image**东软睿驰秘密(Neusoft Reach Secret)**

文件编号(File No.)：项目编号- D00-COP02-SW-T31-流水号

NeuSAR aCore

软件需求规格说明书

(Software requirement specification)

**(EM)**

版本(Version)：0.7

日期(Date)：2022-04-01

|  |  |  |
| --- | --- | --- |
| 编制  Prepared by | 审核  Reviewed by | 批准  Approved by |
|  |  |  |

东软睿驰汽车技术(沈阳)有限公司

**(版权所有，翻版必究)**

Neusoft Reach Automotive Technology(Shenyang) Co., Ltd

**(Copyright by Reach Corporation, All Rights Reserved)**

变更履历(Change Log)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **修改编号**  **No.** | **版本**  **Version** | **修改内容**  **Contents Revised** | **状态**  **Status** | **修改人/日期**  **Reviser/Date** | **审批人/日期**  **Approve/Date** |
| 1 | 0.7 | 新建 | Draft | 董名  2022.3.28 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Contents**

[1 引言(Introduction) 1](#_Toc99810348)

[1.1 目的(Goal) 1](#_Toc99810349)

[1.2 范围(Scope) 1](#_Toc99810350)

[1.3 参考文档(Reference) 1](#_Toc99810351)

[1.4 术语和缩略语(Terms And Abbreviations) 1](#_Toc99810352)

[2 软件系统概述(Software System Overview) 1](#_Toc99810353)

[2.1 软件系统背景(Software System Background) 1](#_Toc99810354)

[2.2 软件系统目标(Software System Goal) 2](#_Toc99810355)

[2.3 外部关联 (External Association) 2](#_Toc99810356)

[3 功能需求(Functional Requirement) 2](#_Toc99810357)

[3.1 执行管理General Requirement 2](#_Toc99810358)

[3.1.1 [SWRD\_EM\_00001] EM功能描述 2](#_Toc99810359)

[3.1.2 [SWRD\_EM\_00002] EM限制进程能力 2](#_Toc99810360)

[3.1.3 [SWRD\_EM\_00003] EM获取错误信息 3](#_Toc99810361)

[3.2 进程生命周期管理 3](#_Toc99810362)

[3.2.1 [SWRD\_EM\_00004]进程自终止行为 3](#_Toc99810363)

[3.2.2 [SWRD\_EM\_00004]process终止 3](#_Toc99810364)

[3.2.3 [SWRD\_EM\_00005]进程报告状态 4](#_Toc99810365)

[3.2.4 [SWRD\_EM\_00006]进程启动参数 4](#_Toc99810366)

[3.3 功能组状态管理 5](#_Toc99810367)

[3.3.1 [SWRD\_EM\_00007]MachineFG-General 5](#_Toc99810368)

[3.3.2 [SWRD\_EM\_00008]MachineFG初始状态 6](#_Toc99810369)

[3.3.3 [SWRD\_EM\_00009]MachineFG状态 6](#_Toc99810370)

[3.3.4 [SWRD\_EM\_00010]MachineFG状态 7](#_Toc99810371)

[3.3.5 [SWRD\_EM\_00011]MachineFG状态 7](#_Toc99810372)

[3.3.9 [SWRD\_EM\_00012]进程启动 8](#_Toc99810373)

[3.3.7 [SWRD\_EM\_00013]进程启动 8](#_Toc99810374)

[3.3.8 [SWRD\_EM\_00014]进程启动 9](#_Toc99810375)

[3.3.9 [SWRD\_EM\_00015]状态转换 9](#_Toc99810376)

[3.3.10 [SWRD\_EM\_00016] Unexpected Termination of starting processes 9](#_Toc99810377)

[3.3.11 [SWRD\_EM\_00017] Unexpected Termination of terminating processes 10](#_Toc99810378)

[3.4 Deterministic Execution 10](#_Toc99810379)

[3.4.1 [SWRD] 10](#_Toc99810380)

[3.5 Resource Limitation 11](#_Toc99810381)

[3.5.1 [ SWRD\_EM\_00018] Core Affinity 11](#_Toc99810382)

[3.5.2 [ SWRD\_EM\_00019] Memory Budget and Monitoring 11](#_Toc99810383)

[3.5.3 [ SWRD\_EM\_00020] Fault Tolerance 12](#_Toc99810384)

[3.6 Trusted Platform 12](#_Toc99810385)

[3.6.1 [ SWRD\_EM\_00021] Fault Tolerance 12](#_Toc99810386)

[3.6.2 [ SWRD\_EM\_00022] Fault Tolerance 14](#_Toc99810387)

[4 非功能需求(Non-Functional Requirements) 14](#_Toc99810388)

[4.1 制约(Constraint) 14](#_Toc99810389)

[[SWRD-ID]非功能需求1(Non-Function Requirement No.1) 14](#_Toc99810390)

[[SWRD-ID]非功能需求2(Non-Function Requirement No.2) 15](#_Toc99810391)

[4.2 性能质量要求(Performance Quality Requirements) 15](#_Toc99810392)

[4.2.1 [ SWRD\_EM\_00023] EM性能——升级情况 15](#_Toc99810393)

[4.2.2 [ SWRD\_EM\_00024] EM性能——切换状态情况 15](#_Toc99810394)

[4.2.3 [ SWRD\_EM\_00025] EM质量 16](#_Toc99810395)

[5 接口说明(API) 16](#_Toc99810396)

[5.1 接口头文件(API Header files) 16](#_Toc99810397)

[5.2 接口共同数据类型(API Common Data Types) 16](#_Toc99810398)

[5.2.1 [ SWRD\_API\_EM\_00001] ExecutionState 16](#_Toc99810399)

[5.2.2 [ SWRD\_API\_EM\_00002] ActivationReturnType 17](#_Toc99810400)

[5.2.3 [ SWRD\_API\_EM\_00003] DeterministicClient::TimeStamp 18](#_Toc99810401)

[5.2.4 [ SWRD\_API\_EM\_00004] ExecutionError 18](#_Toc99810402)

[5.2.5 [ SWRD\_API\_EM\_00005] ExecutionErrorEvent 19](#_Toc99810403)

[5.2.6 [ SWRD\_API\_EM\_00006] ExecutionErrorEvent::executionError 19](#_Toc99810404)

[5.2.7 [ SWRD\_API\_EM\_00007] ExecutionErrorEvent::functionGroup 20](#_Toc99810405)

[5.3 接口定义(API Reference) 20](#_Toc99810406)

[5.3.1 [ SWRD\_API\_EM\_00008] ExecutionClient class 20](#_Toc99810407)

[5.3.1.1 [ SWRD\_API\_EM\_00009] ExecutionClient::ExecutionClient 21](#_Toc99810408)

[5.3.1.2 [ SWRD\_API\_EM\_00010] ExecutionClient::~ExecutionClient 21](#_Toc99810409)

[5.3.1.3 [ SWRD\_API\_EM\_00011] ExecutionClient::ReportExecutionState 22](#_Toc99810410)

[5.3.2 [ SWRD\_API\_EM\_00012] WorkerRunnable class 23](#_Toc99810411)

[5.3.2.1 [ SWRD\_API\_EM\_00013] WorkerRunnable::Run 23](#_Toc99810412)

[5.3.3 [ SWRD\_API\_EM\_00014] WorkerThread class 24](#_Toc99810413)

[5.3.3.1 [ SWRD\_API\_EM\_00015] WorkerThread::WorkerThread 24](#_Toc99810414)

[5.3.3.2 [ SWRD\_API\_EM\_00016] WorkerThread::~WorkerThread 25](#_Toc99810415)

[5.3.3.3 [ SWRD\_API\_EM\_00017] WorkerThread::GetRandom 25](#_Toc99810416)

[5.3.4 [ SWRD\_API\_EM\_00018] DeterministicClient class 26](#_Toc99810417)

[5.3.4.1 [ SWRD\_API\_EM\_00019] DeterministicClient::DeterministicClient 26](#_Toc99810418)

[5.3.4.2 [ SWRD\_API\_EM\_00020] DeterministicClient::~DeterministicClient 27](#_Toc99810419)

[5.3.4.3 [ SWRD\_API\_EM\_00021] DeterministicClient::WaitForActivation 27](#_Toc99810420)

[5.3.4.4 [ SWRD\_API\_EM\_00022] DeterministicClient::RunWorkerPool 28](#_Toc99810421)

[5.3.4.5 [ SWRD\_API\_EM\_00023] DeterministicClient::GetRandom 29](#_Toc99810422)

[5.3.4.6 [ SWRD\_API\_EM\_00024] DeterministicClient::SetRandomSeed 30](#_Toc99810423)

[5.3.4.7 [ SWRD\_API\_EM\_00025] DeterministicClient::GetActivationTime 30](#_Toc99810424)

[5.3.4.8 [ SWRD\_API\_EM\_00026] DeterministicClient::GetNextActivationTime 31](#_Toc99810425)

[5.3.5 [ SWRD\_API\_EM\_00027] FunctionGroup class 32](#_Toc99810426)

[5.3.5.1 [ SWRD\_API\_EM\_00028] FunctionGroup::Create 32](#_Toc99810427)

[5.3.5.2 [ SWRD\_API\_EM\_00029] FunctionGroup::FunctionGroup 33](#_Toc99810428)

[5.3.5.3 [ SWRD\_API\_EM\_00030] FunctionGroup::FunctionGroup (Copy Constructor) 34](#_Toc99810429)

[5.3.5.4 [ SWRD\_API\_EM\_00031] FunctionGroup::FunctionGroup (Move Constructor) 34](#_Toc99810430)

[5.3.5.5 [ SWRD\_API\_EM\_00032] FunctionGroup:: operator= (Copy assignment operator) 35](#_Toc99810431)

[5.3.5.6 [ SWRD\_API\_EM\_00033] FunctionGroup::operator= (Move assignment operator) 36](#_Toc99810432)

[5.3.5.7 [ SWRD\_API\_EM\_00034] FunctionGroup::~FunctionGroup 36](#_Toc99810433)

[5.3.5.8 [ SWRD\_API\_EM\_00035] FunctionGroup::operator== 37](#_Toc99810434)

[5.3.5.9 [ SWRD\_API\_EM\_00036] FunctionGroup::operator!= 37](#_Toc99810435)

[5.3.6 [ SWRD\_API\_EM\_00037] FunctionGroupState class 38](#_Toc99810436)

[5.3.6.1 [ SWRD\_API\_EM\_00038] FunctionGroupState::Create 39](#_Toc99810437)

