- Additional data files: AUTOSAR_SRC_AdaptivePlatformDemonstratorTestdata.zip [3]
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, or any of the other sample applications contained within the release archive.
- Explanatory material that details and documents the software design of the included Functional Clusters, see table 5.1.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
924	AUTOSAR_EXP_FCDesignPlatformHealthManagement
1016	AUTOSAR_EXP_FCDesignStateManagement
905	AUTOSAR_EXP_FCDesignTimeSynchronization
894	AUTOSAR_EXP_FCDesignUpdateAndConfigurationManagement

Table 5.1: FCDesign documents contained in this release

With this release, the document AUTOSAR_EXP_FCDesignRESTfulCommunication has been removed from the standard; the corresponding implementation has been kept and turned into an implementation detail of the APD because it is needed by the implementation of ara::ucm. At the same time, the document AUTOSAR_EXP_FCDesignStateManagement has been newly introduced, together with an implementation of the ara::sm Functional Cluster.

For this release, the Demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

5.2 Licenses

See [A.1.1](#).

5.3 Known issues

Please note that most known Functional Clusters' deficiencies are given in their respective FCDesign documents.

The known issues listed here are either affecting the overall formation of the Demonstrator, or have no other suitable location into which to put them.

- Trying to use the SDK to build the radar/fusion applications with DDS bindings for `ara::com` will fail with an error message. In order to work around this issue, you need to define an environment variable with e.g. `export TAO_IDL_PREPROCESSOR=/usr/bin/g++` prior to building.
- Some of the `ara::core` types' implementations violate their specifications because they allow implicit conversion from and/or to the `std::` types on which they are modeled. For instance, `ara::core::Vector` is currently implemented as a type alias for `std::vector` and can therefore be used interchangeably with it – this is not standardized though and should not be assumed by user code.
- The classes `ara::core::Future` and `ara::core::Promise` contain nested type aliases, which are not specified. `ara::core::Promise` additionally also contains the unspecified function `Promise::set_exception`.
- The asynchronous stackable continuations feature of `ara::core::Future` (using `Future::then`) is implemented in a way that makes it neither asynchronous, nor a continuation, nor stackable.
- The implementations of the classes `ara::core::Optional` and `ara::core::StringView` perform heap allocations, which is not allowed by the specification. This issue also makes the affected member functions unsuitable for use in constant expressions.
- There is no implementation of `ara::core::Variant`.
- Some Functional Clusters' implementations do not accurately reflect the state of their respective specifications. In particular, the implementation of the DIAG Functional Cluster is still based on the R18-10 standard.
- The values serialized into JSON files by `ara::per` lose their type information; this means that `ara::per` cannot know which data type is actually

meant for an entry. Consequently, the implementation now allows conversion into any data type with `KVS::GetValue()`, and never returns `PerErrc::kDataTypeMismatch`.

- Some Adaptive Applications that use `ara::com` with SOME/IP bindings throw an exception when terminating.
- The SystemTest application wrongly marks the test item STS_E2E_00001 as “passed”, even though it should always fail due to missing implementations in the E2E subsystem. This happens due to the fact that SystemTest cannot determine whether E2E is in fact available. Hence it assumes that E2E is available and treats any successful data transmission as proof that data can be sent with E2E protection.
- The SystemTest application reports a number of other test failures.

6 R20-11 APD Release

6.1 Overview

This chapter provides an overview of the AUTOSAR “R20-11 Adaptive Platform Demonstrator Release”. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R20-11, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code: AUTOSAR_SRC_AdaptivePlatformDemonstrator.zip [2]
- Additional data files: AUTOSAR_SRC_AdaptivePlatformDemonstratorTestdata.zip [3]
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, or any of the other sample applications contained within the release archive.
- Explanatory material that details and documents the software design of the included Functional Clusters, see table 6.1.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
924	AUTOSAR_EXP_FCDesignPlatformHealthManagement
904	AUTOSAR_EXP_FCDesignRESTfulCommunication
905	AUTOSAR_EXP_FCDesignTimeSynchronization
894	AUTOSAR_EXP_FCDesignUpdateAndConfigurationManagement

Table 6.1: FCDesign documents contained in this release

For this release, the Demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

6.2 Licenses

See [A.1.1](#).

