Software Detail Design Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | - | | |
| Deliver to | Generic Generation Tool AUTOSAR R4.2.2  Software Detail Design Document | T.B.D | | |
| RVC/Senior Managers  RVC/Managers  RVC/QA  REL/QA  REL/Managers  REL/Project Leaders | Renesas Design Vietnam  Software Engineering Division  MCU Software Engineering Dept | | |
| Approved | Checked | Issued |
| - | - | Loc Nguyen |

Table of contents

[0 Preface 7](#_Toc8056946)

[0.1 Manual conventions 7](#_Toc8056947)

[0.2 Related documents 7](#_Toc8056948)

[0.3 How to read this document 7](#_Toc8056949)

[0.4 Abbreviations & Acronyms 7](#_Toc8056950)

[1 Introduction 9](#_Toc8056951)

[1.1 Scope 9](#_Toc8056952)

[1.2 Generation Tool Overview 9](#_Toc8056953)

[2 Architecture 11](#_Toc8056954)

[2.1 Overview of Generation Tool Architecture 11](#_Toc8056955)

[2.2 Dependencies to other modules 13](#_Toc8056956)

[3 Environment 14](#_Toc8056957)

[3.1 Hardware Requirements 14](#_Toc8056958)

[3.2 Software Requirements 14](#_Toc8056959)

[4 Design 15](#_Toc8056960)

[4.1 Concept 15](#_Toc8056961)

[4.1.1 Presentation............................................. ......16](#_Toc8056962)

[4.1.2 Business 16](#_Toc8056963)

[4.1.3 Data ..................................................................................................................................................18](#_Toc8056964)

[4.1.4 Cross cutting 19](#_Toc8056965)

[4.2 File structure 20](#_Toc8056966)

[5 Class 23](#_Toc8056967)

[5.1 Renesas::Generator::MCALConfGen::Business::CommandLine::CommandLine 42](#_Toc8056968)

[5.1.1 Fields 43](#_Toc8056969)

[5.1.2 Methods 44](#_Toc8056970)

[5.2 Renesas::Generator::MCALConfGen::Business::CommandLine::CommandLineInput 55](#_Toc8056971)

[5.2.1 Fields 56](#_Toc8056972)

[5.2.2 Methods 56](#_Toc8056973)

[5.3 Renesas::Generator::MCALConfGen::Business::CommandLine::OptionAttribute 56](#_Toc8056974)

[5.3.1 Fields 57](#_Toc8056975)

[5.3.2 Methods 57](#_Toc8056976)

[5.4 Renesas::Generator::MCALConfGen::Business::Generation::FileGeneration 58](#_Toc8056977)

[5.4.1 Fields 58](#_Toc8056978)

[5.4.2 Methods 60](#_Toc8056979)

[5.5 Renesas::Generator::MCALConfGen::Business::Generation::Generation 65](#_Toc8056980)

[5.5.1 Fields 66](#_Toc8056981)

[5.5.2 Methods 67](#_Toc8056982)

[5.6 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructReferenceGenerationItem 68](#_Toc8056983)

[5.6.1 Fields 68](#_Toc8056984)

[5.6.2 Methods 68](#_Toc8056985)

[5.7 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::BreakLineGenerationItem 69](#_Toc8056986)

[5.7.1 Fields 69](#_Toc8056987)

[5.7.2 Methods 69](#_Toc8056988)

[5.8 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructMemberGenerationItem 70](#_Toc8056989)

[5.8.1 Fields 70](#_Toc8056990)

[5.8.2 Methods 70](#_Toc8056991)

[5.9 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::DefineGenerationItem 72](#_Toc8056992)

[5.9.1 Fields 72](#_Toc8056993)

[5.9.2 Methods 72](#_Toc8056994)

[5.10 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StringGenerationItem 75](#_Toc8056995)

[5.10.1 Fields 75](#_Toc8056996)

[5.10.2 Methods 75](#_Toc8056997)

[5.11 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructGenerationItem 76](#_Toc8056998)

[5.11.1 Fields 77](#_Toc8056999)

[5.11.2 Methods 78](#_Toc8057000)

[5.12 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::VariableGenerationItem 83](#_Toc8057001)

[5.12.1 Fields 84](#_Toc8057002)

[5.12.2 Methods 84](#_Toc8057003)

[5.13 Renesas::Generator::MCALConfGen::Business::Generation::HeaderFileGeneration 86](#_Toc8057004)

[5.13.1 Fields 86](#_Toc8057005)

[5.13.2 Methods 86](#_Toc8057006)

[5.14 Renesas::Generator::MCALConfGen::Business::Generation::ItemGenerationAttribute 89](#_Toc8057007)

[5.14.1 Fields 89](#_Toc8057008)

[5.14.2 Methods 90](#_Toc8057009)

[5.15 Renesas::Generator::MCALConfGen::Business::Generation::NewLineAttribute 91](#_Toc8057010)

[5.15.1 Fields 91](#_Toc8057011)

[5.15.2 Methods 91](#_Toc8057012)

[5.16 Renesas::Generator::MCALConfGen::Business::Generation::Section 91](#_Toc8057013)

[5.16.1 Fields 92](#_Toc8057014)

[5.16.2 Methods 92](#_Toc8057015)

[5.17 Renesas::Generator::MCALConfGen::Business::Generation::Settings::GenerationSettings 93](#_Toc8057016)

[5.17.1 Fields 93](#_Toc8057017)

[5.17.2 Methods 93](#_Toc8057018)

[5.18 Renesas::Generator::MCALConfGen::Business::Generation::Settings::InstanceSetting 93](#_Toc8057019)

[5.18.1 Fields 94](#_Toc8057020)

[5.18.2 Methods 94](#_Toc8057021)

[5.19 Renesas::Generator::MCALConfGen::Business::Generation::SourceFileGeneration 95](#_Toc8057022)

[5.19.1 Fields 95](#_Toc8057023)

[5.19.2 Methods 95](#_Toc8057024)

[5.20 Renesas::Generator::MCALConfGen::Business::Intermediate::BaseIntermediate 98](#_Toc8057025)

[5.20.1 Fields 98](#_Toc8057026)

[5.20.2 Methods 99](#_Toc8057027)

[5.21 Renesas::Generator::MCALConfGen::Business::Parser::Parser 101](#_Toc8057028)

[5.21.1 Fields 101](#_Toc8057029)

[5.21.2 Methods 101](#_Toc8057030)

[5.22 Renesas::Generator::MCALConfGen::Business::Parser::ParserXML 103](#_Toc8057031)

[5.22.1 Fields 105](#_Toc8057032)

[5.22.2 Methods 106](#_Toc8057033)

[5.23 Renesas::Generator::MCALConfGen::Business::Validation::BaseValidation 125](#_Toc8057034)

[5.23.1 Fields 126](#_Toc8057035)

[5.23.2 Methods 128](#_Toc8057036)

[5.24 Renesas::Generator::MCALConfGen::Business::Validation::ValidationResult 138](#_Toc8057037)

[5.24.1 Fields 138](#_Toc8057038)

[5.24.2 Methods 139](#_Toc8057039)

[5.25 Renesas::Generator::MCALConfGen::Business::Validation::ValidationRuleAttribute 141](#_Toc8057040)

[5.25.1 Fields 142](#_Toc8057041)

[5.25.2 Methods 143](#_Toc8057042)

[5.26 Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::ConsoleWrapper 144](#_Toc8057043)

[5.26.1 Fields 144](#_Toc8057044)

[5.26.2 Methods 144](#_Toc8057045)

[5.27 Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::EnvironmentWrapper 145](#_Toc8057046)

[5.27.1 Fields 146](#_Toc8057047)

[5.27.2 Methods 146](#_Toc8057048)

[5.28 Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::IOWrapper 146](#_Toc8057049)

[5.28.1 Fields 147](#_Toc8057050)

[5.28.2 Methods 147](#_Toc8057051)

[5.29 Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::ReflectionWrapper 155](#_Toc8057052)

[5.29.1 Fields 156](#_Toc8057053)

[5.29.2 Methods 156](#_Toc8057054)

[5.30 Renesas::Generator::MCALConfGen::CrossCutting::Logger::DebugLogger 158](#_Toc8057055)

[5.30.1 Fields 158](#_Toc8057056)

[5.30.2 Methods 159](#_Toc8057057)

[5.31 Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactory 160](#_Toc8057058)

[5.31.1 Fields 160](#_Toc8057059)

[5.31.2 Methods 161](#_Toc8057060)

[5.32 Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactory::CacheObject 167](#_Toc8057061)

[5.32.1 Fields 168](#_Toc8057062)

[5.32.2 Methods 168](#_Toc8057063)

[5.33 Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactoryAttribute 168](#_Toc8057064)

[5.33.1 Fields 169](#_Toc8057065)

[5.33.2 Methods 170](#_Toc8057066)

[5.34 Renesas::Generator::MCALConfGen::CrossCutting::UserInterface::UserInterface 170](#_Toc8057067)

[5.34.1 Fields 171](#_Toc8057068)

[5.34.2 Methods 173](#_Toc8057069)

[5.35 Renesas::Generator::MCALConfGen::CrossCutting::Util::Constant 181](#_Toc8057070)

[5.35.1 Fields 187](#_Toc8057071)

[5.35.2 Methods 216](#_Toc8057072)

[5.36 Renesas::Generator::MCALConfGen::CrossCutting::Util::StringUtils 216](#_Toc8057073)

[5.36.1 Fields 217](#_Toc8057074)

[5.36.2 Methods 217](#_Toc8057075)

[5.37 Renesas::Generator::MCALConfGen::Data::BasicConfiguration::BasicConfiguration 226](#_Toc8057076)

[5.37.1 Fields 226](#_Toc8057077)

[5.37.2 Methods 229](#_Toc8057078)

[5.38 Renesas::Generator::MCALConfGen::Data::CDFData::BaseRegisterProcessing 230](#_Toc8057079)

[5.38.1 Fields 231](#_Toc8057080)

[5.38.2 Methods 232](#_Toc8057081)

[5.39 Renesas::Generator::MCALConfGen::Data::CDFData::CdfDataDictionary 235](#_Toc8057082)

[5.39.1 Fields 236](#_Toc8057083)

[5.39.2 Methods 237](#_Toc8057084)

[5.40 Renesas::Generator::MCALConfGen::Data::CDFData::Items::BswConfig 244](#_Toc8057085)

[5.40.1 Fields 245](#_Toc8057086)

[5.40.2 Methods 248](#_Toc8057087)

[5.41 Renesas::Generator::MCALConfGen::Data::CDFData::Items::Configuration 248](#_Toc8057088)

[5.41.1 Fields 248](#_Toc8057089)

[5.41.2 Methods 249](#_Toc8057090)

[5.42 Renesas::Generator::MCALConfGen::Data::CDFData::Items::Container 249](#_Toc8057091)

[5.42.1 Fields 250](#_Toc8057092)

[5.42.2 Methods 252](#_Toc8057093)

[5.43 Renesas::Generator::MCALConfGen::Data::CDFData::Items::ItemValue 256](#_Toc8057094)

[5.43.1 Fields 256](#_Toc8057095)

[5.43.2 Methods 257](#_Toc8057096)

[5.44 Renesas::Generator::MCALConfGen::Data::CDFData::RepositoryLinQ 258](#_Toc8057097)

[5.44.1 Fields 259](#_Toc8057098)

[5.44.2 Methods 260](#_Toc8057099)

[5.45 Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateData 271](#_Toc8057100)

[5.45.1 Fields 271](#_Toc8057101)

[5.45.2 Methods 272](#_Toc8057102)

[5.46 Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateItemAttribute 273](#_Toc8057103)

[5.46.1 Fields 274](#_Toc8057104)

[5.46.2 Methods 274](#_Toc8057105)

[5.47 Renesas::Generator::MCALConfGen::Data::IntermediateData::Items::BaseIntermediateItem 274](#_Toc8057106)

[5.47.1 Fields 275](#_Toc8057107)

[5.47.2 Methods 276](#_Toc8057108)

[5.48 Renesas::Generator::MCALConfGen::Data::RuntimeConfiguration::RuntimeConfiguration 278](#_Toc8057109)

[5.48.1 Fields 279](#_Toc8057110)

[5.48.2 Methods 282](#_Toc8057111)

[5.49 Renesas::Generator::MCALConfGen::Presentation::MainEntry::MainEntry 285](#_Toc8057112)

[5.49.1 Fields 286](#_Toc8057113)

[5.49.2 Methods 286](#_Toc8057114)

[6 Data Types 291](#_Toc8057115)

[6.1 SectionName 291](#_Toc8057116)

[6.2 MessageType 292](#_Toc8057117)

[6.3 ParseConfigStatus 292](#_Toc8057118)

[7 Error, Warning and Information Messages 294](#_Toc8057119)

[8 Error handling 295](#_Toc8057120)

[8.1 Responsibility 295](#_Toc8057121)

[8.2 Implementation 295](#_Toc8057122)

[8.3 Error handling flow 296](#_Toc8057123)

[Uncovered List 297](#_Toc8057124)

[9 Revision History 300](#_Toc8057125)

***List of Figures***

[Figure 1 Generic Generation Tool Hierarchical Architecture 11](file:///E:\02_repo\U2A16\99_Other\u2x_baseline\internal\X2x\common\generic\docs\sds\RH850_U2x_Generic_Gentool_DD.docx#_Toc8056855)

[Figure 2 Concept of Generic Generation Tool 15](file:///E:\02_repo\U2A16\99_Other\u2x_baseline\internal\X2x\common\generic\docs\sds\RH850_U2x_Generic_Gentool_DD.docx#_Toc8056856)

[Table 1: Generic File List 20](#_Toc8056857)

[Figure 3 Overview of class diagram 23](#_Toc8056858)