[5.3.6.2 [ SWRD\_API\_EM\_00039] FunctionGroupState::FunctionGroupState 40](#_Toc99810438)

[5.3.6.3 [ SWRD\_API\_EM\_00040] FunctionGroupState::FunctionGroupState (Copy Constructor) 40](#_Toc99810439)

[5.3.6.4 [ SWRD\_API\_EM\_00041] FunctionGroupState::FunctionGroupState (Move Constructor) 41](#_Toc99810440)

[5.3.6.5 [ SWRD\_API\_EM\_00042] FunctionGroupState::operator= (Copy assignment operator) 41](#_Toc99810441)

[5.3.6.6 [ SWRD\_API\_EM\_00043] FunctionGroupState::operator= (Move assignment operator) 42](#_Toc99810442)

[5.3.6.7 [ SWRD\_API\_EM\_00044] FunctionGroupState::~FunctionGroupState 42](#_Toc99810443)

[5.3.6.8 [ SWRD\_API\_EM\_00045] FunctionGroupState::operator== 43](#_Toc99810444)

[5.3.6.9 [ SWRD\_API\_EM\_00046] FunctionGroupState::operator!= 44](#_Toc99810445)

[5.3.7 [ SWRD\_API\_EM\_00047] StateClient class 44](#_Toc99810446)

[5.3.7.1 [ SWRD\_API\_EM\_00048] StateClient::StateClient 45](#_Toc99810447)

[5.3.7.2 [ SWRD\_API\_EM\_00049] StateClient::SetState 46](#_Toc99810448)

[5.3.7.3 [ SWRD\_API\_EM\_00050] StateClient::GetInitialMachineStateTransitionResult 47](#_Toc99810449)

[5.3.7.4 [ SWRD\_API\_EM\_00051] StateClient::GetExecutionError 48](#_Toc99810450)

[5.3.8 [ SWRD\_API\_EM\_00052] Execution Management error codes 49](#_Toc99810451)

[5.3.9 [ SWRD\_API\_EM\_00053] ExecException type 51](#_Toc99810452)

[5.3.9.1 [ SWRD\_API\_EM\_00054] ExecException::ExecException 51](#_Toc99810453)

[5.3.10 [ SWRD\_API\_EM\_00055] GetExecErrorDomain function 52](#_Toc99810454)

[5.3.11 [ SWRD\_API\_EM\_00056] MakeErrorCode function 52](#_Toc99810455)

[5.3.12 [ SWRD\_API\_EM\_00057] ExecErrorDomain type 53](#_Toc99810456)

[5.3.12.1 [ SWRD\_API\_EM\_00058] ExecErrorDomain::ExecErrorDomain 53](#_Toc99810457)

[5.3.12.2 [ SWRD\_API\_EM\_00059] ExecErrorDomain::Name 54](#_Toc99810458)

[5.3.12.3 [ SWRD\_API\_EM\_00060] ExecErrorDomain::Message 54](#_Toc99810459)

[5.3.12.4 [ SWRD\_API\_EM\_00061] ExecErrorDomain::ThrowAsException 55](#_Toc99810460)

[附录A- 信息定义 56](#_Toc99810461)

[附录B- 配置信息 58](#_Toc99810462)

# 引言(Introduction)

## 目的(Goal)

编写本文的目的，是为了EM模块提供详细的软件需求的定义，给开发人员和测试人员提供设计和测试执行的标准。

## 范围(Scope)

本文使用者： 开发人员、测试人员、PSM、TeamLeader、TestLeader和产品负责人。

本文使用方法：

* 对于开发人员、根据本文中定义的功能/非功能要求进行后续的设计。
* 对于测试人员、通过理解本文中的要求，进行测试用例的制作和后续测试执行。
* 对于PSM、TeamLeader、TestLeader和产品负责人、来判断需求理解的正确性。

## 参考文档(Reference)

|  |  |  |
| --- | --- | --- |
| **编号** | **SVN路径\文档名** | **文档版本** |
| 1 | 《AUTOSAR\_SWS\_ExecutionManagement》 | R-2111 |
| 2 | 《AUTOSAR\_TPS\_ManifestSpecification》 | R-2111 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 术语和缩略语(Terms And Abbreviations)

|  |  |  |
| --- | --- | --- |
| **编号** | **缩写** | **说明** |
| 1 | EM | Execution Managment执行管理 |
| 2 | SM | State Management，状态管理 |
| 3 | Execution Manifest | 执行清单 |
| 4 | Machine Manifest. | 机器清单 |
| 5 | Process | Process是Executable的加载实例，是Application的一部分。 |

# 软件系统概述(Software System Overview)

## 软件系统背景(Software System Background)

执行管理模块是AUTOSAR Adaptive Platform的functional cluster，负责平台的初始化和Adaptive Applications的启动和终止。

## 软件系统目标(Software System Goal)

执行管理模块开发的目标是根据AUTOSAR规范，完成进程生命周期管理、状态管理需求、确定性执行、资源限制、容错和安全相关以及错误处理的相关需求。

## 外部关联 (External Association)

根据规范定义，EM需要与SM模块通过API完成状态管理相关需求。

# 功能需求(Functional Requirement)

EM作为AUTOSAR Adaptive Platform启动阶段的一部分，负责启动和终止进程。EM根据机器清单和执行清单中的信息确定何时以及可能以何种顺序启动或者停止processes。

EM确保检查所有Executables和Executables相关的数据（例如清单）的完整性和真实性。在完整性或真实性检查失败的情况下，EM将执行Security需求中定义的措施。

EM不负责processes的运行时调度，因为这是操作系统的职责，但是EM负责操作系统的初始化和配置，以使操作系统能够根据执行管理从“机器清单”和“执行清单”中提取的信息执行必要的运行时调度和资源管理。

EM不负责标准化的终止处理，如收到SIGTERM信号后的处理。

## 执行管理General Requirement

### [SWRD\_EM\_00001] EM功能描述

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00001 |
| **Type** | TBD |
| **Description** | WorkerThread 类提供了定义 DeterministicClient 工作线程的预期接口的类，特别是对确定性随机数的访问。 |
| **Upstream ID** | [SWS\_EM\_02530] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | TBD |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | L |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00002] EM限制进程能力

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00002 |
| **Type** | Valid |
| **Description** | 进程的进程创建权限限制。执行管理应限制进程的权限，使其不能启动其他进程。 |
| **Upstream ID** | [SWS\_EM\_01030] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 被EM启动的进程，无法再fork其子进程 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | M |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_EM\_00003] EM获取错误信息

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00003 |
| **Type** | TBD |
| **Description** | 执行管理应提供一种获取 API 调用期间错误信息的方法。  执行管理应提供创建错误信息的方法。 |
| **Upstream ID** | [SWS\_EM\_02547] [SWS\_EM\_02548][2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | TBD |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | L |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

## 进程生命周期管理

### [SWRD\_EM\_00004]进程自终止行为

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00004 |
| **Type** | Valid |
| **Description** | 执行管理的默认值应将没有指定终止行为的建模流程视为仅在执行管理请求时终止的流程。 |
| **Upstream ID** | [SWS\_EM\_01314][2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 配置为不可自终止的进程只能由EM发出终止信号，其才能终止。如果该进程自己停止了，EM会日志输出相关错误。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00004]process终止

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00004 |
| **Type** | Valid |
| **Description** | 执行管理应通过将SIGTERM信号发送给进程来引发Process终止。收到SIGTERM后，进程便开始实际的终止。  在Terminating状态期间，process应该保存持久化数据，并且释放所有内部使用资源。  在意外终止的情况下，执行管理应执行以下操作：  1. 如需要，记录日志；  2. 把当前的Function Group State设置为Undefined Function Group State；  3. 调用ara::exec::StateClient定义的undefinedStateCallback。  4. 通过ara::exec::StateClient::GetExecutionError 接口报告配置的executionError。  （3实现的优先级较低） |
| **Upstream ID** | [SWS\_EM\_01055][SWS\_EM\_01309][2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 1. EM记录相关日志 2. 将有进程意外退出的功能组设置为Undefined状态 3. 调用ara::exec::StateClient定义的undefinedStateCallback（优先级低） 4. SM可通过ara::exec::StateClient::GetExecutionError 接口获得配置的executionError |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00005]进程报告状态

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00005 |
| **Type** | Valid |
| **Description** | 进程需要通过ara::exec::ExecutionClient::ReportExecutionState接口报告kRunning状态。  Execution Management 应在进程报告 Execution State kRunning（使用方法 ara::exec::ExecutionClient::ReportExecutionState）且进程未处于 Process State Starting 时返回 kInvalidTransition。 |
| **Upstream ID** | [SWS\_EM\_02243] [2111M] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | EM在进程报告krunning时增加处理，新增返回值kInvalidTransition。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

### [SWRD\_EM\_00006]进程启动参数

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00006 |
| **Type** | Valid |
| **Description** | Process Argument.argument 应与第一个 Process Argument 一起按顺序传递给进程。 从过程参数 1 开始的参数。 |
| **Upstream ID** | [SWS\_EM\_01078] [2111M] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 主要测试点：确认可以按照配置的启动参数启动相应的进程。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | M |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

## 功能组状态管理

### [SWRD\_EM\_00007]MachineFG-General

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00007 |
| **Type** | Valid |
| **Description** | 执行管理应从具有类别 PLATFORM\_CORE 的 SoftwareCluster 中的功能组“MachineFG”获取机器状态的配置。  从初始状态 Startup 到状态管理 SM 请求初始运行机器状态 StateXYZ 的启动顺序如图 7.8 所示。    机器状态 StateXYZ 的任意状态变化序列如图 7.9 所示。 这里，在收到状态更改请求后，执行管理终止正在运行的进程，然后在向状态管理确认状态更改之前启动在新状态下活动的进程。 |
| **Upstream ID** | [SWS\_EM\_01032] [2111M] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 2111版本中，MachineState更名为MachineFG，对于上述EM与SM的交互过程，1911已实现的函数中，除MachineState更名为MachineFG外，其余部分测试方法与1911相同。对于EM和SM新增交互，以其单独需求分析为准。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

### [SWRD\_EM\_00008]MachineFG初始状态

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00008 |
| **Type** | Valid |
| **Description** | 如果未为功能组“MachineFG”配置启动（Startup）状态，执行管理应停止 AUTOSAR 自适应平台启动。 |
| **Upstream ID** | [SWS\_EM\_02250] [2111M] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 如果MachineFG没有Startup状态，EM将不启动平台进程。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