6.3 Implemented changes

Changes made include the following:

- Added the AUTOSAR XML Schema `AUTOSAR_00049.xsd` to `ara-gen`
- Extended `ara-gen` to generate artifacts according to AUTOSAR XML Schema `AUTOSAR_00049.xsd`
- Updated all components' ARXML files to conform against the AUTOSAR XML Schema `AUTOSAR_00049.xsd`

6.4 Known issues

Please note that most known Functional Clusters' deficiencies are given in their respective FCDesign documents.

The known issues listed here are either affecting the overall formation of the Demonstrator, or have no other suitable location into which to put them.

- The “minimal image” that is created by the recipe `core-image-apd-minimal` has an invalid IP address configured in its `/etc/network/interfaces`, which results in a system that has no IP address for its external network interface.
- Some of the `ara::core` types' implementations violate their specifications because they allow implicit conversion from and/or to the `std::` types on which they are modeled. For instance, `ara::core::Vector` is currently implemented as a type alias for `std::vector` and can therefore be used interchangeably with it – this is not standardized, though, and should not be assumed by user code.
- The classes `ara::core::Future` and `ara::core::Promise` contain nested type aliases, which are not specified. `ara::core::Promise` additionally also contains the unspecified function `Promise::set_exception`.
- The asynchronous stackable continuations feature of `ara::core::Future` (using `Future::then`) is implemented in a way that makes it neither asynchronous, nor a continuation, nor stackable.
- The implementations of the classes `ara::core::Optional` and `ara::core::StringView` perform heap allocations, which is not allowed by the specification. This issue also makes the affected member functions unsuitable for use in constant expressions.
- There is no implementation of `ara::core::Variant`.

- Some Functional Clusters' implementations do not accurately reflect the state of their respective specifications.
- The `runqemu-x86-autosar` script for running an image in QEMU may run into a kernel panic. This can be fixed by modifying the script `yocto-layers/meta-ara/scripts/runqemu-x86-autosar`, replacing `-cpu pentium2` in line 90 into `-cpu core2duo` if necessary.
- The values serialized into JSON files by `ara::per` lose their type information; this means that `ara::per` cannot know which data type is actually meant for an entry. Consequently, the implementation now allows conversion into any data type with `KVS::GetValue()`, and never returns `PerErrc::kDataTypeMismatch`.
- Some Adaptive Applications that are using `ara::com` with SOME/IP bindings throw an exception when terminating.
- The SystemTest application wrongly marks the test item STS_E2E_00001 as "passed", even though it should always fail due to missing implementations in the E2E subsystem. This is due to the fact that SystemTest cannot determine whether E2E is in fact available. It hence assumes that E2E is available and treats any successful data transmission as proof that data can be sent with E2E protection.
- The SystemTest application reports a number of other test failures.

7 R20-11 APD Pre-Release

7.1 Overview

This chapter provides an overview of the AUTOSAR “R20-11 Adaptive Platform Demonstrator Pre-Release”. This release is called “provisional” due to the fact that some quality measures on the source code have not yet concluded. AUTOSAR will perform these measures and attest the “provisional” release at a later point in time. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R20-11, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code: AUTOSAR_SRC_AdaptivePlatformDemonstrator.zip [2]
- Additional data files: AUTOSAR_SRC_AdaptivePlatformDemonstratorTestdata.zip [3]
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, or any of the other sample applications contained within the release archive.
- Explanatory material that details and documents the software design of the included Functional Clusters, see table 7.1.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
924	AUTOSAR_EXP_FCDesignPlatformHealthManagement
904	AUTOSAR_EXP_FCDesignREST
905	AUTOSAR_EXP_FCDesignTimeSynchronization
894	AUTOSAR_EXP_FCDesignUpdateAndConfigManagement

Table 7.1: FCDesign documents contained in this release

For this release, the Demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

7.2 Licenses

See [A.1.1](#).

7.3 Known issues

Please note that most known Functional Clusters' deficiencies are given in their respective FCDesign documents.

The known issues listed here are either affecting the overall formation of the Demonstrator, or have no other suitable location into which to put them.