# Preface

## Manual conventions

1. Grey highlighted text marks topics, which need to be discussed, hence are not fixed until Grey highlighting is removed. (E.g.: Bus speed of 1 Mbit/s @ φCAN = 8 MHz)
2. Gold shaded text marks topics, which cannot be opened to customer. (E.g. CGMCONF register)

## Related documents

| **Sl. No.** | **Title/File name** | **Revision** |
| --- | --- | --- |
| [1] | RH850/U2x AUTOSAR R4.2.2 MCAL Getting Started 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | 1.0.0 |
| [2] | Gentool Class Diagram Generic U2x Document  RH850\_U2x\_GENERIC\_Gentool\_DD\_Diagrams.EAP | 1.0.0 |
| [3] | AUTOSAR R4.2.2 RH850/X2x Generic Generation Tool Error List  RH850\_X2x\_Generic\_GenTool\_ErrorList.xlsx | 1.0.0 |

## How to read this document

| **Chapter** | **Contents** |
| --- | --- |
| Chapter 1 (Introduction) | This section contains the Scope of the document and Generation Tool overview. |
| Chapter 2 (Architecture) | This section contains overview of Generic Generation Tool architecture and dependency to other modules. |
| Chapter 3 (Environment) | This section contains Hardware/Software requirements. |
| Chapter 4 (Design) | This section contains Concept, File Structure for Generic Generation Tool. |
| Chapter 5 (Class) | This section contains Class and Function Flow for Generic Generation Tool. |
| Chapter 6 (Data Types) | This section contains Data types of Generic Generation Tool. |
| Chapter 7 (Error, Warning and Information Messages) | This section provides Error, Warning and Information Messages. |

## Abbreviations & Acronyms

| **Abbreviations/Acronym** | **Description** |
| --- | --- |
| API | Application Programming Interface |
| AUTOSAR | AUTomotive Open System ARchitecture |
| BSWMDT | Basic Software Module Description Template |
| CAN | Controller Area Network |
| CFD | Control Flow Diagram |
| CFGXML | Configuration XML File |
| DFD | Data Flow Diagram |
| ECU | Electronic Control Unit |
| Id | Identifier |
| IMR | Interrupt Mask Register |
| ISR | Interrupt Service Request |
| MHz | Mega Hertz |
| PWM | Pulse Width Modulation |
| TAU | Timer Array Unit |
| TRXML | Translation XML File |
| XML | eXtensible Markup Language |

| **Definitions** | **Description** |
| --- | --- |
| Sl. No. | Serial Number |
| e.g. | Example |

# Introduction

## Scope

This document provides generation Tool design for development of R4.2.2 AUTOSAR Common Executable for Renesas U2x. It also provides used, un-used parameters/container list, error, warning and information messages that are specific to AUTOSAR R4.2.2 and Renesas U2x Common Executable.

## Generation Tool Overview

Generation Tool is a command line tool that will parse the ECU Configuration Description File(s), BSWMDT File, TRXML file and CFGXML file and generate modules’ Drivers C Source and Header files.

ECU Configuration Description File(s) for a module Driver contain information about the module General Configuration and Configuration of individual module configuration.

The Generation Tool will generate multiple header files (\*.h), source code files (\*.c) which is used to combine driver.

# Architecture

## Overview of Generation Tool Architecture

Configuration XML File (.cfgxml)

ECU Configuration Description File (.arxml) and BSWMDT File

Translation XML File (.trxml)

Generation Tool

<MSN> DLLs

MCALConfGen Executable



<MSN> Intermediate

Intermediate

<MSN>Post-validation

<MSN>Generation

Generation

Object Factor

Configuration Loader

<MSN> Loader

<MSN>Pre-validation

Pre-validation

Post-validation

Command Line

Parser

C Source & Header Files

Figure 1 Generic Generation Tool Hierarchical Architecture

Generic Driver Generation Tool will accept ECU Configuration Description File (.arxml), BSWMDT File, Translation XML File (.trxml) and Configuration XML File (.cfgxml) as input and generates Generic Driver specific C Source and Header Files as output.

* ECU Configuration Description File (.arxml):

This file will contain Generic Driver specific configuration information. This file should be generated by AUTOSAR specified Configuration Editor.

* BSWMDT File:

Generic Driver Generation Tool uses “Common Published Information” from Generic module specific BSWMDT File. Generic Driver specific BSWMDT File should not be updated manually since it is “Static Configuration” file.

* Translation XML File (.trxml):

This file will contain the path of translation header file and device header file for particular device (microcontroller). Translation header file will contain the mapping of register names while device header file will contain register names and associated address. Based upon the configuration in ECU Configuration Description File (.arxml) and by using Translation XML, Generic Generation Tool fetches the addresses from device header file via translation header file.

* Configuration XML File (.cfgxml):

This file will contain the setting of command line options. This file should be located in the same location where Generic Generation Tool is present. This file is optional file and Generic Generation Tool will use this file only if command line argument contains only Generic Generation Tool.

Generation tool is grouped into 2 parts, MCALConfGen Executable and DLLs of specific modules. When MCALConfGen launches, it only loads appropriate DLLs for a specifc module to start generation process.

* Object Factor:

Look for appropriate DLLs of a module specified by user from command line options and load classes of the DLLs into memory.

This compoment is also used to resolve an interface into one or many instances of concrete classes which actually handle requests at runtime.

* Configuration Loader:

After DLLs are loaded, Configuration Loader will trigger module specific Loader to load module detail information such as module id, module name, etc.

* Command Line:

Accept ECU Configuration Description File(s), BSWMDT File, Translation XML File, Configuration XML File and options provided in the command line as inputs. It will then validate command line inputs for correctness and create data structures to store the command line inputs. These data structures are then used by Parser, Intermediate, Validate and Generate.

* Parser:

Parse ECU Configuration Description File(s), BSWMDT File and Translation XML File and Configuration XML File as input. It will validate files’ content for correctness and create data structures to store data of the files. These data structures are then used by Intermediate, Validate and Generate.

* Pre-validation and Post-validation:

Validate the user configuration. It will display error, warning and information messages, if any unexpected values found in the user configuration.

Validation is grouped into pre-validation and post-validation to improve performance and maintainability. Pre-validation is called before intermediate calculation to validate unexpected values in CDF files and post-validation is called to validate result of intermediate calculation. Generation tool will be aborted if errors found in CDFs data. It means no calculation is performed by intermediate to reduce unneccessary execution time and to prevent posibility of crash with invalid data in CDFs.

MCALConfGen Executable will trigger validation rules in the specific module in both steps pre-validation and post-validation.

* Intermediate:

Calculate intermediate values used for post-validation and output generation.

MCALConfGen Executable will trigger Intermediate calculation in the specific module to calculate Intermediate data.

* Generation:

Generate output header files (\*.h) and source code files (\*.c).

MCALConfGen Executable will trigger generation rules in the specific module to have content of output files.

{Ref: GENERIC\_TUM\_U2X\_GT\_OVV\_001, GENERIC\_TUM\_U2X\_GT\_OVV\_002}

## Dependencies to other modules

MCALConfGen Executable does NOT depend to any specific modules.

# Environment

The minimum hardware and software requirements for proper installation of Module Specific Generation Tool are listed below. This ensures optimal performance of the Tool.

## Hardware Requirements

|  |  |
| --- | --- |
|  |  |
| **Processor** | Pentium/equivalent processor @ 1 GHz or greater |
| **Memory Ram** | RAM 2 GB or greater |
| **Hard Disk Drive** | 20 GB or greater storage capacity |

**{Ref:** GENERIC\_TUM\_U2X\_GT\_INSTREQ\_001**}**

## Software Requirements

|  |  |
| --- | --- |
|  |  |
| **Operating System** | Windows 10 |
| **.NET Framework** | 4.5 |

{Ref: GENERIC\_TUM\_U2X\_GT\_INSTREQ\_002}

# Design

## Concept

Concept of execution for Generation Tool is as follows:

**Runtime Configuration**

**Basic Configuration**

**Command Line**

**CDF Data**

**Main Enty**

Presentation Layer

Data Layer

Business Layer

**Parser**

**Validation**

**Intermediate Data**

**Intermediate**

**Generation**

**Configuration Loader**

Figure 2 Concept of Generic Generation Tool

Cross Cutting

**Object Factory**

**User Interface**

**Util**

**Logger**

**Note:** In the above figure, dotted lines represent data flow and solid lines represent control flow.

“Presentation Layer” contains “Main Entry” which is the entry point of Generation Tool. This coordinates components of Business Layer to perform generation flow:

* Initiate loading DLLs of a specific module.
* Load basic configuration information.
* Read and process command line inputs.
* Parse input files.
* (Pre-)Validate unexpected values in CDF files.
* Calculate intermediate data.
* (Post-)Validate intermediate data.
* Generate header files (\*.h), source code files (\*.c).

“Business Layer” implements the core functionality of the system. It is collection of components which shall handle processing of command line, parser and prepare common framework for validation, intermediate, generation.

“Data Layer” provides access to data used within the Generation Tool. It is collection of components which shall store data and provide read/write methods for difference type of data.

“Cross Cutting” is collection of components which is used by all layers. For example, UserInterface is used by CommandLine when there is any error in input option list. It is also used by Validation in case there are any errors.

### Presentation

#### Main Enty

|  |  |
| --- | --- |
| Location | The “Main Entry” file is as following:  1. MCALConfGen/ Presentation/ MainEntry/ MainEntry.cs |
| Input | Arguments will be input by user to this component. |
| Output | This component will start the execution of the Generation Tool. This initiates loading DLLs, processing of the command line options, parsing input files and generating of output files. |

### Business

#### Configuration Loader

|  |  |
| --- | --- |
| Location | The “Configuration Loader” folder is as following:  1. MCALConfGen/ Business/ ConfigurationLoader  2. <MSN>U2x/ Common/ ConfigurationLoader |
| Input | None |
| Output | * This component will load basic configuration information of a specific module, e.g. module name, module id, required modules, etc. and store the information into basic configuration data. * MCALConfGen Executable provides an interface IBasicConfigurationLoader and class BasicConfigurationLoader which has an abstract method LoadConfiguration that needs to be implemented by module specific configuration loader. |

#### Command Line

|  |  |
| --- | --- |
| Location | The “Command Line” folder is as following:  1. MCALConfGen/ Business/ CommandLine |
| Input | Arguments will be input by user to this component. |
| Output | This component will process inputs from user. It also validates command line inputs for correctness and creates runtime configuration data which is then used by Parser, Intermediate, Validate and Generate. |

#### Parser

|  |  |
| --- | --- |
| Location | The “Parser” folder is as following:  1. MCALConfGen/ Business/ Parser |
| Input | All input files will be parsed by this component. |
| Output | This component will parse XML input files and store into CDF Data. The data will then be validated and configured into output generation. |

#### Validation

|  |  |
| --- | --- |
| Location | The “Validation” folder is as following:  1. MCALConfGen/ Business/ Validation |
| Input | 1. Configuration value from CDF files  2. Calculated intermediate data |
| Output | This component will validate the user configuration. It will display error, warning and information messages, if any unexpected values found in the user configuration.  1. Configuration value from CDF files will be validated by pre-validation.  2. Calculated intermediate data will be validated by post-validation.  Validation is grouped into pre-validation and post-validation to improve performance and maintainability. Pre-validation is called before intermediate calculation to validate unexpected values in CDF files and post-validation is called to validate result of intermediate calculation. Generation tool will be aborted if errors found in CDFs data. It means no calculation is performed by intermediate to reduce unneccessary execution time and to prevent posibility of crash with invalid data in CDFs. |

#### Intermediate

|  |  |
| --- | --- |
| Location | The “Intermediate” folder is as following:  1. MCALConfGen/ Business/ Intermediate |
| Input | CDF data will be used by this component to generate intermediate data. |
| Output | This component creates intermediate data structures, which will contain information required to generate Generation Tool output files. |

#### Generation

|  |  |
| --- | --- |
| Location | The “Generation” folder is as following:  1. MCALConfGen/ Business/ Generation |
| Input | Intermediate data structures will be used as input in this component. |
| Output | This component generates C Source and Header files. |

### Data

#### BasicConfiguration

|  |  |
| --- | --- |
| Location | The “BasicConfiguration” folder is as following:  1. MCALConfGen/ Data/ BasicConfiguration |
| Input | Information in this component is loaded by Configuration Loader. |
| Output | This component will be used by Command Line, Parser, Validation, Intermediate and Generation. |

#### RuntimeConfiguration

|  |  |
| --- | --- |
| Location | The “RuntimeConfiguration” folder is as following:  1. MCALConfGen/ Data/ RuntimeConfiguration |
| Input | Information in this component is loaded by Command Line. |
| Output | This component will be used by Parser, Validation, Intermediate and Generation |

#### CDFData

|  |  |
| --- | --- |
| Location | The “CDFData” folder is as following:  1. MCALConfGen/ Data/ CDFData |
| Input | Information in this component is loaded by Parser. |
| Output | This component will be used by Validation, Intermediate and Generation. |

#### IntermediateData

|  |  |
| --- | --- |
| Location | The “IntermediateData” folder is as following:  1. MCALConfGen/ Data/ IntermediateData |
| Input | Information in this component is calculated by Intermediate. |
| Output | This component will be used by Validation and Generation. |

### Cross cutting

#### ObjectFactory

|  |  |
| --- | --- |
| Location | The “ObjectFactory” folder is as following:  1. MCALConfGen/ CrossCutting/ ObjectFactory |
| Input | Loads all DLLs under dlls subfolder of executable directory which is retrieved from BasicConfiguration. |
| Output | * Resolve an interface into one or many instances of concrete classes which actually handle requests at runtime. * There are multiple concrete classes that implemented the same interface. If only one is needed, ObjectFactory will select one based on AutoSar version, Device Name, and internal Version (Refer to class ObjectFactoryAttribute). |