### [SWRD\_EM\_00009]MachineFG状态

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00009 |
| **Type** | TBD |
| **Description** | 机器状态启动完成。当初始（自启动）机器状态转换到启动状态结束时，执行管理应通知状态管理初始转换的结果（使用 ara::exec::StateClient::GetInitialMachineStateTransitionResult） |
| **Upstream ID** | [SWS\_EM\_02241] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | MachineFG的Startup状态完成后，EM通过GetInitialMachineStateTransitionResult接口通知SM |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | M |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00010]MachineFG状态

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00010 |
| **Type** | TBD |
| **Description** | 执行管理应拒绝将“MachineFG”功能组状态更改为关闭的请求，并出现错误 kInvalidTransition。 |
| **Upstream ID** | [SWS\_EM\_02549] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | EM拒绝将MachineFG切换为Off状态的请求，并返回kInvalidTransition。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00011]MachineFG状态

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00011 |
| **Type** | TBD |
| **Description** | 错误配置的进程 - 分配给多个功能组  在功能组状态转换期间，涉及的任何进程引用来自多个功能组的状态，EM 执行以下操作：  1. 停止功能组状态转换，以便状态管理可以决定如何进行。  2. 根据需要记录事件  3. 将当前功能组状态设置为未定义功能组状态。  4. ara::exec::StateClient::SetState接口报告kFailed，表示State change请求无法完成。  5. 通过 ara::exec::StateClient::GetExecutionError 接口报告为请求的功能组状态配置的 executionError。 |
| **Upstream ID** | [SWS\_EM\_02254] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 进程不能配置在不同的功能组，此处1911版本的上位机已经做了相应校验，该需求的场景不存在。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

### [SWRD\_EM\_00012]进程启动

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00012 |
| **Type** | TBD |
| **Description** | 错误配置的进程 - 分配给多个功能组  在功能组状态转换期间，涉及的任何进程引用来自多个功能组的状态，EM 执行以下操作：  1. 停止功能组状态转换，以便状态管理可以决定如何进行。  2. 根据需要记录事件  3. 将当前功能组状态设置为未定义功能组状态。  4. ara::exec::StateClient::SetState接口报告kFailed，表示State change请求无法完成。  5. 通过 ara::exec::StateClient::GetExecutionError 接口报告为请求的功能组状态配置的executionError。 |
| **Upstream ID** | [SWS\_EM\_02254] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 进程不能配置在不同的功能组，此处1911版本的上位机已经做了相应校验，该需求的场景不存在。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

### [SWRD\_EM\_00013]进程启动

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00013 |
| **Type** | TBD |
| **Description** | 每个功能组（包括功能组“MachineFG”）都有一个关闭状态，如果没有请求其他状态，则执行管理应将其用作初始功能组状态。  MachineFG是否需要Off状态 |
| **Upstream ID** | [SWS\_EM\_01110] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 主要测试点：确认每个功能组都包含"Off"状态。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00014]进程启动

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00014 |
| **Type** | TBD |
| **Description** | 根据 [SWS\_EM\_02260] 的重新启动尝试不应满足任何终止的依赖关系。 |
| **Upstream ID** | [SWS\_EM\_02280] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | - |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | L |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00015]状态转换

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00015 |
| **Type** | Valid |
| **Description** | **State transition** -如果在执行管理尝试重新启动进程numberOfRestartAttempts 次后发生进程启动超时，执行管理应请求操作系统终止底层进程。  **State transition** -进程启动超时报告 当进程启动导致超时时，执行管理应执行以下操作：  1. 停止功能组状态转换，以便状态管理可以决定如何进行  2. 记录事件  3. 将 CurrentState 设置为未定义的功能组状态  4. ara::exec::StateClient::SetState接口报告kFailed，表示State change请求无法完成  5. 通过 ara::exec::StateClient::GetExecutionError 接口报告配置的 executionError  执行管理应在向状态管理部门报告前执行终止反应。 |
| **Upstream ID** | [SWS\_EM\_02310] [SWS\_EM\_02259][ SWS\_EM\_02312][2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 1. 当进程启动配置启动次数后，EM负责kill该进程 2. 进程启动超时时，EM按规范要求流程处理 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00016] Unexpected Termination of starting processes

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00016 |
| **Type** | Valid |
| **Description** | **Unexpected Termination of starting processes during Function Group State transition**：一旦process启动期间发生未预期的关闭，则EM应执行以下动作：   1. 停止功能组状态转换，SM将决定如何实施动作； 2. 如需要，记录日志； 3. 设置当前Function Group State为Undefined Function Group State； 4. 通过ara::exec::StateClient::SetState接口报告；kFailedUnexpectedTerminationOnEnter，意味着状态转换请求不能被满足；   通过ara::exec::StateClient::GetExecutionError接口报告配置的executionError。 |
| **Upstream ID** | [SWS\_EM\_02313] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 确认process启动期间发生未预期的关闭时按照需求描述的动作处理。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_EM\_00017] Unexpected Termination of terminating processes

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00017 |
| **Type** | Valid |
| **Description** | 在功能组状态转换期间终止进程的意外终止，EM应记录日志 |
| **Upstream ID** | [SWS\_EM\_02314] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 确认process终止期间发生未预期的关闭时按照需求描述的动作处理。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

## Deterministic Execution

该部分优先级靠后

### [SWRD]

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD |
| **Type** |  |
| **Description** |  |
| **Upstream ID** |  |
| **Dependencies** |  |
| **Verification method** |  |
| **Verification Criteria** |  |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

## Resource Limitation

执行管理对平台内进程使用的资源进行限制。

### [ SWRD\_EM\_00018] Core Affinity

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00018 |
| **Type** | Valid |
| **Description** | [SWS\_EM\_02104]允许将初始线程（进程的“主”线程）绑定到某些cores。 根据操作系统的功能，子集可以是single core。 如果操作系统不支持绑定到特定cores，则唯一支持的子集是整个cores。  执行管理应根据配置 ProcessToMachineMapping.shallRunOn 和 ProcessToMachineMapping.shallNotRunOn 配置进程初始线程的核心亲和性（将其限制为系统中的核心子集）。 |
| **Upstream ID** | [SWS\_EM\_02104][2111M] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 确认可以按照需求描述进行Core关联配置 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

### [ SWRD\_EM\_00019] Memory Budget and Monitoring

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00019 |
| **Type** | Valid |
| **Description** | Memory Budget and Monitoring：  Maximum heap：EM应配置进程的最大内存使用量。  Maximum system memory usage：执行管理应配置进程的最大系统内存使用量。  process pre-mapping：如果相应的执行清单要求，执行管理应pre-map进程。 |
| **Upstream ID** | [SWS\_EM\_02107][ SWS\_EM\_02108][ SWS\_EM\_02109][2111M] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 确认可以按照需求描述进行内存的限制和监控。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | H |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

### [ SWRD\_EM\_00020] Fault Tolerance

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00020 |
| **Type** | Invlid[优先级低] |
| **Description** | EM对AUTOSAR Adaptive Platform的整体系统行为具有至关重要的影响。根据不同的操作模式和错误类型，EM中错误的严重性可能产生不同的影响。因此，不仅需要指定正确的行为，而且在出现偏差的情况下还必须引入其他行为。  Fault tolerance机制的实现基于两个连贯的步骤- Error Detection和subsequent Error Recovery。  在设计阶段中Error Detection为主要重点，因此容错的重点是对潜在故障模式的分析以及随后应纳入实现中的错误检测机制。  相反，Error Recovery包含应采取的措施，以恢复系统的状态，使系统可以再次执行正确的服务交付。Error Detection和Recovery Actions的结合应该成为平台范围的容错模型的主题。  当没有其他措施时，EM可以进入Unrecoverable状态，Unrecoverable状态仅由EM触发。  在进入Unrecoverable状态前，EM执行pre-cleanup action；  在pre-cleanup action后，所有被EM管理的processes都应关闭；  所有被EM管理的processes关闭后，post-cleanup action应被调用。 |
| **Upstream ID** | [SWS\_EM\_02032][ SWS\_EM\_02033] [2111M] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 确认可以按照需求描述进行内存的限制和监控。 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | L |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

## Trusted Platform

### [ SWRD\_EM\_00021] Fault Tolerance

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00021 |
| **Type** | Valid |
| **Description** | 从security角度来看，必须确保在Adaptive Platform上执行的所有软件都是受信任的，即确保软件的完整性和真实性。EM作为负责进程创建的实体-应接替这项任务。  有很多方法可以验证Adaptive Platform的完整性和真实性。 可以实现可信平台通过（但不限于）   * Bootloader验证完整的Ramdisk * 验证单个可执行文件和数据文件，例如使用OSfunctionalities或trusted third-party process * 加载后验证individual memory pages，例如使用OSfunctionalities或trusted third-party process   EM必须确保检查processed机器清单的完整性和真实性。  EM应确保对于将要启动的每个process，都要检查Executable本身的完整性和真实性。  EM确保对于将要启动的每个process，检查每个相关shared object的完整性和真实性。  EM应确保针对将要启动的每个process，检查其相应的processed执行清单的完整性和真实性。  执行管理应确保针对将要启动的每个process，检查其相应的processed服务实例清单的完整性和真实性。  EM应确保在执行之前，对Adaptive Platform上的所有Executable code行身份验证。完整的经过身份验证的启动顺序如下所示：    如果完整性和真实性已成功验证，则系统应继续其正常的启动过程。 但是，如果完整性和真实性检查失败，则EM应提供有关如何继续启动过程的配置选项。  EM应提供两种模式来处理失败的真实性检查：Monitoring Mode和 Strict Mode。两种模式的配置是通过Machine Manifest完成的。 只有在验证了“机器清单”的完整性和真实性之后，才能处理配置选项。  如果机器清单的完整性或真实性检查失败，则应停止Adaptive Platform的启动。  **Monitoring Mode**：在Monitoring Mode下，执行完整性和真实性检查，但启动过程不受影响。因此，即使文件系统遭到破坏，Adaptive Platform也会启动。  当集成商希望系统保持运行时，即使平台不被认为是受信任的，“Monitoring Mode”也很有用。在这种情况下，集成商可能会在Adaptive AUTOSAR范围之外使用其他措施，例如使用支持此功能的HSM时限制密钥访问。  **Strict Mode**：在此模式下，自适应平台确保在无法验证相应的可执行文件，清单或链接库的完整性和真实性的时候不执行任何进程。  在Strict Mode下，如果相应的processed执行清单的完整性或真实性检查失败，则EM不应启动可执行文件的执行。  在Strict Mode下，如果相应的processed服务实例清单的完整性或真实性检查失败，则EM不应启动可执行文件的执行。  在Strict Mode下，EM仅在其完整性和真实性检查成功的情况下才执行代码。Executable code可以由可执行文件提供，也可以由executable链接的共享对象提供。 |
| **Upstream ID** | [SWS\_EM\_02299][SWS\_EM\_02300][SWS\_EM\_02301][SWS\_EM\_02302]  [SWS\_EM\_02303][SWS\_EM\_02304][SWS\_EM\_02305][SWS\_EM\_02306]  [SWS\_EM\_02307][SWS\_EM\_02308][SWS\_EM\_02309][2111M] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | EM根据规范要求，执行可信任平台相关操作 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | M |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 修改 |