- The “minimal image” that is created by the recipe `core-image-apd-minimal` has an invalid IP address configured in its `/etc/network/interfaces`, which results in a system that has no IP address for its external network interface.
- The ARXML files contained in this release still refer to version 00048 of the AUTOSAR schema, which is the one from the last release R19-11.
- Some of the `ara::core` types' implementations violate their specifications because they allow implicit conversion from and/or to the `std::` types on which they are modeled. For instance, `ara::core::Vector` is currently implemented as a type alias for `std::vector` and can therefore be used interchangeably with it – this is not standardized, though, and should not be assumed by user code.
- The classes `ara::core::Future` and `ara::core::Promise` contain nested type aliases, which are not specified. `ara::core::Promise` additionally also contains the unspecified function `Promise::set_exception`.
- The asynchronous stackable continuations feature of `ara::core::Future` (using `Future::then`) is implemented in a way that makes it neither asynchronous, nor a continuation, nor stackable.
- The implementations of the classes `ara::core::Optional` and `ara::core::StringView` perform heap allocations, which is not allowed by the specification. This issue also makes the affected member functions unsuitable for use in constant expressions.
- There is no implementation of `ara::core::Variant`.
- Some Functional Clusters' implementations do not accurately reflect the state of their respective specifications.
- Some Adaptive Applications that are using `ara::com` with SOME/IP bindings throw an exception when terminating.
- The SystemTest application wrongly marks the test item STS_E2E_00001 as “passed”, even though it should always fail due to missing implementations in the E2E subsystem. This is due to the fact that SystemTest cannot determine whether E2E is in fact available. It hence assumes that E2E is available and

treats any successful data transmission as proof that data can be sent with E2E protection.

- The SystemTest application reports a number of other test failures.

8 R19-11

8.1 Overview

This chapter provides an overview of the “AUTOSAR Adaptive Platform Demonstrator” provisional release R19-11. This release is called “provisional” due to the fact that some quality measures on the source code have not yet concluded. AUTOSAR will perform these measures and attest the “provisional” release at a later point in time. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R19-11, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code of the Demonstrator: `SRC_AdaptivePlatformDemonstrator.zip` [2]
- Additional data files: `SRC_AdaptivePlatformDemonstratorTestdata.zip` [3]
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, or any of the other sample applications contained within the release archive.
- Explanatory material that details and documents the software design of the included functional clusters, see table 8.1.

Please note that the document `AUTOSAR_EXP_FCDesignSecurityManagement` has been set to “obsolete” with this release, as it does not have any content anymore.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
924	AUTOSAR_EXP_FCDesignPlatformHealthManagement
904	AUTOSAR_EXP_FCDesignREST
882	AUTOSAR_EXP_FCDesignSecurityManagement
905	AUTOSAR_EXP_FCDesignTimeSynchronization
894	AUTOSAR_EXP_FCDesignUpdateAndConfigManagement

Table 8.1: FCDesign documents contained in this release

For this release, the Demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to

increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

8.2 Known issues

Please note that most known Functional Clusters' deficiencies are given in their respective FCDesign documents.

The known issues listed here are either affecting the overall formation of the Demonstrator, or have no other suitable location into which to put them.

- Some of the `ara::core` types' implementations violate their specifications because they allow implicit conversion from and/or to the `std::` types on which they are modeled. For instance, `ara::core::Vector` is currently implemented as a type alias for `std::vector` and can therefore be used interchangeably with it – this is not standardized, though, and should not be assumed by user code.
- The classes `ara::core::Future` and `ara::core::Promise` contain nested type aliases, which are not specified. `ara::core::Promise` additionally also contains the unspecified function `Promise::set_exception`.
- The asynchronous stackable continuations feature of `ara::core::Future` (using `Future::then`) is implemented in a way that makes it neither asynchronous, nor a continuation, nor stackable.
- The class `ara::core::StringView` is only `constexpr`-capable as specified when compiled with a suitable compiler (such as e.g. g++ 7.3.0) in C++17 mode. However, most components in this release are configured to be built in C++11 or C++14 mode and are thus unable to use `ara::core::StringView` in constant expressions.
- The implementations of the classes `ara::core::Optional` and `ara::core::StringView` perform heap allocations, which is not allowed by the specification. This issue also makes the affected member functions unsuitable for use in constant expressions.
- There is no implementation of `ara::core::Variant`.
- The `ara::core` types that are to replace `std::` types (such as `String`, `Vector`, `Map`, and `Array`) are not yet used throughout the whole codebase; some components are still using the corresponding `std::` types.
- Some Functional Clusters' implementations do not accurately reflect the state of their respective specifications.
- The ARXML files contained in this release still refer to version 00047 of the AUTOSAR schema, which is the one from the last release R19-03.

- The code generator performs validation against the AUTOSAR schema by default, but the AUTOSAR XML Schema requires the xml namespace definition file `xml.xsd` as `xsd:import`. This file is not part of this release and can be downloaded from <https://www.w3.org/2001/03/xml.xsd>.