#### UserInterface

|  |  |
| --- | --- |
| Location | The “UserInterface” folder is as following:  1. MCALConfGen/ CrossCutting/ UserInterface |
| Input | None |
| Output | Displays, logs message(s) which Gentool wants to show to user. |

#### Util

|  |  |
| --- | --- |
| Location | The “Util” folder is as following:  1. MCALConfGen/ CrossCutting/ Util |
| Input | None |
| Output | Contains utilities which are widely used in Gentool, e.g. string common methods or Constant values. |

#### Logger

|  |  |
| --- | --- |
| Location | The “Logger” folder is as following:  1. MCALConfGen/ CrossCutting/ Logger |
| Input | None |
| Output | Records log when Gentool is executed. Used for debugging purpose.  NOTES: this component is just a placeholder and will be implemented in the future. |

## File structure

**GENERIC\_GDD\_FST\_001:**

The file organization for Generic Generation Tool code will be done as shown below:

Table 1: Generic File List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MCALConfGen** | | | | |
| +---Business | | | | |
| | | +---CommandLine | | | |
| | | | |  | CommandLine.cs | |
| | | | |  | CommandLineInput.cs | |
| | | | |  | ICommandLine.cs | |
| | | | |  | OptionAttribute.cs | |
| | | +---ConfigurationLoader | | | |
| | | | |  | BasicConfigurationLoader.cs | |
| | | | |  | IBasicConfigurationLoader.cs | |
| | | +---Generation | | | |
| | | | | | | FileGeneration.cs | |
| | | | | | | Generation.cs | |
| | | | | | | HeaderFileGeneration.cs | |
| | | | | | | IFileGeneration.cs | |
| | | | | | | IGeneration.cs | |
| | | | | | | NewLineAttribute.cs | |
| | | | | | | Section.cs | |
| | | | | | | SectionGenAttribute.cs | |
| | | | | | | SourceFileGeneration.cs | |
| | | | | +---GenerationItems | | |
| | | | |  |  | BaseGenerationItem.cs |
| | | | |  |  | BreakLineGenerationItem.cs |
| | | | |  |  | CommentGenerationItem.cs |
| | | | |  |  | DefineGenerationItem.cs |
| | | | |  |  | StringGenerationItem.cs |
| | | | |  |  | StructGenerationItem.cs |
| | | | |  |  | StructMemberGenerationItem.cs |
| | | | |  |  | StructReferenceGenerationItem.cs |
| | | | |  |  | VariableGenerationItem.cs |
| | | | | +--- Settings | | |
| | | | |  |  | GenerationSettings.cs |
| | | | |  |  | InstanceSetting.cs |
| | | +---Intermediate | | | |
| | | | |  | BaseIntermediate.cs | |
| | | | |  | IIntermediate.cs | |
| | | +---Parser | | | |
| | | | |  | IParser.cs | |
| | | | |  | ParserXML.cs | |
| | | +---Validation | | | |
| | |  |  | BaseValidation.cs | |
| | |  |  | IValidation.cs | |
| | |  |  | ValidationResult.cs | |
| | |  |  | ValidationRuleAttribute.cs | |
| +---CrossCutting | | | | |
| | | +---Logger | | | |
| | | | |  | DebugLogger.cs | |
| | | | |  | ILogger.cs | |
| | | | |  | Minidump.cs | |
| | | +---ObjectFactory | | | |
| | | | |  | ObjectFactory.cs | |
| | | | |  | ObjectFactoryAttribute.cs | |
| | | +---UserInterface | | | |
| | | | |  | IUserInterface.cs | |
| | | | |  | UserInterface.cs | |
| | | +---Util | | | |
| | |  |  | Constant.cs | |
| | |  |  | StringUtils.cs | |
| | | +---DotNetApiWrapper | | | |
| | | | |  | ConsoleWrapper.cs | |
| | | | |  | EnvironmentWrapper.cs | |
| | | | |  | IOWrapper.cs | |
| | | | |  | ReflectionWrapper.cs | |
| +---Data | | | | |
| | | +---BasicConfiguration | | | |
| | | | |  | BasicConfiguration.cs | |
| | | | |  | IBasicConfiguration.cs | |
| | | +---CDFData | | | |
| | | | | | | BaseRegisterProcessing.cs | |
| | | | | | | CdfDataDictionary.cs | |
| | | | | | | ICdfData.cs | |
| | | | | | | IRegisterProcessing.cs | |
| | | | | | | IRepository.cs | |
| | | | | | | RepositoryLinQ.cs | |
| | | | | +---Items | | |
| | | | |  |  | BswConfig.cs |
| | | | |  |  | Configuration.cs |
| | | | |  |  | Container.cs |
| | | | |  |  | ItemValue.cs |
| | | +---IntermediateData | | | |
| | | | | | | IIntermediateData.cs | |
| | | | | | | IntermediateData.cs | |
| | | | | | | IntermediateItemAttribute.cs | |
| | | | | +---Items | | |
| | | | |  |  | BaseIntermediateItem.cs |
| | | +---RuntimeConfiguration | | | |
| | |  |  | IRuntimeConfiguration.cs | |
| | |  |  | RuntimeConfiguration.cs | |
| +---Presentation | | | | |
|  | +---MainEntry | | | |
|  |  |  | MainEntry.cs | |

{Ref: }

# Class

This section contains class diagram and list of the function for Generation Tool.



Figure 3 Overview of class diagram

Note: All methods with type “Protected” are represented by “protected virtual” in source code and can be overrided by subclasses.



1. Renesas::Generator::MCALConfGen::Business::CommandLine::CommandLine



1. Renesas::Generator::MCALConfGen::Business::CommandLine::OptionAttribute



1. Renesas::Generator::MCALConfGen::Business::Generation::FileGeneration



1. Renesas::Generator::MCALConfGen::Business::Generation::Generation



1. Renesas::Generator::MCALConfGen::Business::Generation::HeaderFileGeneration



1. Renesas::Generator::MCALConfGen::Business::Generation::ItemGenerationAttribute



1. Renesas::Generator::MCALConfGen::Business::Generation::NewLineAttribute



1. Renesas::Generator::MCALConfGen::Business::Generation::SourceFileGeneration



1. Renesas::Generator::MCALConfGen::Business::Intermediate::BaseIntermediate



1. Renesas::Generator::MCALConfGen::Business::Parser::ParserXML



1. Renesas::Generator::MCALConfGen::Business::Parser::Parser



1. Renesas::Generator::MCALConfGen::Business::Validation::BaseValidation



1. Renesas::Generator::MCALConfGen::Business::Validation::ValidationRuleAttribute



1. Renesas::Generator::MCALConfGen::CrossCutting::Logger::DebugLogger



1. Renesas::Generator::MCALConfGen::CrossCutting::UserInterface::UserInterface



1. Renesas::Generator::MCALConfGen::Data::BasicConfiguration::BasicConfiguration



1. Renesas::Generator::MCALConfGen::Data::CDFData::BaseRegisterProcessing



1. Renesas::Generator::MCALConfGen::Data::CDFData::CdfDataDictionary



1. Renesas::Generator::MCALConfGen::Data::CDFData::RepositoryLinQ



1. Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateData



1. Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateItemAttribute



1. Renesas::Generator::MCALConfGen::Data::RuntimeConfiguration::RuntimeConfiguration



1. Renesas::Generator::MCALConfGen::Business::Generation::Section



1. Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::ConsoleWrapper



1. Renesas::Generator::MCALConfGen::Business::Generation::Settings::GenerationSettings



1. Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactory



1. Renesas::Generator::MCALConfGen::CrossCutting::Util::Constant



1. Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructReferenceGenerationItem



1. Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StringGenerationItem



1. Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::ReflectionWrapper



1. Renesas::Generator::MCALConfGen::Data::CDFData::Items::Configuration



1. Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::DefineGenerationItem



1. Renesas::Generator::MCALConfGen::Data::CDFData::Items::Container



1. Renesas::Generator::MCALConfGen::Data::CDFData::Items::ItemValue



1. Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::EnvironmentWrapper



1. Renesas::Generator::MCALConfGen::Presentation::MainEntry::MainEntry



1. Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::VariableGenerationItem



1. Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::BreakLineGenerationItem



1. Renesas::Generator::MCALConfGen::Data::IntermediateData::Items::BaseIntermediateItem



1. Renesas::Generator::MCALConfGen::CrossCutting::Util::StringUtils



1. Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructMemberGenerationItem



1. Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::IOWrapper



1. Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactoryAttribute



1. Renesas::Generator::MCALConfGen::Business::Generation::Settings::InstanceSetting



1. Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructGenerationItem



1. Renesas::Generator::MCALConfGen::Data::CDFData::Items::BswConfig



1. Renesas::Generator::MCALConfGen::Business::Validation::ValidationResult



1. Renesas::Generator::MCALConfGen::Business::CommandLine::CommandLineInput

## Renesas::Generator::MCALConfGen::Business::CommandLine::CommandLine

|  |  |
| --- | --- |
| **Location File Name** | CommandLine.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | runtimeConfiguration | Private | To define information processed by command line. |
| (3) | basicConfiguration | Private | To store the basic information of gentool module. |
| (4) | userInterface | Private | To record Error, Warning, Info information and exit Gentool. |
| (5) | logger | Private | To record log information for debugging. |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | PrintToolHelp | Public | Print Gentool help information |
| (2) | GetAndValidateModuleName | Public | Get and validate module name from command line options |
| (3) | ProcessCommandLine | Public | To coordinate other methods to process command line options |
| (4) | CommandLine | Protected | Command line constructor |
| (5) | CommandLine | Protected | Command line constructor |
| (6) | displayCmdInformation | Private | Display command line information |
| (7) | parseAndValidateOptions | Private | Parse and validate options |
| (8) | parseUserInputs | Private | Parse all information from command line options entried by users |
| (9) | filterValueVendorName | Private | Filter value vendor name |
| (10) | checkMissingValueForOption | Private | Checking missing value for options command line. |
| (11) | checkConfigOptionFileName | Private | Check name gentool configuration |
| (12) | checkInvalidOption | Private | Check invalid options |
| (13) | checkOutputDirectory | Private | Check output directory |
| (14) | validateUserInputs | Private | Validate command line options |
| (15) | parseXmlConfig | Private | Parse all information from .cfgxml file |
| (16) | validateXmlConfig | Private | Validate options from .cfgxml files |
| (17) | checkExplicitOptions | Private | Check explicit options |
| (18) | validateBothUserAndXmlInputs | Private | Validate both command line options and .cfgxml file options |
| (19) | validateAcceptedFiles | Private | Validate input files with extension .trxml, .arxml |
| (20) | showInvalidInputPath | Private | Validate input paths |
| (21) | showInvalidOutputPath | Private | Validate output paths |
| (22) | getOptionValueFromXmlConfig | Private | Get an option value from .cfgxml file |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_001\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly CommandLine |
| **Range** | None |
| **Description** | None |

#### runtimeConfiguration

GENERIC\_GDD\_CLS\_001\_002:

|  |  |
| --- | --- |
| **Variable Name** | runtimeConfiguration |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | To define information processed by command line. |

#### basicConfiguration

GENERIC\_GDD\_CLS\_001\_003:

|  |  |
| --- | --- |
| **Variable Name** | basicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | To store the basic information of gentool module. |

#### userInterface

GENERIC\_GDD\_CLS\_001\_004:

|  |  |
| --- | --- |
| **Variable Name** | userInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | To record Error, Warning, Info information and exit Gentool. |

#### logger

GENERIC\_GDD\_CLS\_001\_005:

|  |  |
| --- | --- |
| **Variable Name** | logger |
| **Type** | ILogger |
| **Range** | None |
| **Description** | To record log information for debugging. |

### Methods

#### GetAndValidateModuleName

GENERIC\_GDD\_CLS\_001\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAndValidateModuleName | | |
| **Arguments** | None | | |
| **Return** | string | the module name | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Get and validate module name from command line options  Algorithm: LET module = NULL  LET args = CALL Environment.GetCommandLineArgs().Skip(1).ToArray() to get all input arguments  IF args is not NULL THEN  IF found module name from -m and -module option and it's in a list of supported module names THEN  break  ELSE  module = NULL  ELSE  Not required  IF module is NULL THEN  Print ERR000031 message  ELSE  Not required | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_USG\_003>}

#### PrintToolHelp

GENERIC\_GDD\_CLS\_001\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | PrintToolHelp | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Print Gentool help information  Algorithm: LET args = CALL Environment.GetCommandLineArgs().Skip(1).ToArray() to get all input arguments  LET enaFlag = runtimeConfiguration.PrintHelp  IF enaFlag is not TRUE THEN  FOREACH option IN list of argument  IF option equals to "-h"/"-help" THEN  LET enaFlag = TRUE  ELSE  Not required  ELSE  Not required  IF enaFlag is TRUE THEN  LET info = CALL StringUtils.GetStringFromTemplate(  Resources.HelpToolTemplate,  new Dictionary<string, string> { }) to get message info  CALL ConsoleWrapper.WriteLine(info) to print message  ELSE  Not required | | |

{Ref: N/A}

#### ProcessCommandLine

GENERIC\_GDD\_CLS\_001\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ProcessCommandLine | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | To coordinate other methods to process command line options  Algorithm: CALL displayCmdInformation to display command line information  CALL parseAndValidateOptions to parse and validate options  Print tool help  CALL showInvalidInputPath to validate input path  CALL showInvalidOutputPath to validate output path | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_USG\_002>}

#### CommandLine

GENERIC\_GDD\_CLS\_001\_009:

|  |  |
| --- | --- |
| **Method Name** | CommandLine |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Command line constructor  Algorithm: Init component |

{Ref: N/A}

#### CommandLine

GENERIC\_GDD\_CLS\_001\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CommandLine | | |
| **Arguments** | logger | Logger | |
| Range: | None |
| runtimeConfiguration | Runtime configuration | |
| Range: | None |
| basicConfiguration | Basic configuration | |
| Range: | None |
| userInterface | User interface | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Command line constructor | | |