### [ SWRD\_EM\_00022] Fault Tolerance

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00022 |
| **Type** | TBD |
| **Description** | 分配给流程执行管理的 IAM 配置属性应在流程创建期间将建模流程身份与流程相关联。身份的形式是特定于实现的，但例如可以是进程标识符、加密令牌、用户 ID 等。  根据实施要求，执行管理可以公开允许 IAM 检索有关流程和建模流程身份之间关联的信息的接口。 这个接口的确切形式是实现定义的。 |
| **Upstream ID** | [SWS\_EM\_02400] [2111A] |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 具体测试方法需要与IAM协同 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | M |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

# 非功能需求(Non-Functional Requirements)

## 制约(Constraint)

[

1. 软件运行环境：考虑以下几个方面：

接口：与其他模块的接口；

环境：使用到其他模块数据类型，AutoSAR头文件的包含关系；

1. 法律、法规和标准：如：必须按照ISO26262标准进行开发。]

### [SWRD-ID]非功能需求1(Non-Function Requirement No.1)

|  |  |
| --- | --- |
| **SWRD-ID** |  |
| **Type** |  |
| **Description** |  |
| **Upstream ID** |  |
| **Dependencies** |  |
| **Verification method** |  |
| **Verification Criteria** |  |
| **ASIL** |  |
| **Status** |  |
| **Priority** |  |
| **CR** |  |
| **Risk** |  |
| **Change Type** |  |

### [SWRD-ID]非功能需求2(Non-Function Requirement No.2)

|  |  |
| --- | --- |
| **SWRD-ID** |  |
| **Type** |  |
| **Description** |  |
| **Upstream ID** |  |
| **Dependencies** |  |
| **Verification method** |  |
| **Verification Criteria** |  |
| **ASIL** |  |
| **Status** |  |
| **Priority** |  |
| **CR** |  |
| **Risk** |  |
| **Change Type** |  |

## 性能质量要求(Performance Quality Requirements)

### [ SWRD\_EM\_00023] EM性能——升级情况

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00023 |
| **Type** | Valid |
| **Description** | 反复升级的场景下EM切换功能组的性能。 |
| **Upstream ID** | NA |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 评估反复升级的场景下EM切换功能组的性能 |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | L |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [ SWRD\_EM\_00024] EM性能——切换状态情况

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00024 |
| **Type** | Valid |
| **Description** | 进程状态频繁变化时，与PHM接口的交互性能。 |
| **Upstream ID** | NA |
| **Dependencies** | - |
| **Verification method** | 集成测试 |
| **Verification Criteria** | 评估进程状态频繁变化时，与PHM接口的交互性能（此接口2111删除，具体情况参考PHM和SM是否要求修改此接口） |
| **ASIL** | QM |
| **Status** | Draft |
| **Priority** | L |
| **CR** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [ SWRD\_EM\_00025] EM质量

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_EM\_00025 |
| **Type** |  |
| **Description** |  |
| **Upstream ID** |  |
| **Dependencies** |  |
| **Verification method** |  |
| **Verification Criteria** |  |
| **ASIL** |  |
| **Status** |  |
| **Priority** |  |
| **CR** |  |
| **Risk** |  |
| **Change Type** |  |

# 接口说明(API)

## 接口头文件(API Header files)

NA

## 接口共同数据类型(API Common Data Types)

### [ SWRD\_API\_EM\_00001] ExecutionState

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00001 | |
| **Type** | Valid | |
| **Priority** | H | |
| **Upstream ID** | SWS\_EM\_02000 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| **Kind** | enumeration | |
| **Symbol** | ExecutionState | |
| **Scope** | namespace ara::exec | |
| **Underlying type** | uint8\_t | |
| **Syntax** | enum class ExecutionState : uint8\_t {...}; | |
| **Values** | kRunning= 0 | After a Process has been started by Execution Management, it reports ExecutionState kRunning. |
| **Header file** | #include "ara/exec/execution\_client.h" | |
| **Description** | Defines the internal states of a Process (see 7.3.1). Scoped Enumeration of uint8\_t . | |

### [ SWRD\_API\_EM\_00002] ActivationReturnType

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00002 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02201 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| **Kind** | enumeration | |
| **Symbol** | ActivationReturnType | |
| **Scope** | namespace ara::exec | |
| **Underlying type** | std::uint32\_t | |
| **Syntax** | enum class ActivationReturnType : std::uint32\_t {...}; | |
| **Values** | kRegisterServices= 0 | application shall register communication services(this must be the only occasion for performing service registering). |
| kServiceDiscovery= 1 | application shall do communication service discovery (this must be the only occasion for performing service discovery) |
| kInit= 2 | application shall initialize its internal data structures (once) |
| kRun= 3 | application shall perform its normal operation |
| kTerminate= 4 | deterministic execution shall terminate |
| **Header file** | #include "ara/exec/deterministic\_client.h" | |
| **Description** | Defines the return codes for WaitForNextActivation operations. Scoped Enumeration of uint8\_t . | |

### [ SWRD\_API\_EM\_00003] DeterministicClient::TimeStamp

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00003 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02203 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| **Kind** | type alias |
| **Symbol** | TimeStamp |
| **Scope** | class ara::exec::DeterministicClient |
| **Underlying type** | - |
| **Syntax** | using TimeStamp = std::chrono::time\_point<ara::core::SteadyClock>; |
| **Header file** | #include "ara/exec/deterministic\_client.h" |
| **Description** | Time stamp of deterministic cycles . |

### [ SWRD\_API\_EM\_00004] ExecutionError

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00004 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02541 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| **Kind** | type alias |
| **Symbol** | ExecutionError |
| **Scope** | namespace ara::exec |
| **Underlying type** | std::uint32\_t |
| **Syntax** | using ExecutionError = std::uint32\_t; |
| **Header file** | #include "ara/exec/execution\_error\_event.h" |
| **Description** | Represents the execution error. |

### [ SWRD\_API\_EM\_00005] ExecutionErrorEvent

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00005 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02544 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| **Kind** | struct |
| **Symbol** | ExecutionErrorEvent |
| **Scope** | namespace ara::exec |
| **Underlying type** | - |
| **Syntax** | struct ExecutionErrorEvent final {...}; |
| **Header file** | #include "ara/exec/execution\_error\_event.h" |
| **Description** | Represents an execution error event which happens in a Function Group. |

### [ SWRD\_API\_EM\_00006] ExecutionErrorEvent::executionError

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00006 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02545 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| **Kind** | variable |
| **Symbol** | executionError |
| **Scope** | struct ara::exec::ExecutionErrorEvent |
| **Underlying type** | ExecutionError |
| **Syntax** | ExecutionError executionError; |
| **Header file** | #include "ara/exec/execution\_error\_event.h" |
| **Description** | The execution error of the Process which unexpectedly terminated . |

### [ SWRD\_API\_EM\_00007] ExecutionErrorEvent::functionGroup

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00007 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02546 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| **Kind** | variable |
| **Symbol** | functionGroup |
| **Scope** | struct ara::exec::ExecutionErrorEvent |
| **Underlying type** | FunctionGroup |
| **Syntax** | FunctionGroup functionGroup; |
| **Header file** | #include "ara/exec/execution\_error\_event.h" |
| **Description** | The function group in which the error occurred . |

## 接口定义(API Reference)

### [ SWRD\_API\_EM\_00008] ExecutionClient class

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00008 |
| **Type** | Valid |
| **Priority** | H |
| **Upstream ID** | SWS\_EM\_02001 |
| **CR** |  |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | class |
| **Symbol** | ExecutionClient |
| **Scope** | namespace ara::exec |
| **Syntax** | class ExecutionClient final {...}; |
| **Header file** | #include "ara/exec/execution\_client.h" |
| **Description** | Constructor that creates the Execution Client . |
| **Notes:** | Constructor for ExecutionClient which opens the Execution Management communication channel (e.g. POSIX FIFO) for reporting the Execution State. Each Process shall create an instance of this class to report its state |

#### [ SWRD\_API\_EM\_00009] ExecutionClient::ExecutionClient

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00009 |
| **Type** | Valid |
| **Priority** | H |
| **Upstream ID** | SWS\_EM\_02510 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | function |
| **Symbol** | ExecutionClient() |
| **Scope** | class ara::exec::ExecutionClient |
| **Syntax** | ExecutionClient () noexcept; |
| **Header file** | #include "ara/exec/execution\_client.h" |
| **Description** | Constructor that creates the Execution Client . |
| **Additional** | Constructor for ExecutionClient which opens the Execution Management communication channel (e.g. POSIX FIFO) for reporting the Execution State. Each Process shall create an instance of this class to report its state |

#### [ SWRD\_API\_EM\_00010] ExecutionClient::~ExecutionClient

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00010 |
| **Type** | Valid |
| **Priority** | H |
| **Upstream ID** | SWS\_EM\_02002 |
| **CR** |  |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | function |
| **Symbol** | ~ExecutionClient() |
| **Scope** | namespace ara::exec |
| **Syntax** | class ara::exec::ExecutionClient |
| **Header file** | #include "ara/exec/execution\_client.h" |
| **Description** | #include "ara/exec/execution\_client.h" |
| **Description:** | Destructor of the Execution Client instance . |

#### [ SWRD\_API\_EM\_00011] ExecutionClient::ReportExecutionState

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00011 | |
| **Type** | Valid | |
| **Priority** | H | |
| **Upstream ID** | SWS\_EM\_02003 | |
| **CR** |  | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | ReportExecutionState(ExecutionState state) | |
| **Scope** | class ara::exec::ExecutionClient | |
| **Syntax** | ara::core::Result<void> ReportExecutionState (ExecutionState state) const noexcept; | |
| **Parameters (in)** | state | Value of the Execution State |
| **Return value:** | ara::core::Result< void > | An instance of ara::core::Result. The instance holds an ErrorCode containing either one of the specified errors or a void-value. |
| **Exception Safety:** | noexcept | |
| **Errors:** | ara::exec::ExecErrc::kGeneralError | if some unspecified error occurred |
| ara::exec::ExecErrc::kCommunication Error | Communication error between Application and Execution Management, e.g. unable to report state for Non-reporting Process. |
| ara::exec::ExecErrc::kInvalidTransition | Invalid transition request (e.g. to Running when already in Running state) |
| **Header file** | #include "ara/exec/execution\_client.h" | |
| **Description** | Interface for a Process to report its internal state to Execution Management. | |