A Yocto build will fetch this file from that URL and put it into the appropriate location within the build tree; any manual use of the generator or the AUTOSAR schema files will need to do this step manually, by downloading the file and putting it into the same directory as the AUTOSAR schema files, i.e. into `ara-api/apd/ara-gen/generator/parser/schema/`.

- The sample scenario consisting of the two applications `radar` and `fusion` given in `sample-applications/cm_provider_subscriber_scenario` does not always have the two applications successfully establish a connection with each other when they are located on different platform instances. It can be made to work when subsequently restarting them manually (e.g. by running `run/run_app.sh`).
It usually works well when both applications reside on the same platform instance (i.e., they are executed on the same machine and communicate via IPC).
- Some Adaptive Applications that are using `ara::com` with SOME/IP bindings throw an exception when terminating.
- The SystemTest application wrongly marks the test item `STS_E2E_00001` as “passed”, even though it should always fail due to missing implementations in the E2E subsystem. This is due to the fact that SystemTest cannot determine whether E2E is in fact available. It hence assumes that E2E is available and treats any successful data transmission as proof that data can be sent with E2E protection.
- The SystemTest application reports a number of other test failures.

9 R19-03

9.1 Overview

This chapter provides an overview of the “AUTOSAR Adaptive Platform Demonstrator” provisional release R19-03. This release is called “provisional” due to the fact that some quality measures on the source code have not yet concluded. AUTOSAR will perform these measures and attest the “provisional” release at a later point in time. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R19-03, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code of the Demonstrator: `SRC_AdaptivePlatformDemonstrator.zip`
- Additional data files: `SRC_AdaptivePlatformDemonstratorTestdata.zip`
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, or any of the other sample applications contained within the release archive.
- Explanatory material that details and documents the software design of the included functional clusters, see table 9.1.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
924	AUTOSAR_EXP_FCDesignPlatformHealthManagement
904	AUTOSAR_EXP_FCDesignREST
882	AUTOSAR_EXP_FCDesignSecurityManagement
905	AUTOSAR_EXP_FCDesignTimeSynchronization
894	AUTOSAR_EXP_FCDesignUpdateAndConfigManagement

Table 9.1: FCDesign documents contained in this release

For this release, the Demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

9.2 Known issues

- Some of the `ara::core` types' implementations violate their specifications because they allow implicit conversion from and/or to the `std::` types on which they are modeled. For instance, `ara::core::Vector` is currently implemented as a type alias for `std::vector` and can therefore be used interchangeably with it – this is not standardized, though, and shall not be assumed by client code.
- The class `ara::core::StringView` is only `constexpr`-capable when compiled with a suitable compiler (such as e.g. g++ 7.3.0) in C++17 mode. Most components in this release are configured to be built in C++11 or C++14 mode and are thus unable to use `ara::core::StringView` in constant expressions.
- The asynchronous stackable continuations feature of `ara::core::Future` (using `Future::then`) is implemented in a way that makes it neither asynchronous, nor a continuation, nor stackable.
- The `ara::core` types that are to replace `std::` types (such as `String`, `Vector`, `Map`, and `Array`) are not yet used throughout the whole codebase; some components are still using the corresponding `std::` types.
- There is no implementation of `ara::core::Variant`.
- There is no implementation of `ara::core::InstanceSpecifier`.
- The global initialization and shutdown functions `ara::core::Initialize` and `ara::core::Deinitialize` are not implemented.
- Some other Functional Clusters' implementations do not accurately reflect the state of their respective specifications. Please look into their FCDesign documents for a list of known issues.
- The sample scenario consisting of the two applications `radar` and `fusion` given in `sample-applications/cm_provider_subscriber_scenario` does not always have the two applications successfully establish a connection with each other when they are located on different platform instances. It can be made to work when subsequently using a `ssh` connection to restart them manually (e.g. by running `run/run_app.sh`).
It usually works well when both applications reside on the same platform instance (i.e., they are executed on the same machine and communicate via IPC).
- The `execution-manager` application, which acts as `/sbin/init` on the bootable images, uses excessive amounts of CPU time (up to 100%), due to an unrestricted busy loop. The same issue occurs with `access-manager`.
- The DoIP code uses a wrong length argument when sending data via UDP. This may lead to inability to receive response messages.
- The default log level assumed by `ara::log::CreateLogger` is currently `kVerbose`. This violates `SWS_LogAndTrace` which specifies it to be `kWarn`.