{Ref: N/A}

#### checkConfigOptionFileName

GENERIC\_GDD\_CLS\_001\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkConfigOptionFileName | | |
| **Arguments** | userInputs | User inputs | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Check name gentool configuration  Algorithm: LET isMissing = CALL checkMissingValueForOption(configOptions, userInputs) to check missing value of option  IF isMissing is TRUE THEN  CALL userInterface.Error WITH Resources.ERR000002 to print ERR000002  ELSE  Not required | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_002>}

{Ref: [3] < CMN\_GDD\_ERR\_002>}

#### checkExplicitOptions

GENERIC\_GDD\_CLS\_001\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkExplicitOptions | | |
| **Arguments** | userInput | User input | |
| Range: | null/valid |
| xmlConfig | Xml config | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Check explicit options  Algorithm: LET singleCmdOptions = new string[] { "-h", "-l", "-d" }  LET notExplicitOptions = new StringBuilder()  IF userInput is not null AND xmlConfig is not null THEN  FOREACH option IN singleCmdOptions WHERE its value matches with option  IF notExplicitOptions.Length > 0 THEN  CALL notExplicitOptions.Append(", ")  ELSE  Do nothing  CALL notExplicitOptions.Append(option)  ELSE  Do nothing  RETURN notExplicitOptions | | |

{Ref: N/A}

#### checkInvalidOption

GENERIC\_GDD\_CLS\_001\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkInvalidOption | | |
| **Arguments** | userInputs | User inputs | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Check invalid options  Algorithm: LET optionsWithValues = userInputs.OptionsWithValues  LET validOptions = CALL runtimeConfiguration.GetAllCmdOptions() to get all commandline options   FOREACH option IN Keys of optionsWithValues  IF validOptions contain option THEN  CALL userInterface.Error WITH Resources.ERR000008 to print ERR000008  ELSE  Not required  IF option is repeated THEN  CALL userInterface.Warn WITH Resources.WRN000002 to print WRN000002  ELSE  Not required | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_008 GENERIC\_TUM\_U2x\_GT\_WRN\_002>}

{Ref: [3] < CMN\_GDD\_ERR\_008>}

#### checkMissingValueForOption

GENERIC\_GDD\_CLS\_001\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkMissingValueForOption | | |
| **Arguments** | options | A list of options that need checking | |
| Range: | null/valid array of string |
| userInputs | User inputs | |
| Range: | null/valid |
| **Return** | bool | The result of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Checking missing value for options command line.  Algorithm: LET ret = FALSE  LET optionsWithValues = userInputs.OptionsWithValues  FOREACH option IN options  IF keys of optionsWithValues contain option THEN  LET ret = TRUE  ELSE  Not required   RETURN ret | | |

{Ref: N/A}

#### checkOutputDirectory

GENERIC\_GDD\_CLS\_001\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkOutputDirectory | | |
| **Arguments** | userInputs | User inputs | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Check output directory  Algorithm: LET optionsWithValues = userInputs.OptionsWithValues  LET outputOptions = new string[] { "-o", "-output", "-oinc", "-osrc" }  FOREACH each IN outputOptions WHERE keys of optionsWithValues contain each  LET optionValue = optionsWithValues[each]  IF optionValue is not NULL THEN  CALL userInterface.Error WITH Resources.ERR000005 to print ERR000005  ELSE IF optionValue is invalid THEN  CALL userInterface.Error WITH Resources.ERR000010 to print ERR000010  ELSE  Do nothing | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_005 GENERIC\_TUM\_U2x\_GT\_ERR\_010>}

{Ref: [3] < CMN\_GDD\_ERR\_005 CMN\_GDD\_ERR\_010>}

#### displayCmdInformation

GENERIC\_GDD\_CLS\_001\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | displayCmdInformation | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Display command line information  Algorithm: LET args = CALL Environment.GetCommandLineArgs().Skip(1).ToArray() to get all input arguments  LET basicConfiguration.ExecutionName = CALL Path.GetFileName(System.Reflection.Assembly.GetExecutingAssembly().Location)  to get execution name  CALL userInterface.Info WITH Resources.Tool\_Version to print INF000001  CALL userInterface.Info WITH basicConfiguration.ExecutionName, String.Join(" ", args) to print INF000002 | | |

{Ref: N/A}

#### filterValueVendorName

GENERIC\_GDD\_CLS\_001\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | filterValueVendorName | | |
| **Arguments** | userInputs | User inputs | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Filter value vendor name  Algorithm: LET isMissing = CALL checkMissingValueForOption(filterOptions, userInputs) to check missing value of filter option  IF isMissing is TRUE THEN  CALL userInterface.Error WITH Resources.ERR000032 to print ERR000032  ELSE  Not required | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_032>}

{Ref: [3] < CMN\_GDD\_ERR\_032>}

#### getOptionValueFromXmlConfig

GENERIC\_GDD\_CLS\_001\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getOptionValueFromXmlConfig | | |
| **Arguments** | document | Parsed .cfgxml file | |
| Range: | null/valid XDocument |
| option | Option | |
| Range: | null/valid string |
| **Return** | string | Value of option extracted from xml config | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Get an option value from .cfgxml file  Algorithm: LET result = String.Empty  LET element = collection of pair of value/option  IF element is not null AND element.Value is not null THEN  LET result = element.Value.ToString()  ELSE  Do nothing   RETURN result | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_USG\_006>}

#### parseAndValidateOptions

GENERIC\_GDD\_CLS\_001\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseAndValidateOptions | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Parse and validate options  Algorithm: LET configOptions = CALL string[] { "-c", "-configfile" }  LET xmlConfigPath = NULL  LET userInputs = CALL parseUserInputs() to parse all information from command line options  CALL validateUserInputs(userInputs) to validate the command line options to handle errors and warning  FOREACH option IN configOptions  IF configuration file is NOT specified by -c/-configfile THEN  LET xmlConfigPath = userInputs.OptionsWithValues[option]  ELSE  Not required  IF xmlConfigPath is NULL THEN  LET xmlConfigPath = default configuration file {MSN}\_{MICRO\_FAMILY\_NAME}.cfgxml at the same MCALConfGen.exe location  ELSE  Not required   LET xmlConfigInputs = CALL parseXmlConfig(xmlConfigPath) to parse all information  CALL validateXmlConfig(xmlConfigPath, xmlConfigInputs) to validate options from .cfgxml files to handle errors and warning   IF output folder is NOT specified by -o/-output THEN  Use default output folder {MODULE\_NAME}\_Output  ELSE  Not required  CALL validateBothUserAndXmlInputs(userInputs, xmlConfigInputs) to validate both user/xml input | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_002 GENERIC\_TUM\_U2x\_GT\_ERR\_005 GENERIC\_TUM\_U2x\_GT\_ERR\_006 GENERIC\_TUM\_U2x\_GT\_ERR\_007, GENERIC\_TUM\_U2x\_GT\_ERR\_008 GENERIC\_TUM\_U2x\_GT\_ERR\_010 GENERIC\_TUM\_U2x\_GT\_ERR\_032 GENERIC\_TUM\_U2x\_GT\_WRN\_001, GENERIC\_TUM\_U2x\_GT\_USG\_004 GENERIC\_TUM\_U2x\_GT\_USG\_005 GENERIC\_TUM\_U2x\_GT\_USG\_006 GENERIC\_TUM\_U2x\_GT\_USG\_007, GENERIC\_TUM\_U2x\_GT\_USG\_008 GENERIC\_TUM\_U2x\_GT\_USG\_009 GENERIC\_TUM\_U2x\_GT\_USG\_010>}

#### parseUserInputs

GENERIC\_GDD\_CLS\_001\_020:

|  |  |
| --- | --- |
| **Method Name** | parseUserInputs |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Parse all information from command line options entried by users  Algorithm: LET args = CALL Environment.GetCommandLineArgs().Skip(1).ToArray() to get all input arguments  IF args is not NULL THEN  FOREACH argument IN args  IF argument start with "-" THEN  ADD argument and its value in to option list of CommandLineInput instance  ELSE  ADD argument into files list of CommandLineInput instance  ELSE  Not required |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_USG\_006>}

#### parseXmlConfig

GENERIC\_GDD\_CLS\_001\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseXmlConfig | | |
| **Arguments** | path | Path to .cfgxml file | |
| Range: | null/valid string |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Parse all information from .cfgxml file  Algorithm: LET result = null  IF path does not exist THEN  LET document = CALL XDocument.Load(path) to load data  LET result = new CommandLineInput()  FOREACH pair IN mapping  LET optionValue = CALL getOptionValueFromXmlConfig(document, pair.Key)  SWITCH pair.Key  CASE Constant.OUTPUT:  IF optionValue equals to Constant.STR\_ON THEN  LET outputPath = CALL getOptionValueFromXmlConfig(document, Constant.OUTPUT\_PATH)  LET result.OptionsWithValues[pair.Value] = outputPath  ELSE  Do nothing  CASE Constant.INPUT\_FILE:  IF optionValue is not NULL THEN  CALL result.Files.AddRange(optionValue.Split(',').Select(e => e.Trim()))  ELSE  Do nothing  CASE Constant.FILTER:  LET vendor = CALL getOptionValueFromXmlConfig(document, Constant.FILTER\_NAME)  IF vendor is not NULL THEN  CALL result.OptionsWithValues[pair.Value] = vendor  ELSE  Do nothing  DEFAULT:  IF optionValue equals to Constant.STR\_ON THEN  CALL result.OptionsWithValues[pair.Value] = Boolean.TrueString  ELSE IF optionValue equals to Constant.STR\_OFF THEN  CALL result.OptionsWithValues[pair.Value] = Boolean.FalseString  ELSE  Do nothing  ELSE  CALL userInterface.ErrorFileNotFound WITH path  RETURN result | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_001, GENERIC\_TUM\_U2X\_GT\_ERR\_003>}

#### showInvalidInputPath

GENERIC\_GDD\_CLS\_001\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | showInvalidInputPath | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Validate input paths  Algorithm: LET listPath = new List<string>()  IF runtimeConfiguration.CDFFilePaths.Count is 0 THEN  CALL userInterface.Error WITH Resources.ERR000011 to raise ERR000011  ELSE  Do nothing  LET listPath.AddRange WITH runtimeConfiguration.CDFFilePaths  LET listPath.AddRange WITH new string[]  {  runtimeConfiguration.ConfigFilePath,  runtimeConfiguration.TranslationFilePath  }  FOREACH path IN listPath WHERE path does not exist or null  CALL userInterface.ErrorFileNotFound WIH path | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_001 GENERIC\_TUM\_U2X\_GT\_ERR\_011>}

{Ref: [3] < CMN\_GDD\_ERR\_011>}

#### showInvalidOutputPath

GENERIC\_GDD\_CLS\_001\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | showInvalidOutputPath | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Validate output paths  Algorithm: IF runtimeConfiguration.FolderOutpu is not null THEN  IF runtimeConfiguration.FolderOutput does not exist THEN  CALL userInterface.Error WITH Resources.ERR000009 to raise ERR000009  ELSE  Do nothing  ELSE  Do nothing  LET paths = new string[]  {  runtimeConfiguration.FolderOutput,  runtimeConfiguration.IncOutputDirectory,  runtimeConfiguration.SrcOutputDirectory  }  FOREACH directory IN paths  CALL IOWrapper.CreateDirectory WITH directory to create new directory | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_009 GENERIC\_TUM\_U2X\_GT\_ERR\_018>}

{Ref: [3] < CMN\_GDD\_ERR\_009 CMN\_GDD\_ERR\_018>}

#### validateAcceptedFiles

GENERIC\_GDD\_CLS\_001\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | validateAcceptedFiles | | |
| **Arguments** | filePaths | File paths | |
| Range: | null/valid list of string |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Validate input files with extension .trxml, .arxml  Algorithm: IF filePaths is not null THEN  LET allowExtensions = new string[] { Constant.TRANSLATION\_FILE\_EXTENSION, Constant.CDF\_EXTENSION }  FOREACH file IN filePaths  IF extension of file does not end with any string in allowExtensions THEN  CALL userInterface.Warn WITH Resources.WRN000001 to raise WRN000001  ELSE  Do nothing  ELSE  Do nothing | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_INF\_001 GENERIC\_TUM\_U2x\_GT\_WRN\_001>}

#### validateBothUserAndXmlInputs

GENERIC\_GDD\_CLS\_001\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | validateBothUserAndXmlInputs | | |
| **Arguments** | userInput | User input | |
| Range: | null/valid |
| xmlConfig | Xml config | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Validate both command line options and .cfgxml file options  Algorithm: LET singleCmdOptions = new string[] { "-h", "-l", "-d" }  LET notExplicitOptions = new StringBuilder()  LET noUserInputs = userInput is null OR value of option '-m' is not provided from user input  LET noXmlInputs = xmlConfig is null OR value of option '-m' is not provided from xml input  IF noUserInputs && noXmlInputs is true THEN  CALL userInterface.Error WITH Resources.ERR000007 to raise ERR000007  CALL userInterface.Info WITH Resources.INF000003 to raise INF000003  ELSE  Do nothing   LET notExplicitOptions = CALL checkExplicitOptions(userInput, xmlConfig)  IF notExplicitOptions.Length > 0 THEN  CALL userInterface.Error WITH Resources.ERR000034 to raise ERR000034  ELSE  Do nothing | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_007 GENERIC\_TUM\_U2x\_GT\_ERR\_034 GENERIC\_TUM\_U2x\_GT\_INF\_003>}