### [ SWRD\_API\_EM\_00012] WorkerRunnable class

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00012 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02510 | |
| **CR** |  | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | class | |
| **Symbol** | WorkerRunnable | |
| **Scope** | namespace ara::exec | |
| **Syntax** | template <typename ValueType>  class WorkerRunnable {...}; | |
| **Template param:** | typename ValueType | Value type of Container passed to Deterministic Client::RunWorkerPool |
| **Header file** | #include "ara/exec/worker\_runnable.h" | |
| **Description** | Base-class for implementation of worker runnable for Deterministic Client. | |

#### [ SWRD\_API\_EM\_00013] WorkerRunnable::Run

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00013 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02520 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | Run(ValueType &element, ara::exec::WorkerThread &t) | |
| **Scope** | class ara::exec::WorkerRunnable | |
| **Syntax** | virtual void Run (ValueType &element, ara::exec::WorkerThread &t)=0; | |
| **Parameters (in)** | element | Reference to container element |
| t | Reference to worker thread (for random numbers) |
| **Return value:** | None | |
| **Header file** | #include "ara/exec/worker\_runnable.h" | |
| **Description** | Deterministic client worker runnable. | |

### [ SWRD\_API\_EM\_00014] WorkerThread class

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00014 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02530 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| ***Kind*** | class |
| **Symbol** | WorkerThread |
| **Scope** | namespace ara::exec |
| **Syntax** | class WorkerThread {...}; |
| **Header file** | #include "ara/exec/worker\_thread.h" |
| **Description** | Class to implement worker thread for Deterministic Client . |

#### [ SWRD\_API\_EM\_00015] WorkerThread::WorkerThread

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00015 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02531 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| ***Kind*** | function |
| **Symbol** | WorkerThread() |
| **Scope** | class ara::exec::WorkerThread |
| **Syntax** | WorkerThread (); |
| **Header file** | #include "ara/exec/worker\_thread.h" |
| **Description** | Constructor . |

#### [ SWRD\_API\_EM\_00016] WorkerThread::~WorkerThread

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00016 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02532 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| ***Kind*** | function |
| **Symbol** | ~WorkerThread() |
| **Scope** | class ara::exec::WorkerThread |
| **Syntax** | virtual ~WorkerThread (); |
| **Header file** | #include "ara/exec/worker\_thread.h" |
| **Description** | Destructor . |

#### [ SWRD\_API\_EM\_00017] WorkerThread::GetRandom

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00017 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02540 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | GetRandom() | |
| **Scope** | class ara::exec::WorkerThread | |
| **Syntax** | std::uint64\_t GetRandom () noexcept; | |
| **Return value** | std::uint64\_t | Deterministic random number |
| **Header file** | #include "ara/exec/worker\_thread.h" | |
| **Exception Safety** | noexcept | |
| **Description** | Returns a deterministic pseudo-random number which is unique for each container element. | |

### [ SWRD\_API\_EM\_00018] DeterministicClient class

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00018 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02210 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | class |
| **Symbol** | DeterministicClient |
| **Scope** | namespace ara::exec |
| **Syntax** | class DeterministicClient final {...}; |
| **Header file** | #include "ara/exec/deterministic\_client.h" |
| **Description** | Class to implement operations on Deterministic Client . |

#### [ SWRD\_API\_EM\_00019] DeterministicClient::DeterministicClient

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00019 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02211 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | function |
| **Symbol** | DeterministicClient() |
| **Scope** | class ara::exec::DeterministicClient |
| **Syntax** | DeterministicClient () noexcept; |
| **Header file** | #include "ara/exec/deterministic\_client.h" |
| **Description** | Constructor for DeterministicClient which opens the Execution Management communication channel (e.g. POSIX FIFO) to access a wait point for cyclic execution, a worker pool, deterministic random numbers and time stamps . |

#### [ SWRD\_API\_EM\_00020] DeterministicClient::~DeterministicClient

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00020 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02215 |
| **CR** |  |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| Kind: | function |
| Symbol: | ~DeterministicClient() |
| Scope: | class ara::exec::DeterministicClient |
| Syntax: | ~DeterministicClient () noexcept; |
| Exception Safety: | noexcept |
| Header file: | #include "ara/exec/deterministic\_client.h" |
| Description: | Destructor of the Deterministic Client instance . |

#### [ SWRD\_API\_EM\_00021] DeterministicClient::WaitForActivation

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00021 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02217 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | WaitForActivation() | |
| **Scope** | class ara::exec::DeterministicClient | |
| **Syntax** | ara::core::Result<ActivationReturnType> WaitForActivation () noexcept; | |
| **Return value:** | ara::core::Result< ActivationReturn Type > | Process control value (or error) In the absence of an error, the return value contains the activation state defined by ara::exec::ActivationReturnType. |
| **Exception Safety:** | noexcept | |
| **Errors:** | ara::exec::ExecErrc::kCycleOverrun | The deterministic activation cycle time exceeded. |
| ara::exec::ExecErrc::kFailed | Requested operation could not be performed. |
| **Header file** | #include "ara/exec/deterministic\_client.h" | |
| **Description:** | Blocks and returns with a process control value when the next activation is triggered by the Runtime . | |

#### [ SWRD\_API\_EM\_00022] DeterministicClient::RunWorkerPool

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00022 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02221 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | RunWorkerPool(WorkerRunnable< ValueType > &runnableObj, Container &container) | |
| **Scope** | class ara::exec::DeterministicClient | |
| **Syntax** | template <typename ValueType, typename Container> ara::core::Result<void> RunWorkerPool (WorkerRunnable< ValueType > &runnableObj, Container &container) noexcept; | |
| **Template param:** | ValueType | Element type of container |
| Container | Container for which method WorkerRunnable::Run is invoked for each element |
| **Parameters (in):** | runnableObj | Object derived from WorkerRunnable that provides a method called Run(...), which will be called on every container element |
| container | C++ container which supports a standard iterator interface with - begin() - end() - operator\*() operator++ |
| **Return value:** | ara::core::Result< void > | - |
| **Exception Safety:** | noexcept | |
| **Errors:** | ara::exec::kFailed | Not in ActivationReturnType::kRun/ActivationReturn Type::kInit cycle state |
| **Header file** | #include "ara/exec/deterministic\_client.h" | |
| **Description** | Run a deterministic worker pool.  Uses a pool of Deterministic workers to call a method WorkerRunnable::Run for every element of the container. The sequential iteration is guaranteed by using the container’s increment operator. The API provides the guarantee that no other iteration scheme is used.  This function shall not participate in overload resolution unless ValueType is compatible with Container::value\_type. | |

#### [ SWRD\_API\_EM\_00023] DeterministicClient::GetRandom

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00023 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02225 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | GetRandom() | |
| **Scope** | class ara::exec::DeterministicClient | |
| **Syntax** | std::uint64\_t GetRandom () noexcept; | |
| **Return value:** | std::uint64\_t | uint64\_t 64 bit uniform distributed pseudo random number |
| **Header file** | #include "ara/exec/deterministic\_client.h" | |
| **Description** | Return deterministic sequence of random numbers.  This returns the next in a sequence of ‘Deterministic’ random numbers. Deterministic’ means, that the returned random numbers are identical within redundant DeterministicClient::WaitFor NextActivation() cycles, which are used within redundantly executed Processes. | |

#### [ SWRD\_API\_EM\_00024] DeterministicClient::SetRandomSeed

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00024 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02226 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | SetRandomSeed(std::uint64\_t seed) | |
| **Scope** | class ara::exec::DeterministicClient | |
| **Syntax** | void SetRandomSeed (std::uint64\_t seed) noexcept; | |
| **Parameters (in)** | seed | Random number seed to DeterministicClient::Set RandomSeed. |
| **Return value:** | None | |
| **Exception Safety:** | noexcept | |
| **Header file** | #include "ara/exec/deterministic\_client.h | |
| **Description** | Seed random number generator used for redundantly executed deterministic clients. | |

#### [ SWRD\_API\_EM\_00025] DeterministicClient::GetActivationTime

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00025 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02231 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | GetActivationTime() | |
| **Scope** | class ara::exec::DeterministicClient | |
| **Syntax** | ara::core::Result<TimeStamp> GetActivationTime () noexcept; | |
| **Return value:** | ara::core::Result< TimeStamp > | TimeStamp of current activation cycle |
| **Exception Safety:** | noexcept | |
| **Errors:** | ara::exec::ExecErrc::kNoTimeStamp | Time stamp not available |
| **Header file** | #include "ara/exec/deterministic\_client.h | |
| **Description** | TimeStamp of activation point.  This method provides the timestamp that represents the point in time when the activation was triggered by DeterministicClient::WaitForNextActivation() with return value kRun. Subsequent calls within an activation cycle will always provide the same value. The same value will also be provided within redundantly executed Processes | |

#### [ SWRD\_API\_EM\_00026] DeterministicClient::GetNextActivationTime

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00026 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02236 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | GetNextActivationTime() | |
| **Scope** | class ara::exec::DeterministicClient | |
| **Syntax** | ara::core::Result<TimeStamp> GetNextActivationTime () noexcept; | |
| **Return value** | ara::core::Result< TimeStamp > | TimeStamp of next activation cycle |
| **Errors:** | ara::exec::ExecErrc::kNoTimeStamp | Time stamp not available |
| **Header file** | #include "ara/exec/deterministic\_client.h" | |
| **Exception Safety** | noexcept | |
| **Description** | Timestamp of next activation point.  This method provides the timestamp that represents the point in time when the next activation will be triggered by DeterministicClient::WaitForNextActivation() with return value kRun.  Subsequent calls within an activation cycle will always provide the same value. The same value will also be provided within redundantly executed Process | |

### [ SWRD\_API\_EM\_00027] FunctionGroup class

此类的一个实例将表示在元模型 (ARXML) 中定义的功能组。 此类旨在成为元模型内部信息的实现特定表示。 一旦基于 ARXML 路径创建，其内部值将在对象的整个生命周期内保持绑定。