- The SystemTest application wrongly marks the test item STS_E2E_00001 as "passed", even though it should always fail due to missing implementations in the E2E subsystem. This is due to the fact that SystemTest cannot determine whether E2E is in fact available. It hence assumes that E2E is available and treats any successful data transmission as proof that data can be sent with E2E protection.
- SystemTest's CM test case #3 and #4 cannot be run in direct succession, as the ECU needs to reboot between these two test cases.
- The SystemTest application reports a number of test failures.

10 R18-10-1

This is a maintenance release based on R18-10; it fixes a critical issue that severely impairs the functioning of the Demonstrator in certain situations.

Apart from the changes mentioned below, this release behaves identically to R18-10 on which it is based.

10.1 Implemented changes

Changes made include the following:

- The implementation of the SOME/IP binding for `ara::com` contained a flaw in the field validation code. This validation code incorrectly called the validation method for `get` methods also for `set` methods, thereby rejecting all `set` method calls on a `field` because of a mismatching method identifier.

For `fields` that have a `set` method, but no `get` method, this issue resulted in a build error, because the validation method contains references to a symbol that does not exist in this case.

This issue has been fixed by adding a separate validation method for `set` methods and calling it appropriately.

10.2 Known issues

In addition to the known issues given for release R18-10, the following issue has been discovered:

- The ZIP archive that contains the source code cannot be fully uncompressed onto file systems that are case insensitive (such as NTFS), because `ara::com`'s DDS binding library contains a directory with a number of files whose filenames only differ in casing.

As the DDS binding is unused in this release, you can simply skip decompression of all affected files.

11 R18-10

11.1 Overview

This chapter provides an overview of the “AUTOSAR Adaptive Platform Demonstrator” provisional release for R18-10. This release is called “provisional” due to the fact that some quality measures on the source code have not yet concluded. AUTOSAR will perform these measures and attest the “provisional” release at a later point in time. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R18-10, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code of the Demonstrator: `SRC_AdaptivePlatformDemonstrator.zip`
- Additional data files: `SRC_AdaptivePlatformDemonstratorTestdata.zip`
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, and most sample applications.
- Explanatory material that details and documents the software design of the included functional clusters, see table 11.1.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
924	AUTOSAR_EXP_FCDesignPlatformHealthManagement
904	AUTOSAR_EXP_FCDesignREST
882	AUTOSAR_EXP_FCDesignSecurityManagement
905	AUTOSAR_EXP_FCDesignTimeSynchronization
894	AUTOSAR_EXP_FCDesignUpdateAndConfigManagement

Table 11.1: FCDesign documents contained in R18-10

For the release R18-10, the Demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

11.2 Known issues

- Some of the `ara::core` types' implementations violate their specifications because they allow implicit conversion from and to the `std::` types on which they are based. For instance, `ara::core::Vector` is currently implemented as a type alias to `std::vector` and can therefore be used interchangeably with it – this is not standardized, though, and shall not be assumed.
- Some of the `ara::core` types are not conforming to their specifications in other ways as well. For instance, the implementation of `ara::core::Result` has an lvalue ref-qualifier overload of the member function `Value()` that does not exist in the SWS, and it does not properly restrict the template function overload of `FromValue` from being applied to “T” types that are specializations of `Result`.
- The sample scenario consisting of the two applications `radar` and `fusion` given in `sample-applications/cm_provider_subscriber_scenario` does not always have the two applications successfully establish a connection with each other when they are located on different platform instances. It can be made to work when subsequently using a `ssh` connection to restart them manually (e.g. by running `run/run_app.sh`).
It usually works well when both applications reside on the same platform instance (i.e., they are executed on the same machine and communicate via IPC).
- The `SystemTest` application wrongly marks the test item `STS_E2E_00001` as “passed”, even though it should always fail due to missing implementations in the E2E subsystem. This is due to the fact that `SystemTest` cannot determine whether E2E is in fact available. It hence assumes that E2E is available and treats any successful data transmission as proof that data can be sent with E2E protection.