{Ref: [3] < CMN\_GDD\_ERR\_007 CMN\_GDD\_ERR\_034>}

#### validateUserInputs

GENERIC\_GDD\_CLS\_001\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | validateUserInputs | | |
| **Arguments** | userInputs | User inputs | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Validate command line options  Algorithm: IF userInputs is not NULL THEN  CALL checkInvalidOption WITH userInputs  CALL checkConfigOptionFileName WITH userInputs  CALL checkOutputDirectory WITH userInputs  CALL filterValueVendorName WITH userInputs  CALL validateAcceptedFiles WITH userInputs.Files  ELSE  Do nothing | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_002 GENERIC\_TUM\_U2x\_GT\_ERR\_005 GENERIC\_TUM\_U2x\_GT\_ERR\_008 GENERIC\_TUM\_U2x\_GT\_ERR\_010, GENERIC\_TUM\_U2x\_GT\_ERR\_032 GENERIC\_TUM\_U2x\_GT\_WRN\_001>}

#### validateXmlConfig

GENERIC\_GDD\_CLS\_001\_027:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | validateXmlConfig | | |
| **Arguments** | xmlConfigFile | .cfgxml files file path | |
| Range: | null/valid string |
| xmlConfig | Xml config | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Validate options from .cfgxml files  Algorithm: IF xmlConfig is not null THEN  LET optionsWithValues = xmlConfig.OptionsWithValues  IF keys of optionsWithValues contain '-o' AND value of optionsWithValues["-o"] is not null THEN  CALL userInterface.Error WITH Resources.ERR000006 to print ERR000006  ELSE  Do nothing  CALL validateAcceptedFiles(xmlConfig.Files) to check file extension as .trxml, .arxml  ELSE  Do nothing | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_006 GENERIC\_TUM\_U2x\_GT\_WRN\_001>}

{Ref: [3] < CMN\_GDD\_ERR\_006>}

## Renesas::Generator::MCALConfGen::Business::CommandLine::CommandLineInput

|  |  |
| --- | --- |
| **Location File Name** | CommandLineInput.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | OptionsWithValues | Public | Options with their values |
| (2) | OptionsAppearances | Public | Options appearance |
| (3) | Files | Public | Files |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | CommandLineInput | Public | Command line input constructor |

### Fields

#### OptionsWithValues

GENERIC\_GDD\_CLS\_002\_001:

|  |  |
| --- | --- |
| **Variable Name** | OptionsWithValues |
| **Type** | Dictionary< string, string > |
| **Range** | None |
| **Description** | Options with their values |

#### OptionsAppearances

GENERIC\_GDD\_CLS\_002\_002:

|  |  |
| --- | --- |
| **Variable Name** | OptionsAppearances |
| **Type** | Dictionary< string, int > |
| **Range** | None |
| **Description** | Options appearance |

#### Files

GENERIC\_GDD\_CLS\_002\_003:

|  |  |
| --- | --- |
| **Variable Name** | Files |
| **Type** | List< string > |
| **Range** | None |
| **Description** | Files |

### Methods

#### CommandLineInput

GENERIC\_GDD\_CLS\_002\_004:

|  |  |
| --- | --- |
| **Method Name** | CommandLineInput |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Command line input constructor |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::CommandLine::OptionAttribute

|  |  |
| --- | --- |
| **Location File Name** | OptionAttribute.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | option | Private | Option |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | OptionAttribute | Public | Option attribute constructor |
| (2) | GetOption | Public | Get option |

### Fields

#### option

GENERIC\_GDD\_CLS\_003\_001:

|  |  |
| --- | --- |
| **Variable Name** | option |
| **Type** | string |
| **Range** | None |
| **Description** | Option |

### Methods

#### GetOption

GENERIC\_GDD\_CLS\_003\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetOption | | |
| **Arguments** | None | | |
| **Return** | string | the option of attribute | |
| Range: | string.Empty/valid string |
| **Generated Value** | None | | |
| **Description** | Get option  Algorithm: RETURN option | | |

{Ref: N/A}

#### OptionAttribute

GENERIC\_GDD\_CLS\_003\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | OptionAttribute | | |
| **Arguments** | option | Option | |
| Range: | null/valid string |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Option attribute constructor | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::FileGeneration

|  |  |
| --- | --- |
| **Location File Name** | FileGeneration.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | FileName | Public | File name |
| (2) | OutputDirectory | Public | Output directory |
| (3) | Description | Public | Description |
| (4) | Sections | Public | Sections |
| (5) | Logger | Protected | To record log information for debugging. |
| (6) | BasicConfiguration | Protected | To store basic information of Gentool. |
| (7) | RuntimeConfiguration | Protected | To define information processed by command line. |
| (8) | IntermediateData | Protected | To save and retrive data intermediation. |
| (9) | UserInterface | Protected | To record Error, Warning,Info information and exit Gentool. |
| (10) | GenSettings | Protected | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetAttrSection | Public | Get attribute section |
| (2) | GetAttrSectionOrder | Public | Get attribute sectionOrder |
| (3) | GenerateFile | Public | Generate an output file. It can be .h or .c file. |
| (4) | WriteToFile | Public | Write to file |
| (5) | FileGeneration | Protected | File generation constructor |
| (6) | GenCOPYRIGHTSection | Protected | Generate copyright |
| (7) | GenRevisionHistorySection | Protected | Generate revision history |
| (8) | GenIncludeSection | Protected | Generate include section |
| (9) | GenVersionInformationSection | Protected | Generate version information |
| (10) | GenEND\_OF\_FILE | Protected | Generate end of file section |
| (11) | Gen\_GentoolVersion | Protected | Generate tool version |
| (12) | Gen\_InputFiles | Protected | Generate input files' paths |

### Fields

#### FileName

GENERIC\_GDD\_CLS\_004\_001:

|  |  |
| --- | --- |
| **Variable Name** | FileName |
| **Type** | string |
| **Range** | None |
| **Description** | File name |

#### OutputDirectory

GENERIC\_GDD\_CLS\_004\_002:

|  |  |
| --- | --- |
| **Variable Name** | OutputDirectory |
| **Type** | string |
| **Range** | None |
| **Description** | Output directory |

#### Description

GENERIC\_GDD\_CLS\_004\_003:

|  |  |
| --- | --- |
| **Variable Name** | Description |
| **Type** | string |
| **Range** | None |
| **Description** | Description |

#### Sections

GENERIC\_GDD\_CLS\_004\_004:

|  |  |
| --- | --- |
| **Variable Name** | Sections |
| **Type** | Section |
| **Range** | None |
| **Description** | Sections |

#### Logger

GENERIC\_GDD\_CLS\_004\_005:

|  |  |
| --- | --- |
| **Variable Name** | Logger |
| **Type** | ILogger |
| **Range** | None |
| **Description** | To record log information for debugging. |

#### BasicConfiguration

GENERIC\_GDD\_CLS\_004\_006:

|  |  |
| --- | --- |
| **Variable Name** | BasicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | To store basic information of Gentool. |

#### RuntimeConfiguration

GENERIC\_GDD\_CLS\_004\_007:

|  |  |
| --- | --- |
| **Variable Name** | RuntimeConfiguration |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | To define information processed by command line. |

#### IntermediateData

GENERIC\_GDD\_CLS\_004\_008:

|  |  |
| --- | --- |
| **Variable Name** | IntermediateData |
| **Type** | IIntermediateData |
| **Range** | None |
| **Description** | To save and retrive data intermediation. |

#### UserInterface

GENERIC\_GDD\_CLS\_004\_009:

|  |  |
| --- | --- |
| **Variable Name** | UserInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | To record Error, Warning,Info information and exit Gentool. |

#### GenSettings

GENERIC\_GDD\_CLS\_004\_010:

|  |  |
| --- | --- |
| **Variable Name** | GenSettings |
| **Type** | GenerationSettings |
| **Range** | None |
| **Description** | None |

### Methods

#### GenerateFile

GENERIC\_GDD\_CLS\_004\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenerateFile | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate an output file. It can be .h or .c file.  Algorithm: LET stringBuilder = new StringBuilder()  LET sections = Collection of all non-public methods which have ItemGeneration attribute and sort them by Section  FOREACH section IN sections  methods = Collection of all methods sorted by its order  FOREACH method IN methods  LET newLineAttribute = method.GetCustomAttribute<NewLineAttribute>()  IF newLineAttribute is not null THEN  CALL new BreakLineGenerationItem(newLineAttribute.NumberOfNewLines) then ADD to items  ELSE  Do nothing  IF type of method.ReturnType is BaseGenerationItem[] THEN  LET addedItems = (BaseGenerationItem[])method.Invoke(this, new object[] { })  IF addedItems is not null THEN  CALL AddRange(addedItems) to add to items  ELSE  Do nothing  ELSE  Do nothing | | |

{Ref: N/A}

#### GetAttrSection

GENERIC\_GDD\_CLS\_004\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAttrSection | | |
| **Arguments** | method | method metadata | |
| Range: | null/valid MethodInfo |
| **Return** | SectionName | The section name | |
| Range: | COPY\_RIGHT,  REVISION\_HISTORY,  GENTOOL\_VERSION,  INPUT\_FILES,  START\_MACRO,  INSTANCE\_INDEX,  INCLUDE,  QAC\_WARNING,  MISRAC\_RULE,  VERSION\_INFO,  COMMON\_PUBLISHED\_INFO,  GLOBAL\_SYMBOLS,  VERSION\_CHECK,  GLOBAL\_DATA\_TYPE,  GLOBAL\_DATA,  FUNCTION\_PROTOTYPE,  FUNTION\_DEFINE,  END\_MACRO,  END\_OF\_FILE |
| **Generated Value** | None | | |
| **Description** | Get attribute section  Algorithm: LET attribute = CALL method.GetCustomAttribute(typeof(ItemGenerationAttribute), false) to get method's attribute  IF attribute is not null THEN  LET ret = attribute.Section  ELSE  Do nothing  RETURN ret | | |

{Ref: N/A}

#### GetAttrSectionOrder

GENERIC\_GDD\_CLS\_004\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAttrSectionOrder | | |
| **Arguments** | method | method metadata | |
| Range: | null/valid MethodInfo |
| **Return** | float | The section name | |
| Range: | -3.4E+38 to +3.4E+38 |
| **Generated Value** | None | | |
| **Description** | Get attribute sectionOrder  Algorithm: LET attribute = CALL method.GetCustomAttribute(typeof(ItemGenerationAttribute), false) to get method's attribute  IF attribute is not null THEN  LET ret = attribute.SectionOrder  ELSE  Do nothing  RETURN ret | | |

{Ref: N/A}

#### WriteToFile

GENERIC\_GDD\_CLS\_004\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | WriteToFile | | |
| **Arguments** | buffer | The buffer string to be written | |
| Range: | null/valid string |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Write to file  Algorithm: IF RuntimeConfiguration.DryRun is true THEN  IF OutputDirectory is not null AND OutputDirectory does not exist THEN  CALL IOWrapper.CreateDirectory WITH OutputDirectory to create new folder  ELSE  Do nothing  TRY  LET path = string.Concat(OutputDirectory, FileName)  CALL IOWrapper.StreamWrite WITH buffer to write to path  CATCH  Raise ERR000019  ELSE  Do nothing | | |

{Ref: [1] < GENERIC\_TUM\_U2X\_GT\_ERR\_009>}

{Ref: [3] < CMN\_GDD\_ERR\_019>}

#### FileGeneration

GENERIC\_GDD\_CLS\_004\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | FileGeneration | | |
| **Arguments** | fileName | File name | |
| Range: | null/valid string |
| outputDirectory | Output directory | |
| Range: | null/valid string |
| logger | Logger | |
| Range: | null/valid ILogger |
| basicConfiguration | Basic configuration | |
| Range: | null/valid IBasicConfiguration |
| runtimeConfiguration | Runtime configuration | |
| Range: | null/valid IRuntimeConfiguration |
| intermediateData | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | File generation constructor  Algorithm: LET GenSettings.InstanceSetting = new InstanceSetting  {  IsMultiInstance = basicConfiguration.HasInstanceSetting,  InstanceIndex = basicConfiguration.InstanceIndex  }  LET FileName = fileName  IF basicConfiguration.InstanceOutType equals to InstanceOutputType.FILES THEN  LET name = CALL Path.GetFileNameWithoutExtension(fileName) to get file name  LET extension = CALL Path.GetExtension(fileName) to get extension  LET FileName = String.Format("{0}{1}{2}",  name,  GenSettings.InstanceSetting.GetInstanceSuffix(),  extension)   ELSE  Do nothing  LET OutputDirectory = outputDirectory  IF basicConfiguration.InstanceOutType equals to InstanceOutputType.FOLDERS THEN  LET parts = split outputDirectory by DirectorySeparatorChar  CALL basicConfiguration.InstanceIndex.ToString then ADD to end of parts  LET OutputDirectory = Join parts WITH DirectorySeparatorChar  ELSE  Do nothing  LET this.Logger = logger  LET this.BasicConfiguration = basicConfiguration  LET this.RuntimeConfiguration = runtimeConfiguration  LET this.IntermediateData = intermediateData | | |

{Ref: N/A}

#### GenCOPYRIGHTSection

GENERIC\_GDD\_CLS\_004\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenCOPYRIGHTSection | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of copyright info | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate copyright  Algorithm: Generate copyright from template | | |

{Ref: N/A}

#### GenEND\_OF\_FILE

GENERIC\_GDD\_CLS\_004\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenEND\_OF\_FILE | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of end of file section | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate end of file section  Algorithm: RETURN null | | |

{Ref: N/A}

#### GenIncludeSection

GENERIC\_GDD\_CLS\_004\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenIncludeSection | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of include section | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate include section  Algorithm: RETURN null | | |

{Ref: N/A}

#### GenRevisionHistorySection

GENERIC\_GDD\_CLS\_004\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenRevisionHistorySection | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of revision history | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate revision history  Algorithm: RETURN null | | |

{Ref: N/A}

#### GenVersionInformationSection

GENERIC\_GDD\_CLS\_004\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenVersionInformationSection | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of version information | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate version information  Algorithm: RETURN null | | |