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00027 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02263 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | class |
| **Symbol** | FunctionGroup |
| **Scope** | namespace ara::exec |
| **Syntax** | class FunctionGroup final {...}; |
| **Header file** | #include "ara/exec/function\_group.h" |
| **Description** | Class representing Function Group defined in meta-model (ARXML). |
| **Notes:** | Once created based on ARXML path, it’s internal value stay bounded to it for entire lifetime of an object. |

#### [ SWRD\_API\_EM\_00028] FunctionGroup::Create

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00028 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02323 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | Create(ara::core::StringView metaModelIdentifier) | |
| **Scope** | class ara::exec::FunctionGroup | |
| **Syntax** | static ara::core::Result<FunctionGroup> Create (ara::core::StringView metaModelIdentifier) noexcept; | |
| **Parameters (in)** | metaModelIdentifier | stringified meta model identifier (short name path) where path separator is ’/’. |
| **Return value** | ara::core::Result<FunctionGroup > | an instance of FunctionGroup, or ExecErrc error. |
| **Exception Safety** | noexcept | |
| **Thread Safety** | Thread-safe | |
| **Errors** | ara::exec::ExecErrc::kMetaModelError | if metaModelIdentifier passed is incorrect (e.g. FunctionGroupState identifier has been passed). |
| ara::exec::ExecErrc::kGeneralError | if any other error occurs. |
| **Header file** | #include "ara/exec/function\_group.h" | |
| **Description** | Named constructor for FunctionGroup.  This method shall validate/verify meta-model path passed and perform FunctionGroup object creation. | |

#### [ SWRD\_API\_EM\_00029] FunctionGroup::FunctionGroup

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00029 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02321 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| ***Kind*** | function |
| **Symbol** | FunctionGroup() |
| **Scope** | class ara::exec::FunctionGroup |
| **Syntax** | FunctionGroup ()=delete; |
| **Header file** | #include "ara/exec/function\_group.h" |
| **Description** | Default constructor. |
| **Notes** | Default constructor is deleted in favour of named constructor (Create). |

#### [ SWRD\_API\_EM\_00030] FunctionGroup::FunctionGroup (Copy Constructor)

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00030 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02322 |
| **CR** |  |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| ***Kind*** | function |
| **Symbol** | FunctionGroup(const FunctionGroup &other) |
| **Scope** | class ara::exec::FunctionGroup |
| **Syntax** | FunctionGroup (const FunctionGroup &other)=delete; |
| **Header file** | #include "ara/exec/function\_group.h" |
| **Description** | Copy constructor. |
| **Notes** | To prevent problems with resource allocations during copy operation, this class is non-copyable. |

#### [ SWRD\_API\_EM\_00031] FunctionGroup::FunctionGroup (Move Constructor)

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00031 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02328 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | FunctionGroup(FunctionGroup &&other) | |
| **Scope** | class ara::exec::FunctionGroup | |
| **Syntax** | FunctionGroup (FunctionGroup &&other) noexcept; | |
| **DIRECTION NOT**  **DEFINED** | other | - |
| **Header file** | #include "ara/exec/function\_group.h" | |
| **Description** | Move constructor. | |

#### [ SWRD\_API\_EM\_00032] FunctionGroup:: operator= (Copy assignment operator)

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00032 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02327 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| ***Kind*** | function |
| **Symbol** | operator=(const FunctionGroup &other) |
| **Scope** | class ara::exec::FunctionGroup |
| **Syntax** | FunctionGroup& operator= (const FunctionGroup &other)=delete; |
| **Header file** | #include "ara/exec/function\_group.h" |
| **Description** | Copy assignment operator. |
| **Notes** | To prevent problems with resource allocations during copy operation, this class is non-copyable. |

#### [ SWRD\_API\_EM\_00033] FunctionGroup::operator= (Move assignment operator)

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00033 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02329 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 新增 | |
| ***Kind*** | function | |
| **Symbol** | operator=(FunctionGroup &&other) | |
| **Scope** | class ara::exec::FunctionGroup | |
| **Syntax** | FunctionGroup& operator= (FunctionGroup &&other) noexcept; | |
| **Return value** | FunctionGroup & | - |
| **Header file** | #include "ara/exec/function\_group.h" | |
| **Description** | Move assignment operator. | |

#### [ SWRD\_API\_EM\_00034] FunctionGroup::~FunctionGroup

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00034 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02266 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 新增 |
| ***Kind*** | function |
| **Symbol** | ~FunctionGroup() |
| **Scope** | class ara::exec::FunctionGroup |
| **Syntax** | ~FunctionGroup () noexcept; |
| **Header file** | #include "ara/exec/function\_group.h" |
| **Description** | Destructor of the FunctionGroup instance. |

#### [ SWRD\_API\_EM\_00035] FunctionGroup::operator==

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00035 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02267 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | operator==(const FunctionGroup &other) | |
| **Scope** | class ara::exec::FunctionGroup | |
| **Syntax** | bool operator== (const FunctionGroup &other) const noexcept; | |
| **Parameters (in)** | other | FunctionGroup instance to compare this one with. |
| **Return value** | bool | true in case both FunctionGroups are representing exactly the same meta-model element, false otherwise. |
| **Exception Safety** | noexcept | |
| **Header file** | #include "ara/exec/function\_group.h" | |
| **Thread Safety** | Thread-safe | |
| **Description** | eq operator to compare with other FunctionGroup instance. | |

#### [ SWRD\_API\_EM\_00036] FunctionGroup::operator!=

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00036 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02268 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | operator!=(const FunctionGroup &other) | |
| **Scope** | class ara::exec::FunctionGroup | |
| **Syntax** | bool operator!= (const FunctionGroup &other) const noexcept; | |
| **Parameters (in)** | other | FunctionGroup instance to compare this one with. |
| **Return value** | bool | false in case both FunctionGroups are representing exactly the same meta-model element, true otherwise. |
| **Header file** | #include "ara/exec/function\_group.h" | |
| **Exception Safety** | noexcept | |
| **Thread Safety** | Thread-safe | |
| **Description** | uneq operator to compare with other FunctionGroup instance. | |

### [ SWRD\_API\_EM\_00037] FunctionGroupState class

此类的一个实例将表示在元模型 (ARXML) 中定义的功能组状态。 此类旨在成为元模型内部信息的实现特定表示。 一旦基于 ARXML 路径创建，其内部值将在对象的整个生命周期内保持绑定。

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00037 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02269 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | class |
| **Symbol** | FunctionGroupState |
| **Scope** | namespace ara::exec |
| **Syntax** | class FunctionGroupState final {...}; |
| **Header file** | #include "ara/exec/function\_group\_state.h" |
| **Description** | Class representing Function Group State defined in meta-model (ARXML). |
| **Notes** | Once created based on ARXML path, it’s internal value stay bounded to it for entire lifetime of  an object. |

#### [ SWRD\_API\_EM\_00038] FunctionGroupState::Create

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00038 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02326 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | Create(const FunctionGroup &functionGroup, ara::core::StringView metaModelIdentifier) | |
| **Scope** | class ara::exec::FunctionGroupState | |
| **Syntax** | static ara::core::Result<FunctionGroupState> Create (const Function Group &functionGroup, ara::core::StringView metaModelIdentifier) noexcept; | |
| **Parameters (in)** | functionGroup | the FunctionGroup instance the state shall be connected with. |
| metaModelIdentifier | stringified meta model identifier (short name path) where path separator is ’/’. |
| **Return value** | ara::core::Result<FunctionGroupState > | an instance of FunctionGroupState, or ExecError Domain error. |
| **Errors** | ara::exec::ExecErrc::kMetaModelError | if metaModelIdentifier passed is incorrect (e.g.FunctionGroup identifier has been passed). |
| ara::exec::ExecErrc::kGeneralError | if any other error occurs |
| **Header file** | #include "ara/exec/function\_group\_state.h" | |
| **Description** | Named constructor for FunctionGroupState.  This method shall validate/verify meta-model path passed and perform FunctionGroupState object creation. | |
| **Exception Safety** | noexcept | |
| **Thread Safety** | Thread-safe | |

#### [ SWRD\_API\_EM\_00039] FunctionGroupState::FunctionGroupState

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00039 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02324 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | function |
| **Symbol** | FunctionGroupState() |
| **Scope** | class ara::exec::FunctionGroupState |
| **Syntax** | FunctionGroupState ()=delete; |
| **Header file** | #include "ara/exec/function\_group\_state.h" |
| **Description** | Default constructor. |
| **Notes** | Default constructor is deleted in favour of named constructor (Create). |

#### [ SWRD\_API\_EM\_00040] FunctionGroupState::FunctionGroupState (Copy Constructor)

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00040 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02325 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | function |
| **Symbol** | FunctionGroupState(const FunctionGroupState &other) |
| **Scope** | class ara::exec::FunctionGroupState |
| **Syntax** | FunctionGroupState (const FunctionGroupState &other)=delete; |
| **Header file** | #include "ara/exec/function\_group\_state.h" |
| **Description** | Copy constructor. |
| **Notes** | To prevent problems with resource allocations during copy operation, this class is non-copyable. |

#### [ SWRD\_API\_EM\_00041] FunctionGroupState::FunctionGroupState (Move Constructor)

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00041 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02331 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | FunctionGroupState(FunctionGroupState &&other) | |
| **Scope** | class ara::exec::FunctionGroupState | |
| **Syntax** | FunctionGroupState (FunctionGroupState &&other) noexcept; | |
| **DIRECTION NOT**  **DEFINED** | other | - |
| **Header file** | #include "ara/exec/function\_group\_state.h" | |
| **Description** | Move constructor. | |

#### [ SWRD\_API\_EM\_00042] FunctionGroupState::operator= (Copy assignment operator)

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00042 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02330 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | function |
| **Symbol** | operator=(const FunctionGroupState &other) |
| **Scope** | class ara::exec::FunctionGroupState |
| **Syntax** | FunctionGroupState& operator= (const FunctionGroupState &other)=delete; |
| **Header file** | #include "ara/exec/function\_group\_state.h" |
| **Description** | Copy assignment operator. |
| **Notes** | To prevent problems with resource allocations during copy operation, this class is non-copyable. |

#### [ SWRD\_API\_EM\_00043] FunctionGroupState::operator= (Move assignment operator)

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00043 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02332 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | operator=(FunctionGroupState &&other) | |
| **Scope** | class ara::exec::FunctionGroupState | |
| **Syntax** | FunctionGroupState& operator= (FunctionGroupState &&other) noexcept; | |
| **DIRECTION NOT**  **DEFINED** | other | - |
| **Return value** | FunctionGroupState & | - |
| **Header file** | #include "ara/exec/function\_group\_state.h" | |
| **Exception Safety** | noexcept | |
| **Description** | Move assignment operator. | |