12 R18-03

12.1 Overview

This chapter provides an overview of the “AUTOSAR Adaptive Platform Demonstrator” provisional release for R18-03. This release is called “provisional” due to the fact that some quality measures on the source code have not yet concluded. AUTOSAR will perform these measures and attest the “provisional” release at a later point in time. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R18-03, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code of the demonstrator: `SRC_AdaptivePlatformDemonstrator.zip`
- Additional data files: `SRC_AdaptivePlatformDemonstratorTestdata.zip`
This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, and most sample applications.
- Explanatory material that details and documents the software design of the included functional clusters, see table 12.1.

Document ID	Document Name
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics
843	AUTOSAR_EXP_FCDesignExecutionManagement
901	AUTOSAR_EXP_FCDesignIdentityAndAccessManagement
861	AUTOSAR_EXP_FCDesignLogAndTrace
859	AUTOSAR_EXP_FCDesignPersistency
904	AUTOSAR_EXP_FCDesignREST
882	AUTOSAR_EXP_FCDesignSecurityManagement
905	AUTOSAR_EXP_FCDesignTimeSync
894	AUTOSAR_EXP_FCDesignUpdateAndConfigManagement

Table 12.1: FCDesign documents contained in R18-03

For the release R18-03, the demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

12.2 Known issues

No issues have been discovered so far.

13 R17-10

13.1 Overview

This chapter provides an overview of the “AUTOSAR Adaptive Platform Demonstrator” provisional release for R17-10. The release is called “provisional” due to the fact that some quality measures on the source code have not yet concluded. AUTOSAR will perform these measures and attest the “provisional” release at a later point in time. The Adaptive Platform Demonstrator is delivered as exemplary software to be used “as is”, so whoever builds on the source code needs to cover all aspects of a product development on his own.

The Demonstrator is part of the release R17-10, so the reference information is included in the respective Release Overview [1]

The release comprises:

- Source code of the demonstrator: `APD_R17-10.zip`
- Additional data files: `APD_R17-10_testdata.zip` This file contains optional input data that is only needed for executing the System Test application. It is not required for building and running the Demonstrator itself, its unit tests, and most sample applications.
- Explanatory material that details and documents the software design of the included functional clusters, see table 13.1.

Document ID	Document Name
891	AUTOSAR_EXP_FCDesignAdaptiveCryptoInterface
842	AUTOSAR_EXP_FCDesignCommunicationManagement
844	AUTOSAR_EXP_FCDesignDiagnostics.pdf
843	AUTOSAR_EXP_FCDesignExecutionManagement.pdf
861	AUTOSAR_EXP_FCDesignLogAndTrace.pdf
859	AUTOSAR_EXP_FCDesignPersistency.pdf
882	AUTOSAR_EXP_FCDesignSecurityManagement.pdf

Table 13.1: FCDesign documents contained in R17-10

For the release R17-10, the demonstrator software implementation has undergone some reviews, but a strict quality assurance is still under development. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the demonstrator software implementation.

This software is an exemplary implementation for validation purposes and is not optimized e.g. w.r.t. start-up time.

13.2 Known issues

The Yocto build of the SDK currently generates erroneous absolute path names into a certain CMake file, which will lead to compilation errors when trying to include `ara::com` into your project. To correct that, this CMake file must be manually corrected to only contain leaf filenames. Locate the file `<sdk>/sysroots/i586-poky-linux/usr/lib/cmake/vsomeip/vsomeipTargets.cmake` and find the call to `set_target_properties()`. After correction, this call should look like this:

```
set_target_properties(vsomeip PROPERTIES INTERFACE_LINK_LIBRARIES
    "/usr/lib/libmbedtls.so;
    ____/usr/lib/libmbedx509.so;
    ____/usr/lib/libmbedcrypto.so"
)
```

A Appendix

A.1 Licenses

A.1.1 Usage of W3C XML schema

The AUTOSAR XML Schema requires the XML namespace definition file `xml.xsd`.

There are several occurrences of the “`xml.xsd`” file within this release. For all occurrences the W3C license applies which can be found on <https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document>.

License

By obtaining and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this work, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the work or portions thereof, including modifications:

The full text of this NOTICE in a location viewable to users of the redistributed or derivative work. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software and Document Short Notice should be included. Notice of any changes or modifications, through a copyright statement on the new code or document such as “This software or document includes material copied from or derived from [title and URI of the W3C document]. Copyright (C) [YEAR] W3C^(R) (MIT, ERCIM, Keio, Beihang).”

Disclaimers

THIS WORK IS PROVIDED “AS IS,” AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENT WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENT.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the work without specific, written prior permission. Title to copyright in this work will at all times remain with copyright holders.