{Ref: N/A}

#### Gen\_GentoolVersion

GENERIC\_GDD\_CLS\_004\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Gen\_GentoolVersion | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem | BaseGenerationItem of tool version | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate tool version  Algorithm: Generate tool version from template | | |

{Ref: N/A}

#### Gen\_InputFiles

GENERIC\_GDD\_CLS\_004\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Gen\_InputFiles | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem | BaseGenerationItem of input files' paths | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate input files' paths  Algorithm: Generate file paths which are inputs to generation tool by using a template | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::Generation

|  |  |
| --- | --- |
| **Location File Name** | Generation.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | logger | Private | To record log information for debugging. |
| (3) | basicConfiguration | Private | To store the basic information of gentool module. |
| (4) | runtimeConfiguration | Private | To define information processed by command line. |
| (5) | intermediateData | Private | To save and retrive data intermediation. |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GenerateOuput | Public | Generate all output files |
| (2) | Generation | Private | None |
| (3) | Generation | Private | None |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_005\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly Generation |
| **Range** | None |
| **Description** | None |

#### logger

GENERIC\_GDD\_CLS\_005\_002:

|  |  |
| --- | --- |
| **Variable Name** | logger |
| **Type** | ILogger |
| **Range** | None |
| **Description** | To record log information for debugging. |

#### basicConfiguration

GENERIC\_GDD\_CLS\_005\_003:

|  |  |
| --- | --- |
| **Variable Name** | basicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | To store the basic information of gentool module. |

#### runtimeConfiguration

GENERIC\_GDD\_CLS\_005\_004:

|  |  |
| --- | --- |
| **Variable Name** | runtimeConfiguration |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | To define information processed by command line. |

#### intermediateData

GENERIC\_GDD\_CLS\_005\_005:

|  |  |
| --- | --- |
| **Variable Name** | intermediateData |
| **Type** | IIntermediateData |
| **Range** | None |
| **Description** | To save and retrive data intermediation. |

### Methods

#### GenerateOuput

GENERIC\_GDD\_CLS\_005\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenerateOuput | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate all output files  Algorithm: LET allFiles = ObjectFactory.GetInstances<IFileGeneration>() to get all instances having interface IFileGeneration  IF allFiles is not null THEN  FOREACH file IN allFiles  CALL file.GenerateFile() to generate file  ELSE  Do nothing | | |

{Ref: N/A}

#### Generation

GENERIC\_GDD\_CLS\_005\_007:

|  |  |
| --- | --- |
| **Method Name** | Generation |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** |  |

{Ref: N/A}

#### Generation

GENERIC\_GDD\_CLS\_005\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generation | | |
| **Arguments** | logger | Logger | |
| Range: | null/valid ILogger |
| basicConfiguration | Basic configuration | |
| Range: | null/valid IBasicConfiguration |
| runtimeConfiguration | Runtime configuration | |
| Range: | null/valid IRuntimeConfiguration |
| intermediateData | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructReferenceGenerationItem

|  |  |
| --- | --- |
| **Location File Name** | StructReferenceGenerationItem.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | StructReferenceGenerationItem | Public | Struct reference generation item constructor |

### Fields

### Methods

#### StructReferenceGenerationItem

GENERIC\_GDD\_CLS\_006\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StructReferenceGenerationItem | | |
| **Arguments** | preComment | Pre comment | |
| Range: | null/valid string |
| postComment | Post comment | |
| Range: | null/valid string |
| startQac | start of QAC string | |
| Range: | null/valid string |
| endQac | end of QAC string | |
| Range: | null/valid string |
| value | Value of define statement | |
| Range: | null/valid string |
| hasInstanceSetting | Instance setting | |
| Range: | true/false |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Struct reference generation item constructor | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::BreakLineGenerationItem

|  |  |
| --- | --- |
| **Location File Name** | BreakLineGenerationItem.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | numberOfLines | Private | Number of break lines |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | BreakLineGenerationItem | Public | Break line generation item constructor |
| (2) | Generate | Private | Generate output string of break lines |

### Fields

#### numberOfLines

GENERIC\_GDD\_CLS\_007\_001:

|  |  |
| --- | --- |
| **Variable Name** | numberOfLines |
| **Type** | int |
| **Range** | None |
| **Description** | Number of break lines |

### Methods

#### BreakLineGenerationItem

GENERIC\_GDD\_CLS\_007\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BreakLineGenerationItem | | |
| **Arguments** | numberOfLines | Number of break lines. Default value = 1 | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Break line generation item constructor | | |

{Ref: N/A}

#### Generate

GENERIC\_GDD\_CLS\_007\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | genSettings | generation setting | |
| Range: | null/valid GenerationSettings |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate output string of break lines  Algorithm: LET builder = new StringBuilder()  IF numberOfLines is greater than 0 THEN  CALL builder.Insert(Constant.INT\_ZERO, Environment.NewLine, numberOfLines - Constant.INT\_ONE)  ELSE  Do nothing   RETURN builder.ToString() | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructMemberGenerationItem

|  |  |
| --- | --- |
| **Location File Name** | StructMemberGenerationItem.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | StructMemberGenerationItem | Public | Struct member generation item constructor |
| (2) | Generate | Private | Generate output string of a struct member item |

### Fields

### Methods

#### StructMemberGenerationItem

GENERIC\_GDD\_CLS\_008\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StructMemberGenerationItem | | |
| **Arguments** | preComment | Pre comment | |
| Range: | null/valid string |
| postComment | Post comment | |
| Range: | null/valid string |
| startQac | start of QAC string | |
| Range: | null/valid string |
| endQac | end of QAC string | |
| Range: | null/valid string |
| value | Value of define statement | |
| Range: | null/valid string |
| hasInstanceSetting | Instance setting | |
| Range: | true/false |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Struct member generation item constructor | | |

{Ref: N/A}

#### Generate

GENERIC\_GDD\_CLS\_008\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | genSettings | generation setting | |
| Range: | null/valid GenerationSettings |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate output string of a struct member item  Algorithm: LET stringBuilder = new StringBuilder()  LET indentStr = (new StringBuilder()).Insert(Constant.INT\_ZERO, " ", Level).ToString()   CALL AppendLine(stringBuilder, string.Empty, new string[] { PreComment }) to add pre comment  CALL AppendLine(stringBuilder, string.Empty, StartQAC) to add start of QAC  IF Value is not null THEN  LET instSetting = (genSettings != null) ? genSettings.InstanceSetting : null  LET valueOfInstance = Value  IF HasNameInstanceSetting is true AND instSetting is not null AND instSetting.IsMultiInstance is true  THEN  IF Value is matched with @"(\w+)(.\*\[.\*\])?" THEN  valueOfInstance = String.Format("{0}{1}{2}",  match.Groups[Constant.INT\_ONE].ToString(),  instSetting.GetInstanceSuffix(),  match.Groups[Constant.INT\_TWO].ToString())  ELSE  Do nothing  ELSE  Do nothing  CALL stringBuilder.AppendLine(  indentStr + String.Format("{0}{1}", valueOfInstance, IsLast ? String.Empty : ","))  ELSE  Do nothing   CALL AppendLine(stringBuilder, indentStr, EndQAC) to append end of QAC  CALL AppendLine(stringBuilder, indentStr, new string[] { PostComment }) to append post comment   RETURN stringBuilder.ToString().TrimEnd() | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::DefineGenerationItem

|  |  |
| --- | --- |
| **Location File Name** | DefineGenerationItem.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | DefineGenerationItem | Public | Define generation item constructor |
| (2) | Generate | Private | Generate output string of a define macro |
| (3) | computeSpaces | Private | To compute space needed to generate define items Macro definition has format: #define name {spaces} value This method calculate {spaces} length to make length of macro definition string is MAX\_LINE\_LENGTH Mininum {spaces} length is 1 If length of macro definition string is more than MAX\_LINE\_LENGTH Break into multiple lines End |

### Fields

### Methods

#### DefineGenerationItem

GENERIC\_GDD\_CLS\_009\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | DefineGenerationItem | | |
| **Arguments** | preComment | Pre comment | |
| Range: | null/valid string |
| postComment | Post comment | |
| Range: | null/valid string |
| startQac | start of QAC string | |
| Range: | null/valid string |
| endQac | end of QAC string | |
| Range: | null/valid string |
| name | Name of define statement or struct | |
| Range: | null/valid string |
| value | Value of define statement | |
| Range: | null/valid string |
| level | Level of inner struct | |
| Range: | 0/valid integer |
| hasNameInstanceSetting | Instance setting | |
| Range: | true/false |
| hasValueInstanceSetting | Instance setting | |
| Range: | true/false |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Define generation item constructor | | |

{Ref: N/A}

#### Generate

GENERIC\_GDD\_CLS\_009\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | genSettings | generation setting | |
| Range: | null/valid GenerationSettings |
| **Return** | string | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate output string of a define macro  Algorithm: LET stringBuilder = new StringBuilder()  CALL AppendLine(stringBuilder, string.Empty, new string[] { PreComment }) to add pre comment  CALL AppendLine(stringBuilder, string.Empty, StartQAC) to add start of QAC  LET instSetting = (genSettings != null) ? genSettings.InstanceSetting : null  IF instSetting is not null AND instSetting.IsMultiInstance is true THEN  IF HasNameInstanceSetting is true THEN  nameOfInstance += instSetting.GetInstanceSuffix()  ELSE  Do nothing  IF HasValueInstanceSetting is true THEN  valueOfInstance += instSetting.GetInstanceSuffix()  ELSE  Do nothing  ELSE  Do nothing  CALL stringBuilder.AppendLine(String.Format("#define {0}{1}{2}",  nameOfInstance,  computeSpaces(nameOfInstance, valueOfInstance),  valueOfInstance))  CALL AppendLine(stringBuilder, string.Empty, EndQAC) to append end of QAC  CALL AppendLine(stringBuilder, string.Empty, new string[] { PostComment }) to append post comment | | |

{Ref: N/A}

#### computeSpaces

GENERIC\_GDD\_CLS\_009\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | computeSpaces | | |
| **Arguments** | name | The define name | |
| Range: | null/valid string |
| value | The define value | |
| Range: | null/valid string |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | To compute space needed to generate define items Macro definition has format: #define name {spaces} value This method calculate {spaces} length to make length of macro definition string is MAX\_LINE\_LENGTH Mininum {spaces} length is 1 If length of macro definition string is more than MAX\_LINE\_LENGTH Break into multiple lines End  Algorithm: IF value is not null THEN  LET nameLengthIsOver = name.Length >= (DEFINE\_INDENT - DEFINE\_LENGTH)  LET valueLengthIsOver = value.Length > (MAX\_LINE\_LENGTH - DEFINE\_INDENT + MIN\_SPACE\_LENGTH)  IF nameLengthIsOver is true AND valueLengthIsOver is false THEN  spaces += @" \" + Environment.NewLine + new string(' ', DEFINE\_INDENT)  ELSE IF nameLengthIsOver is false AND valueLengthIsOver is true THEN  spaces += new string(' ', DEFINE\_INDENT - name.Length - DEFINE\_LENGTH) + @"\"  + Environment.NewLine + new string (' ', MAX\_LINE\_LENGTH - value.Length)  ELSE IF nameLengthIsOver is true AND valueLengthIsOver is true THEN  spaces += @" \" + Environment.NewLine + new string(' ', MAX\_LINE\_LENGTH - value.Length)  ELSE  spaces += new string(' ', DEFINE\_INDENT - name.Length - DEFINE\_LENGTH)  ELSE  Do nothing   RETURN spaces | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StringGenerationItem

|  |  |
| --- | --- |
| **Location File Name** | StringGenerationItem.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | StringGenerationItem | Public | String generation item constructor |
| (2) | Generate | Private | Generate output string from Value attribute |

### Fields

### Methods

#### StringGenerationItem

GENERIC\_GDD\_CLS\_010\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StringGenerationItem | | |
| **Arguments** | value | Value of define statement | |
| Range: | null/valid string |
| startQac | start of QAC string | |
| Range: | null/valid string |
| endQac | end of QAC string | |
| Range: | null/valid string |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | String generation item constructor | | |

{Ref: N/A}

#### Generate

GENERIC\_GDD\_CLS\_010\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | genSettings | generation setting | |
| Range: | null/valid GenerationSettings |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate output string from Value attribute  Algorithm: LET stringBuilder = new StringBuilder()  LET indentStr = (new StringBuilder()).Insert(Constant.INT\_ZERO, " ", Level).ToString()  CALL AppendLine(stringBuilder, indentStr, StartQAC) to add start of QAC  CALL stringBuilder.AppendLine(Value) to add value  CALL AppendLine(stringBuilder, indentStr, EndQAC) to add end of QAC   RETURN stringBuilder.ToString() | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructGenerationItem

|  |  |
| --- | --- |
| **Location File Name** | StructGenerationItem.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | MemorySection | Public | The memory section |
| (2) | data | Public | Calculated intermediate item from specific module |
| (3) | qacMapping | Public | The QAC mapping |
| (4) | memorySectionMapping | Public | The memory section mapping |
| (5) | basicConfig | Public | The basic configuration |
| (6) | includeMemMap | Public | The include memmap |
| (7) | children | Private | Inner structs or fields |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | StructGenerationItem | Public | Struct generation item constructor |
| (2) | AddChild | Public | Add an inner struct or a field |
| (3) | GenStructure | Public | Generate the content of struct |
| (4) | Generate | Private | Generate output string of a struct |
| (5) | getQacMessage | Private | Update QAC message |
| (6) | genMemorySection | Private | Generate data for memory section based on struct name |
| (7) | genSingleStruct | Private | Generate a struct that includes only single item |
| (8) | genFieldInStructure | Private | Generate field in a structure |