#### [ SWRD\_API\_EM\_00044] FunctionGroupState::~FunctionGroupState

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00044 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02272 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | function |
| **Symbol** | ~FunctionGroupState() |
| **Scope** | class ara::exec::FunctionGroupState |
| **Syntax** | ~FunctionGroupState () noexcept; |
| **Header file** | #include "ara/exec/function\_group\_state.h" |
| **Exception Safety** | noexcept |
| **Description** | Destructor of the FunctionGroupState instance. |

#### [ SWRD\_API\_EM\_00045] FunctionGroupState::operator==

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00045 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02273 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | operator==(const FunctionGroupState &other) | |
| **Scope** | class ara::exec::FunctionGroupState | |
| **Syntax** | bool operator== (const FunctionGroupState &other) const noexcept; | |
| **Parameters (in)** | other | FunctionGroupState instance to compare this one with. |
| **Return value** | bool | true in case both FunctionGroupStates are representing exactly the same meta-model element,false otherwise. |
| **Header file** | #include "ara/exec/function\_group\_state.h" | |
| **Exception Safety** | noexcept | |
| **Thread Safety** | Thread-safe | |
| **Description** | eq operator to compare with other FunctionGroupState instance | |

#### [ SWRD\_API\_EM\_00046] FunctionGroupState::operator!=

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00046 | |
| **Type** | Valid | |
| **Priority** | L | |
| **Upstream ID** | SWS\_EM\_02274 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | operator!=(const FunctionGroupState &other) | |
| **Scope** | class ara::exec::FunctionGroupState | |
| **Syntax** | bool operator!= (const FunctionGroupState &other) const noexcept; | |
| **Parameters (in)** | other | FunctionGroupState instance to compare this one with. |
| **Return value** | bool | false in case both FunctionGroupStates are  representing exactly the same meta-model element, true otherwise. |
| **Header file** | #include "ara/exec/function\_group\_state.h" | |
| **Exception Safety** | noexcept | |
| **Thread Safety** | Thread-safe | |
| **Description** | uneq operator to compare with other FunctionGroupState instance. | |

### [ SWRD\_API\_EM\_00047] StateClient class

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00047 |
| **Type** | Valid |
| **Priority** | H |
| **Upstream ID** | SWS\_EM\_02275 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | class |
| **Symbol** | StateClient |
| **Scope** | namespace ara::exec |
| **Syntax** | class StateClient final {...}; |
| **Header file** | #include "ara/exec/state\_client.h" |
| **Description** | Class representing connection to Execution Management that is used to request Function Group state transitions (or other operations). |
| **Notes** | StateClient opens communication channel to Execution Management (e.g. POSIX FIFO). Each Process that intends to perform state management, shall create an instance of this class and it shall have rights to use it. |

#### [ SWRD\_API\_EM\_00048] StateClient::StateClient

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00048 | |
| **Type** | Valid | |
| **Priority** | H | |
| **Upstream ID** | SWS\_EM\_02276 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | StateClient(std::function< void(const ara::exec::ExecutionErrorEvent &)> undefinedState Callback) | |
| **Scope** | class ara::exec::StateClient | |
| **Syntax** | explicit StateClient (std::function< void(const ara::exec::Execution ErrorEvent &)> undefinedStateCallback) noexcept; | |
| **Parameters (in)** | undefinedStateCallback | callback to be invoked by StateClient library if a FunctionGroup changes its state unexpectedly to an Undefined Function Group State, i.e. without previous request by SetState(). The affected FunctionGroup and ExecutionError is provided as an argument to the callback in form of ExecutionError Event. |
| **Header file** | #include "ara/exec/state\_client.h" | |
| **Exception Safety** | noexcept | |
| **Description** | Constructor that creates State Client instance.  Registers given callback which is called in case a Function Group changes its state unexpectedly to an Undefined Function Group State. | |

#### [ SWRD\_API\_EM\_00049] StateClient::SetState

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00049 | |
| **Type** | Valid | |
| **Priority** | H | |
| **Upstream ID** | SWS\_EM\_02278 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | SetState(const FunctionGroupState &state) | |
| **Scope** | class ara::exec::StateClient | |
| **Syntax** | ara::core::Future<void> SetState (const FunctionGroupState &state)  const noexcept; | |
| **Parameters (in)** | state | representing meta-model definition of a state inside a specific Function Group. Execution Management will perform state transition from the current state to the state identified by this parameter. |
| **Return value** | ara::core::Future< void > | void if requested transition is successful, otherwise it returns ExecErrorDomain error. |
| **Errors** | ara::exec::ExecErrc::kCancelled | if transition to the requested Function Group state was cancelled by a newer request |
| ara::exec::ExecErrc::kFailed | if transition to the requested Function Group state failed |
| ara::exec::ExecErrc::kFailed UnexpectedTerminationOnExit | if Unexpected Termination in Process of previous Function Group State happened. |
| ara::exec::ExecErrc::kFailed  UnexpectedTerminationOnEnter | if Unexpected Termination in Process of target  Function Group State happened. |
| ara::exec::ExecErrc::kInvalidArguments | if arguments passed doesn’t appear to be valid (e.g.after a software update, given functionGroup doesn’t exist anymore) |
| ara::exec::ExecErrc::kCommunicationError | if StateClient can’t communicate with Execution Management (e.g. IPC link is down) |
| ara::exec::ExecErrc::kAlreadyInState | if the FunctionGroup is already in the requested state |
| ara::exec::ExecErrc::kInTransitionTo SameState | if a transition to the requested state is already ongoing |
| ara::exec::ExecErrc::kInvalidTransition | if transition to the requested state is prohibited (e.g.Off state for MachineFG) |
| ara::exec::ExecErrc::kGeneralError | if any other error occurs |
| **Header file** | #include "ara/exec/state\_client.h" | |
| **Description** | Method to request state transition for a single Function Group.  This method will request Execution Management to perform state transition and return immediately. Returned ara::core::Future can be used to determine result of requested transition. | |

#### [ SWRD\_API\_EM\_00050] StateClient::GetInitialMachineStateTransitionResult

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00050 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02279 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | GetInitialMachineStateTransitionResult() | |
| **Scope** | class ara::exec::StateClient | |
| **Syntax** | ara::core::Future<void> GetInitialMachineStateTransitionResult () const noexcept; | |
| **Return value** | ara::core::Future< void > | void if requested transition is successful, otherwise it returns ExecErrorDomain error. |
| **Errors** | ara::exec::ExecErrc::kCancelled | if transition to the requested Function Group state was cancelled by a newer request |
| ara::exec::ExecErrc::kFailed | if transition to the requested Function Group state failed |
| ara::exec::ExecErrc::kCommunication  Error | if StateClient can’t communicate with Execution Management (e.g. IPC link is down) |
| ara::exec::ExecErrc::kGeneralError | if any other error occurs. |
| **Header file** | #include "ara/exec/state\_client.h" | |
| **Exception Safety** | noexcept | |
| **Thread Safety** | thread-safe | |
| **Description** | Method to retrieve result of Machine State initial transition to Startup state.  This method allows State Management to retrieve result of a transition specified by SWS\_EM\_01023 and SWS\_EM\_02241. Please note that this transition happens once per machine life cycle, thus result delivered by this method shall not change (unless machine is started again). | |

#### [ SWRD\_API\_EM\_00051] StateClient::GetExecutionError

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00051 | |
| **Type** | Valid | |
| **Priority** | H | |
| **Upstream ID** | SWS\_EM\_02542 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | GetExecutionError(const ara::exec::FunctionGroup &functionGroup) | |
| **Scope** | class ara::exec::StateClient | |
| **Syntax** | ara::core::Result<ara::exec::ExecutionErrorEvent> GetExecutionError  (const ara::exec::FunctionGroup &functionGroup) noexcept; | |
| **Parameters (in)** | functionGroup | Function Group of interest. |
| **Return value** | ara::core::Result< ara::exec::Execution ErrorEvent > | The execution error which changed the given  Function Group to an Undefined Function Group State. |
| **Errors** | ara::exec::ExecErrc::kFailed | Given Function Group is not in an Undefined Function Group State. |
| ara::exec::ExecErrc::kCommunication  Error | if StateClient can’t communicate with Execution Management (e.g. IPC link is down) |
| **Header file** | #include "ara/exec/state\_client.h" | |
| **Exception Safety** | noexcept | |
| **Thread Safety** | thread-safe | |
| **Description** | Returns the execution error which changed the given Function Group to an Undefined Function Group State.  This function will return with error and will not return an ExecutionErrorEvent object, if the given Function Group is in a defined Function Group state again. | |

### [ SWRD\_API\_EM\_00052] Execution Management error codes

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00052 | |
| **Type** | Valid | |
| **Priority** | H | |
| **Upstream ID** | SWS\_EM\_02281 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | enumeration | |
| **Symbol** | ExecErrc | |
| **Scope** | namespace ara::exec | |
| **Underlying type:** | ara::core::ErrorDomain::CodeType | |
| **Syntax** | enum class ExecErrc : ara::core::ErrorDomain::CodeType {...}; | |
| **Values** | kGeneralError= 1 | Some unspecified error occurred |
| kInvalidArguments= 2 | Invalid argument was passed |
| kCommunicationError= 3 | Communication error occurred |
| kMetaModelError= 4 | Wrong meta model identifier passed to a function |
| kCancelled= 5 | Transition to the requested Function Group statewas cancelled by a newer request |
| kFailed= 6 | Requested operation could not be performed |
| kFailedUnexpectedTerminationOnExit=7 | Unexpected Termination during transition in Processof previous Function Group State happened |
| kFailedUnexpectedTerminationOnEnter= 8 | Unexpected Termination during transition in Processof target Function Group State happened |
| kInvalidTransition= 9 | Transition invalid (e.g. report kRunning whenalready in Running Process State) |
| kAlreadyInState= 10 | Transition to the requested Function Group statefailed because it is already in requested state |
| kInTransitionToSameState= 11 | Transition to the requested Function Group statefailed because transition to requested state isalready in progress |
| kNoTimeStamp= 12 | DeterministicClient time stamp information is notavailable |
| kCycleOverrun= 13 | Deterministic activation cycle time exceeded |
| **Header file** | #include "ara/exec/exec\_error\_domain.h" | |
| **Description** | Defines an enumeration class for the Execution Management error codes. | |