### Fields

#### MemorySection

GENERIC\_GDD\_CLS\_011\_001:

|  |  |
| --- | --- |
| **Variable Name** | MemorySection |
| **Type** | None |
| **Range** | None |
| **Description** | The memory section |

#### data

GENERIC\_GDD\_CLS\_011\_002:

|  |  |
| --- | --- |
| **Variable Name** | data |
| **Type** | BaseIntermediateItem |
| **Range** | None |
| **Description** | Calculated intermediate item from specific module |

#### qacMapping

GENERIC\_GDD\_CLS\_011\_003:

|  |  |
| --- | --- |
| **Variable Name** | qacMapping |
| **Type** | Dictionary< string, List< string > > |
| **Range** | None |
| **Description** | The QAC mapping |

#### memorySectionMapping

GENERIC\_GDD\_CLS\_011\_004:

|  |  |
| --- | --- |
| **Variable Name** | memorySectionMapping |
| **Type** | Dictionary< string, List< string > > |
| **Range** | None |
| **Description** | The memory section mapping |

#### basicConfig

GENERIC\_GDD\_CLS\_011\_005:

|  |  |
| --- | --- |
| **Variable Name** | basicConfig |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | The basic configuration |

#### includeMemMap

GENERIC\_GDD\_CLS\_011\_006:

|  |  |
| --- | --- |
| **Variable Name** | includeMemMap |
| **Type** | string |
| **Range** | None |
| **Description** | The include memmap |

#### children

GENERIC\_GDD\_CLS\_011\_007:

|  |  |
| --- | --- |
| **Variable Name** | children |
| **Type** | List< BaseGenerationItem > |
| **Range** | None |
| **Description** | Inner structs or fields |

### Methods

#### AddChild

GENERIC\_GDD\_CLS\_011\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AddChild | | |
| **Arguments** | child | Child item | |
| Range: | null/valid BaseGenerationItem |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Add an inner struct or a field  Algorithm: ADD child into children | | |

{Ref: N/A}

#### GenStructure

GENERIC\_GDD\_CLS\_011\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenStructure | | |
| **Arguments** | structName | struct name | |
| Range: | null/valid string |
| qualifier | qualifier | |
| Range: | null/valid string |
| structComment | struct comment | |
| Range: | null/valid string |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate the content of struct  Algorithm: LET ret = new List<BaseGenerationItem>()  LET name = $"{structName}[]"  LET qacMessage = CALL getQacMessage(structName, qacMapping) to get corresponding QAC message  LET module = basicConfig.ModuleName.Split('\_')[0]  ADD memory start section based on struct name  IF structComment is not null THEN  Create struct comment then ADD to ret  ELSE  Do nothing  IF data is not null AND data.Childs.Count is greater than 0 THEN  LET dataType = $"{qualifier}({data.Value}, {module.ToUpper()}\_CONFIG\_DATA)"  LET structGeneration = new StructGenerationItem(String.Empty, string.Empty, qacMessage[0], qacMessage[1], name, dataType)  FOREACH child IN data.Childs  IF child.Childs.Count is greater than 0 THEN  CALL genSingleStruct(child, count, qacMapping) then ADD to structGeneration  ELSE  CALL genFieldInStructure(child, qacMapping) then ADD to structGeneration  ADD structGeneration to ret  ELSE  Do nothing  CALL BreakLineGenerationItem() then ADD to ret  ADD memory stop section based on struct name to ret   RETURN ret | | |

{Ref: N/A}

#### StructGenerationItem

GENERIC\_GDD\_CLS\_011\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StructGenerationItem | | |
| **Arguments** | preComment | Pre comment | |
| Range: | null/valid string |
| postComment | Post comment | |
| Range: | null/valid string |
| startQac | start of QAC string | |
| Range: | null/valid string |
| endQac | end of QAC string | |
| Range: | null/valid string |
| name | Name of define statement or struct | |
| Range: | null/valid string |
| type | Type of a struct | |
| Range: | null/valid string |
| level | Value of define statement | |
| Range: | null/valid string |
| hasInstanceSetting | Level of inner struct | |
| Range: | 0/valid integer |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Struct generation item constructor | | |

{Ref: N/A}

#### Generate

GENERIC\_GDD\_CLS\_011\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | genSettings | generation setting | |
| Range: | null/valid GenerationSettings |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate output string of a struct  Algorithm: LET stringBuilder = new StringBuilder()  LET indentStr = (new StringBuilder()).Insert(Constant.INT\_ZERO, " ", Level).ToString()   CALL AppendLine(stringBuilder, string.Empty, new string[] { PreComment }) to add pre comment  CALL AppendLine(stringBuilder, string.Empty, StartQAC) to add start of QAC  IF Name is not null AND Type is not null THEN  LET instSetting = (genSettings != null) ? genSettings.InstanceSetting : null  LET nameOfInstance = Name  IF HasNameInstanceSetting is true AND instSetting is not null AND instSetting.IsMultiInstance is true  THEN  IF Name is matched with @"(\w+)(.\*\[.\*\])?" THEN  nameOfInstance = String.Format("{0}{1}{2}",  match.Groups[Constant.INT\_ONE].ToString(),  instSetting.GetInstanceSuffix(),  match.Groups[Constant.INT\_TWO].ToString())  ELSE  Do nothing  ELSE  Do nothing  CALL stringBuilder.AppendLine(WrapText(indentStr + String.Format("{0} {1} =", Type, nameOfInstance)))  ELSE  Do nothing  FOREACH childItem IN noNullChilds WHERE childItem is not null  IF childItem does not equal to lastItem THEN  CALL childItem.SetLastItem(true)  ELSE  Do nothing  CALL childItem.SetLevel(Level + CHILD\_LEVEL)  CALL stringBuilder.AppendLine WITH childItem.Generate(genSettings)  CALL AppendLine(stringBuilder, indentStr, EndQAC) to append end of QAC  CALL AppendLine(stringBuilder, indentStr, new string[] { PostComment }) to append post comment   RETURN stringBuilder.ToString().TrimEnd() | | |

{Ref: N/A}

#### genFieldInStructure

GENERIC\_GDD\_CLS\_011\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | genFieldInStructure | | |
| **Arguments** | item | None | |
| Range: | None |
| qacMapping | QAC message mapping | |
| Range: | null/valid string |
| **Return** | BaseGenerationItem |  | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Generate field in a structure  Algorithm: Generate a field in a structure with comment if this field has value in intermediate data  IF (item != null) THEN  CALL getQacMessage WITH item.Name, qacMapping THEN add result into qacMessage  LET string preComment = "<item.Name>"  LET BaseGenerationItem ret = StructMemberGenerationItem(preComment,  string.Empty, qacMessage[0], qacMessage[1], value)  ELSE  LET BaseGenerationItem ret = new BreakLineGenerationItem();  RETURN ret | | |

{Ref: N/A}

#### genMemorySection

GENERIC\_GDD\_CLS\_011\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | genMemorySection | | |
| **Arguments** | structName | struct name | |
| Range: | null/valid string |
| memorySection | Parameter to get QAC message | |
| Range: | null/valid string |
| includeMem | The include memmap | |
| Range: | null/valid string |
| MemorySectionMapping | Memory section mapping | |
| Range: | null/valid dictionary |
| **Return** | List<BaseGenerationItem> |  | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Generate data for memory section based on struct name  Algorithm: LET startSection = String.Empty  LET stopSection = String.Empty  LET moduleName = basicConfig.ModuleName.ToUpper()  LET qacMessage = CALL getQacMessage(includeMem, qacMapping)  LET ret = new List<BaseGenerationItem>()  FOREACH memSection IN Keys of MemorySectionMapping  LET startSection = $"{moduleName}\_START\_SEC\_{memSection}"  LET stopSection = $"{moduleName}\_STOP\_SEC\_{memSection}"  LET structNames = MemorySectionMapping[memSection]  IF memorySection is START\_SECTION AND structName is first item of structNames THEN  CALL new DefineGenerationItem(string.Empty, string.Empty, String.Empty, String.Empty, startSection, string.Empty))  then ADD to ret  CALL new StringGenerationItem(includeMem, qacMessage[0], qacMessage[1]) then ADD to ret  CALL new BreakLineGenerationItem() then ADD to ret  ELSE IF memorySection is END\_SECTION AND structName is last item of structNames THEN  CALL new DefineGenerationItem(string.Empty, string.Empty, String.Empty, String.Empty, stopSection, string.Empty))  then ADD to ret  CALL new StringGenerationItem(includeMem, qacMessage[0], qacMessage[1]) then ADD to ret  CALL new BreakLineGenerationItem() then ADD to ret  ELSE  Do nothing  RETURN ret | | |

{Ref: N/A}

#### genSingleStruct

GENERIC\_GDD\_CLS\_011\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | genSingleStruct | | |
| **Arguments** | child | None | |
| Range: | None |
| index | Index of struct item | |
| Range: | 0/valid integer |
| qacMapping | QAC message mapping | |
| Range: | null/valid string |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Generate a struct that includes only single item  Algorithm: LET childStruct = new StructGenerationItem(  $"Index: {index} - {child.Name}",  string.Empty, String.Empty, String.Empty, string.Empty, string.Empty)  FOREACH subChild IN child.Childs  IF subChild.Childs.Count is greater than 0 THEN  CALL genSingleStruct(subChild, indexSubChild, qacMapping) then ADD to childStruct  ELSE  CALL genFieldInStructure(subChild, qacMapping) then ADD to childStruct  IF indexSubChild != (child.Childs.Count - 1) THEN  CALL new BreakLineGenerationItem() then ADD to childStruct  ELSE  Do nothing  RETURN childStruct | | |

{Ref: N/A}

#### getQacMessage

GENERIC\_GDD\_CLS\_011\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getQacMessage | | |
| **Arguments** | param | Parameter to get QAC message | |
| Range: | null/valid string |
| qacMapping | QAC mapping | |
| Range: | Null/!=Null |
| **Return** | string[] | QAC message of parameter | |
| Range: | Null/!=Null |
| **Generated Value** | None | | |
| **Description** | Update QAC message  Algorithm: LET ret = init array with size is 2  LET qacCode = value of param in dictionary QacMapping  LET startQacMessage = new List<string>()  LET endQacMessage = startQacMessage = new List<string>()  FOREACH qacCode IN qacCodes  LET qacLevel = first number of qacCode  IF qacLevel > 4 THEN  CALL string.Format(Resources.StartQac, Resources.MisraPrefix, qacCode) then ADD to startQacMessage  CALL string.Format(Resources.EndQac, qacCode) then ADD to endQacMessage  ELSE  CALL string.Format(Resources.StartQac, Resources.QacPrefix, qacCode) then ADD to startQacMessage  CALL string.Format(Resources.EndQac, qacCode) then ADD to endQacMessage  CALL endQacMessage.Reverse()  LET ret[0] = string.Join(Environment.NewLine, startQacMessage)  LET ret[1] = string.Join(Environment.NewLine, endQacMessage)   RETURN ret | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::VariableGenerationItem

|  |  |
| --- | --- |
| **Location File Name** | VariableGenerationItem.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | VariableGenerationItem | Public | To generate ARRAY variables |
| (2) | Generate | Private | Generate output string of a variable |

### Fields

### Methods

#### VariableGenerationItem

GENERIC\_GDD\_CLS\_012\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | VariableGenerationItem | | |
| **Arguments** | preComment | Pre comment | |
| Range: | null/valid string |
| postComment | Post comment | |
| Range: | null/valid string |
| startQac | start of QAC string | |
| Range: | null/valid string |
| endQac | end of QAC string | |
| Range: | null/valid string |
| dataType | data type | |
| Range: | null/valid string |
| name | Name of define statement or struct | |
| Range: | null/valid string |
| value | Value of define statement | |
| Range: | null/valid string |
| hasNameInstanceSetting | Instance setting | |
| Range: | true/false |
| hasValueInstanceSetting | Instance setting | |
| Range: | true/false |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | To generate ARRAY variables | | |

{Ref: N/A}

#### Generate

GENERIC\_GDD\_CLS\_012\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | genSettings | generation setting | |
| Range: | null/valid GenerationSettings |
| **Return** | string | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate output string of a variable  Algorithm: LET stringBuilder = new StringBuilder()  LET indentStr = (new StringBuilder()).Insert(Constant.INT\_ZERO, " ", Level).ToString()   CALL AppendLine(stringBuilder, string.Empty, new string[] { PreComment }) to add pre comment  CALL AppendLine(stringBuilder, string.Empty, StartQAC) to add start of QAC  IF Name is not null THEN  LET instSetting = (genSettings != null) ? genSettings.InstanceSetting : null  LET nameOfInstance = Name  IF HasNameInstanceSetting is true AND instSetting is not null AND instSetting.IsMultiInstance is true  THEN  IF Name is matched with @"(\w+)(.\*\[.\*\])?" THEN  nameOfInstance = String.Format("{0}{1}{2}",  match.Groups[Constant.INT\_ONE].ToString(),  instSetting.GetInstanceSuffix(),  match.Groups[Constant.INT\_TWO].ToString())  ELSE  Do nothing  ELSE  Do nothing  LET variable = string.Format("{0} {1}", Type, nameOfInstance)  ELSE  Do nothing  IF Value is not null THEN  LET valueOfInstance = Value  IF HasNameInstanceSetting is true AND instSetting is not null AND instSetting.IsMultiInstance is true  THEN  IF Value is matched with @"(\w+)(.\*\[.\*\])?" THEN  valueOfInstance = String.Format("{0}{1}{2}",  match.Groups[Constant.INT\_ONE].ToString(),  instSetting.GetInstanceSuffix(),  match.Groups[Constant.INT\_TWO].ToString())  ELSE  Do nothing  ELSE  Do nothing  LET variable = string.Format("{0} = {1}", variable, valueOfInstance)   CALL AppendLine(stringBuilder, indentStr, EndQAC) to append end of QAC  CALL AppendLine(stringBuilder, indentStr, new string[] { PostComment }) to append post comment   RETURN stringBuilder.ToString().TrimEnd() | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::HeaderFileGeneration

|  |  |
| --- | --- |
| **Location File Name** | HeaderFileGeneration.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | HeaderFileGeneration | Protected | Header file generation constructor |
| (2) | GenSTART\_MACRO | Protected | Generate start macro |
| (3) | GenGLOBAL\_SYMBOLS | Protected | Generate global symbols |
| (4) | GenGLOBAL\_DATA\_TYPE | Protected | Generate global data types |
| (5) | GenGLOBAL\_DATA | Protected | Generate global data |
| (6) | GenFUNCTION\_PROTOTYPE | Protected | Generate function prototype |
| (7) | GenEND\_MACRO | Protected | Generate end macro |
| (8) | GenREVISION\_HISTORY | Protected | Generate revision history |