### [ SWRD\_API\_EM\_00053] ExecException type

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00053 |
| **Type** | Valid |
| **Priority** | L |
| **Upstream ID** | SWS\_EM\_02282 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | class |
| **Symbol** | ExecException |
| **Scope** | namespace ara::exec |
| **Base class** | ara::core::Exception |
| **Syntax** | class ExecException : public Exception {...}; |
| **Header file** | #include "ara/exec/exec\_error\_domain.h” |
| **Description** | Defines a class for exceptions to be thrown by the Execution Management. |

#### [ SWRD\_API\_EM\_00054] ExecException::ExecException

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00054 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02283 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | ExecException(ara::core::ErrorCode errorCode) | |
| **Scope** | class ara::exec::ExecException | |
| **Syntax** | explicit ExecException (ara::core::ErrorCode errorCode) noexcept; | |
| **Parameters (in)** | errorCode | The error code. |
| **Header file** | #include "ara/exec/exec\_error\_domain.h | |
| **Exception Safety** | noexcept | |
| **Description** | Constructs a new ExecException object containing an error code. | |

### [ SWRD\_API\_EM\_00055] GetExecErrorDomain function

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00055 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02290 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | GetExecErrorDomain() | |
| **Scope** | namespace ara::exec | |
| **Syntax** | const ara::core::ErrorDomain& GetExecErrorDomain () noexcept; | |
| **Return value** | const ara::core::ErrorDomain & | Return a reference to the global ExecErrorDomain object. |
| **Header file** | #include "ara/exec/exec\_error\_domain.h" | |
| **Exception Safety** | noexcept | |
| **Description** | Returns a reference to the global ExecErrorDomain object. | |

### [ SWRD\_API\_EM\_00056] MakeErrorCode function

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00056 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02291 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | MakeErrorCode(ara::exec::ExecErrc code, ara::core::ErrorDomain::SupportDataType data) | |
| **Scope** | namespace ara::exec | |
| **Syntax** | ara::core::ErrorCode MakeErrorCode (ara::exec::ExecErrc code, ara::core::ErrorDomain::SupportDataType data) noexcept; | |
| **Parameters (in)** | code | Error code number. |
| data | Vendor defined data associated with the error. |
| **Return value** | ara::core::ErrorCode | An ErrorCode object |
| **Header file** | #include "ara/exec/exec\_error\_domain.h" | |
| **Exception Safety** | noexcept | |
| **Description** | Creates an instance of ErrorCode. | |

### [ SWRD\_API\_EM\_00057] ExecErrorDomain type

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00057 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02284 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | class |
| **Symbol** | ExecErrorDomain |
| **Scope** | namespace ara::exec |
| **Base class** | ara::core::ErrorDomain |
| **Syntax** | class ExecErrorDomain final : public ErrorDomain {...}; |
| **Header file** | #include "ara/exec/exec\_error\_domain.h" |
| **Description** | Defines a class representing the Execution Management error domain. |

#### [ SWRD\_API\_EM\_00058] ExecErrorDomain::ExecErrorDomain

|  |  |
| --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00058 |
| **Type** | Valid |
| **Priority** | M |
| **Upstream ID** | SWS\_EM\_02286 |
| **CR** | - |
| **Consistency** | Yes |
| **Change Type** | 修改 |
| ***Kind*** | function |
| **Symbol** | ExecErrorDomain() |
| **Scope** | class ara::exec::ExecErrorDomain |
| **Syntax** | ExecErrorDomain () noexcept; |
| **Header file** | #include "ara/exec/exec\_error\_domain.h" |
| **Description** | Constructs a new ExecErrorDomain object. |

#### [ SWRD\_API\_EM\_00059] ExecErrorDomain::Name

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00059 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02287 SWS\_EM\_02292 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | Name() | |
| **Scope** | class ara::exec::ExecErrorDomain | |
| **Syntax** | const char\* Name () const noexcept override; | |
| **Return value** | const char \* | "Exec" |
| **Header file** | #include "ara/exec/exec\_error\_domain.h" | |
| **Exception Safety** | noexcept | |
| **Description** | Returns a string constant associated with ExecErrorDomain. | |

#### [ SWRD\_API\_EM\_00060] ExecErrorDomain::Message

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00060 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02288 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | Message(CodeType errorCode) | |
| **Scope** | class ara::exec::ExecErrorDomain | |
| **Syntax** | const char\* Message (CodeType errorCode) const noexcept override; | |
| **Parameters (in)** | errorCode | The error code number. |
| **Return value** | const char \* | The message associated with the error code. |
| **Header file** | #include "ara/exec/exec\_error\_domain.h" | |
| **Exception Safety** | noexcept | |
| **Description** | Returns the message associated with errorCode. | |

#### [ SWRD\_API\_EM\_00061] ExecErrorDomain::ThrowAsException

|  |  |  |
| --- | --- | --- |
| **SWRD-ID** | SWRD\_API\_EM\_00061 | |
| **Type** | Valid | |
| **Priority** | M | |
| **Upstream ID** | SWS\_EM\_02289 | |
| **CR** | - | |
| **Consistency** | Yes | |
| **Change Type** | 修改 | |
| ***Kind*** | function | |
| **Symbol** | ThrowAsException(const ara::core::ErrorCode &errorCode) | |
| **Scope** | class ara::exec::ExecErrorDomain | |
| **Syntax** | void ThrowAsException (const ara::core::ErrorCode &errorCode) const noexcept(false) override; | |
| **Parameters (in)** | errorCode | The error to throw. |
| **Return value** | None | |
| **Exception Safety** | noexcept(false) | |
| **Header file** | #include "ara/exec/exec\_error\_domain.h" | |
| **Description** | Creates a new instance of ExecException from errorCode and throws it as a C++ exception. | |

附录A- 信息定义

|  |  |  |
| --- | --- | --- |
| 类别 | 结构 | 备注 |
| SWRD-ID | SWRD-{需求类型}-{功能简称}-流水号  功能简称：参见下面功能简称列表  需求类型：功能需求为空，非功能需求为NF,接口为API  流水号：从00001开始的5位自然数 | *例：*  *SWRD-Nvm-00001*  *SWRD-NF\_Nvm-00001*  *SWRD-API-Nvm-00001* |

|  |  |
| --- | --- |
| 功能简称列表(aCore) | 说明 |
| DM\_DEM | Diagnostics management模块的诊断事件管理 |
| DM\_DCM | Diagnostics management模块的诊断通信管理 |
| DM\_DCM\_DOIP | Diagnostics management模块的DO/IP相关功能 |
| CoreTypes | 核心数据类型 |
| CM\_CommunicationAPI | Communication management模块的Communication API相关功能 |
| CM\_SOMEIP | Communication management模块的SOME/IP相关功能 |
| CM\_DDS | Communication management模块的DDS相关功能 |
| CM\_CommunicationGroup | Communication management模块的通信组相关功能 |
| CM\_SHM | Communication management模块的共享内存相关功能 |
| CM\_IPC | Communication management模块的IPC相关功能 |
| CM\_Raw | Communication management模块的raw data streaming相关功能 |
| CM\_TLS | Communication management模块的TLS相关功能 |
| CM\_S2S | Communication management模块的S2S相关功能 |
| CM\_E2E | Communication management模块的E2E相关功能 |
| UCM\_Master | Update and config management模块的主站相关功能 |
| UCM\_Server | Update and config management模块的从站相关功能 |
| LT | Log and trace模块相关功能 |
| PHM | Platform health management模块相关功能 |
| Per | Persistency模块相关功能 |
| SM | State management模块相关功能 |
| Crypto | Cryptography模块相关功能 |
| EM | Execution mangement模块相关功能 |
| NM | Network management模块相关功能 |
| TS | Time synchronization模块相关功能 |

说明：根据项目情况可自己定义，增加功能简称

|  |  |  |
| --- | --- | --- |
| 安全等级(ASIL) | 解释说明 | 备注 |
| ASIL A | 根据S – Severity(严重度)  E – Exposure(暴露度) C – Controllability(可控性) 排定功能安全等级。详细理解可以参考26262标准文件。 | *如果有关于ASIL等级的特殊解释说明，请记录在此* |
| ASIL B |  |
| ASIL C |  |
| ASIL D |  |
| QM(A) | 从ASIL A到ASIL D 中分解出来，分解的标准，参考功能安全体系文件《功能安全需求分解指南\_FS.pdf》  其中:括号内的等级为原始等级。 |  |
| QM(B) |  |
| QM(C) |  |
| QM(D) |  |
| ASIL A(A) |  |
| ASIL A(B) |  |
| ASIL A(C) |  |
| ASIL A(D) |  |
| ASIL B(B) |  |
| ASIL B(C) |  |
| ASIL B(D) |  |
| ASIL C(C) |  |
| ASIL C(D) |  |
| ASIL D(D) |  |

|  |  |  |
| --- | --- | --- |
| 优先级(Priority) | 解释说明 | 备注 |
| H | 高优先级 | *例：被依赖的需求优先级设置为H级别* |
| M | 中优先级 | *例：* |
| L | 低优先级 | *例：其余功能均设置为L级别* |
| N/A | 不适用 |  |

|  |  |  |
| --- | --- | --- |
| 类型  (Type) | 状态说明 | 备注 |
| Valid | 有效 | *例：表示需要对应* |
| InValid | 不适用 | *例：表示不做对应* |
| TBD | 检讨中 | *例：表示正在检讨中* |

说明：根据项目情况可自己定义，但需要明确

|  |  |  |
| --- | --- | --- |
| 状态  (Status) | 状态说明 | 备注 |
| Draft | 草稿 | *例：表示新建* |
| In Review | 评审中 | *例：表示处于评审中* |
| Approved | 批准 | *例：表示通过评审* |
| Released | 发布 | *例：表示通过客户确认* |
| Modified | 修改 | *例：表示正在检讨修改中* |

|  |  |  |
| --- | --- | --- |
| 变更类型  (Change Type) | 解释说明 | 备注 |
| 新增 | 与AutoSAR标准*XXX*相比，新增的需求。 | 如果有关于每个变更类型的特殊解释说明，请记录在此 |
| 修改 | 与AutoSAR标准*XXX*相比，发生了修改的需求 |  |
| 不变 | 与AutoSAR标准*XXX*相比，没有变更的需求。 |  |
| 删除 | 与AutoSAR标准*XXX*相比，没有变更的需求。 |  |

说明：根据项目情况可自己定义，但需要明确

附录B- 配置信息

|  |  |  |  |
| --- | --- | --- | --- |
| 配置信息 | 说明 | 范围 | 备注 |
| *API configuration class* |  | *1、2、3* |  |
|  |  |  |  |
|  |  |  |  |