### Fields

### Methods

#### GenEND\_MACRO

GENERIC\_GDD\_CLS\_013\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenEND\_MACRO | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of end macro | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate end macro  Algorithm: Generate end macro. An abstract method overrided by specific module  to provide end macro information | | |

{Ref: N/A}

#### GenFUNCTION\_PROTOTYPE

GENERIC\_GDD\_CLS\_013\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenFUNCTION\_PROTOTYPE | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of function prototype | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate function prototype  Algorithm: Generate function prototype. An abstract method overrided by specific module  to provide function prototype information | | |

{Ref: N/A}

#### GenGLOBAL\_DATA

GENERIC\_GDD\_CLS\_013\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenGLOBAL\_DATA | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of global data | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate global data  Algorithm: Generate global data. An abstract method overrided by specific module  to provide global data information | | |

{Ref: N/A}

#### GenGLOBAL\_DATA\_TYPE

GENERIC\_GDD\_CLS\_013\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenGLOBAL\_DATA\_TYPE | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of global data types | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate global data types  Algorithm: Generate global data types. An abstract method overrided by specific module  to provide global data types information | | |

{Ref: N/A}

#### GenGLOBAL\_SYMBOLS

GENERIC\_GDD\_CLS\_013\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenGLOBAL\_SYMBOLS | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of global symbols | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate global symbols  Algorithm: Generate global symbols. An abstract method overrided by specific module  to provide global symbols information | | |

{Ref: N/A}

#### GenREVISION\_HISTORY

GENERIC\_GDD\_CLS\_013\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenREVISION\_HISTORY | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of revision history | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate revision history  Algorithm: Generate revision history. An abstract method overrided by specific module  to provide revision history information | | |

{Ref: N/A}

#### GenSTART\_MACRO

GENERIC\_GDD\_CLS\_013\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenSTART\_MACRO | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of start macro | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate start macro  Algorithm: Generate start macro from template | | |

{Ref: N/A}

#### HeaderFileGeneration

GENERIC\_GDD\_CLS\_013\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | HeaderFileGeneration | | |
| **Arguments** | fileName | File name | |
| Range: | null/valid string |
| outputDirectory | Output directory | |
| Range: | null/valid string |
| logger | Logger | |
| Range: | null/valid ILogger |
| basicConfiguration | Basic configuration | |
| Range: | null/valid IBasicConfiguration |
| runtimeConfiguration | Runtime configuration | |
| Range: | null/valid IRuntimeConfiguration |
| intermediateData | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Header file generation constructor | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::ItemGenerationAttribute

|  |  |
| --- | --- |
| **Location File Name** | SectionGenAttribute.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | SectionName | Public | This enum is used to declare where output is generated. |
| (2) | Section | Public | None |
| (3) | SectionOrder | Public | None |
| (4) | sectionString | Private | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetSectionName | Public | Get section name |
| (2) | GetSectionName | Public | Get section name |

### Fields

#### SectionName

GENERIC\_GDD\_CLS\_014\_001:

|  |  |
| --- | --- |
| **Variable Name** | SectionName |
| **Type** | None |
| **Range** | None |
| **Description** | This enum is used to declare where output is generated. |

#### Section

GENERIC\_GDD\_CLS\_014\_002:

|  |  |
| --- | --- |
| **Variable Name** | Section |
| **Type** | SectionName |
| **Range** | None |
| **Description** | None |

#### SectionOrder

GENERIC\_GDD\_CLS\_014\_003:

|  |  |
| --- | --- |
| **Variable Name** | SectionOrder |
| **Type** | float |
| **Range** | None |
| **Description** | None |

#### sectionString

GENERIC\_GDD\_CLS\_014\_004:

|  |  |
| --- | --- |
| **Variable Name** | sectionString |
| **Type** | readonly string [] |
| **Range** | None |
| **Description** | None |

### Methods

#### GetSectionName

GENERIC\_GDD\_CLS\_014\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetSectionName | | |
| **Arguments** | None | | |
| **Return** | string | The section name | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get section name  Algorithm: Get text of section from SectionString array | | |

{Ref: N/A}

#### GetSectionName

GENERIC\_GDD\_CLS\_014\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetSectionName | | |
| **Arguments** | Section | section | |
| Range: | null/valid SectionName |
| **Return** | string | The section name | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get section name  Algorithm: Get text of section from SectionString array | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::NewLineAttribute

|  |  |
| --- | --- |
| **Location File Name** | NewLineAttribute.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | NumberOfNewLines | Public | Number of new lines |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | NewLineAttribute | Public | New line attribute constructor |

### Fields

#### NumberOfNewLines

GENERIC\_GDD\_CLS\_015\_001:

|  |  |
| --- | --- |
| **Variable Name** | NumberOfNewLines |
| **Type** | int |
| **Range** | None |
| **Description** | Number of new lines |

### Methods

#### NewLineAttribute

GENERIC\_GDD\_CLS\_015\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | NewLineAttribute | | |
| **Arguments** | numberOfNewLines | 1 | |
| Range: | Number of new lines |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | New line attribute constructor | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::Section

|  |  |
| --- | --- |
| **Location File Name** | Section.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | title | Private | Title |
| (2) | body | Private | Body |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Section | Public | None |
| (2) | Generate | Public | Generate section information |

### Fields

#### title

GENERIC\_GDD\_CLS\_016\_001:

|  |  |
| --- | --- |
| **Variable Name** | title |
| **Type** | string |
| **Range** | None |
| **Description** | Title |

#### body

GENERIC\_GDD\_CLS\_016\_002:

|  |  |
| --- | --- |
| **Variable Name** | body |
| **Type** | string |
| **Range** | None |
| **Description** | Body |

### Methods

#### Generate

GENERIC\_GDD\_CLS\_016\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate section information  Algorithm: Generate section information including title and body | | |

{Ref: N/A}

#### Section

GENERIC\_GDD\_CLS\_016\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Section | | |
| **Arguments** | title | Title | |
| Range: | None |
| body | Body | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::Settings::GenerationSettings

|  |  |
| --- | --- |
| **Location File Name** | GenerationSettings.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | InstanceSetting | Public | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| - | - | - | - |

### Fields

#### InstanceSetting

GENERIC\_GDD\_CLS\_017\_001:

|  |  |
| --- | --- |
| **Variable Name** | InstanceSetting |
| **Type** | InstanceSetting |
| **Range** | None |
| **Description** | None |

### Methods

## Renesas::Generator::MCALConfGen::Business::Generation::Settings::InstanceSetting

|  |  |
| --- | --- |
| **Location File Name** | InstanceSetting.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | IsMultiInstance | Public | None |
| (2) | InstanceIndex | Public | None |
| (3) | instance\_Prefix | Private | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetInstanceSuffix | Public | Return "\_" + InstanceIndex in case module run with multi instance set. |

### Fields

#### IsMultiInstance

GENERIC\_GDD\_CLS\_018\_001:

|  |  |
| --- | --- |
| **Variable Name** | IsMultiInstance |
| **Type** | bool |
| **Range** | None |
| **Description** | None |

#### InstanceIndex

GENERIC\_GDD\_CLS\_018\_002:

|  |  |
| --- | --- |
| **Variable Name** | InstanceIndex |
| **Type** | int |
| **Range** | None |
| **Description** | None |

#### instance\_Prefix

GENERIC\_GDD\_CLS\_018\_003:

|  |  |
| --- | --- |
| **Variable Name** | instance\_Prefix |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

### Methods

#### GetInstanceSuffix

GENERIC\_GDD\_CLS\_018\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetInstanceSuffix | | |
| **Arguments** | None | | |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Return "\_" + InstanceIndex in case module run with multi instance set.  Algorithm: LET ret = string.Empty  IF IsMultiInstance is true THEN  LET ret = String.Format(instance\_Prefix, InstanceIndex)  ELSE  Do nothing  RETURN ret | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Generation::SourceFileGeneration

|  |  |
| --- | --- |
| **Location File Name** | SourceFileGeneration.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | SourceFileGeneration | Protected | Source file generation constructor |
| (2) | GenVERSION\_CHECK | Protected | Generate version checking |
| (3) | GenINSTANCE\_INDEX | Protected | Generate instance index |
| (4) | GenGLOBAL\_DATA\_TYPE | Protected | Generate global data type |
| (5) | GenGLOBAL\_DATA | Protected | Generate global data |
| (6) | GenFUNTION\_DEFINE | Protected | Generate function definitions |
| (7) | GenREVISION\_HISTORY | Protected | Generate revision history |

### Fields

### Methods

#### GenFUNTION\_DEFINE

GENERIC\_GDD\_CLS\_019\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenFUNTION\_DEFINE | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of function definitions | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate function definitions  Algorithm: Generate function definitions. An abstract method overrided by specific module  to provide function definitions information | | |

{Ref: N/A}

#### GenGLOBAL\_DATA

GENERIC\_GDD\_CLS\_019\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenGLOBAL\_DATA | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of global data | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate global data  Algorithm: Generate global data. An abstract method overrided by specific module  to provide global data information | | |

{Ref: N/A}

#### GenGLOBAL\_DATA\_TYPE

GENERIC\_GDD\_CLS\_019\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenGLOBAL\_DATA\_TYPE | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of global data type | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate global data type  Algorithm: Generate global data type. An abstract method overrided by specific module  to provide global data type information | | |

{Ref: N/A}

#### GenINSTANCE\_INDEX

GENERIC\_GDD\_CLS\_019\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenINSTANCE\_INDEX | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of instance index | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate instance index  Algorithm: Generate instance index | | |

{Ref: N/A}

#### GenREVISION\_HISTORY

GENERIC\_GDD\_CLS\_019\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenREVISION\_HISTORY | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of revision history | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate revision history  Algorithm: Generate revision history. An abstract method overrided by specific module  to provide revision history information | | |

{Ref: N/A}

#### GenVERSION\_CHECK

GENERIC\_GDD\_CLS\_019\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenVERSION\_CHECK | | |
| **Arguments** | None | | |
| **Return** | BaseGenerationItem[] | BaseGenerationItem of version checking | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Generate version checking  Algorithm: Generate version check. An abstract method overrided by specific module  to provide version checking information | | |

{Ref: N/A}

#### SourceFileGeneration

GENERIC\_GDD\_CLS\_019\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SourceFileGeneration | | |
| **Arguments** | fileName | File name | |
| Range: | null/valid string |
| outputDirectory | Output directory | |
| Range: | null/valid string |
| logger | Logger | |
| Range: | null/valid ILogger |
| basicConfiguration | Basic configuration | |
| Range: | null/valid IBasicConfiguration |
| runtimeConfiguration | Runtime configuration | |
| Range: | null/valid IRuntimeConfiguration |
| intermediateData | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Source file generation constructor | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Intermediate::BaseIntermediate

|  |  |
| --- | --- |
| **Location File Name** | BaseIntermediate.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Logger | Protected | To record log information for debugging. |
| (2) | Repo | Protected | To provide more convenience operations to upper layers. |
| (3) | Interdata | Protected | To save and retrive data intermediation. |
| (4) | BasicConfiguration | Protected | To store the basic information of gentool module. |
| (5) | GenSettings | Protected | To store all generation settings, e.g. settings of multi-instance. |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ComputeAll | Public | Compute all intermediate data by calling all intermediation methods |
| (2) | BaseIntermediate | Protected | To initialize a new intermediate data instance. |
| (3) | BaseIntermediate | Protected | This is constructor of |
| (4) | Init | Protected | Initialize required attribute. |

### Fields

#### Logger

GENERIC\_GDD\_CLS\_020\_001:

|  |  |
| --- | --- |
| **Variable Name** | Logger |
| **Type** | ILogger |
| **Range** | None |
| **Description** | To record log information for debugging. |

#### Repo

GENERIC\_GDD\_CLS\_020\_002:

|  |  |
| --- | --- |
| **Variable Name** | Repo |
| **Type** | IRepository |
| **Range** | None |
| **Description** | To provide more convenience operations to upper layers. |

#### Interdata

GENERIC\_GDD\_CLS\_020\_003:

|  |  |
| --- | --- |
| **Variable Name** | Interdata |
| **Type** | IIntermediateData |
| **Range** | None |
| **Description** | To save and retrive data intermediation. |

#### BasicConfiguration

GENERIC\_GDD\_CLS\_020\_004:

|  |  |
| --- | --- |
| **Variable Name** | BasicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | To store the basic information of gentool module. |

#### GenSettings

GENERIC\_GDD\_CLS\_020\_005:

|  |  |
| --- | --- |
| **Variable Name** | GenSettings |
| **Type** | GenerationSettings |
| **Range** | None |
| **Description** | To store all generation settings, e.g. settings of multi-instance. |

### Methods

#### ComputeAll

GENERIC\_GDD\_CLS\_020\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ComputeAll | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Compute all intermediate data by calling all intermediation methods  Algorithm: GLOBAL VARIABLE IN:  None   GLOBAL VARIABLE OUT:  None   PRECONDITION:  None   LET methods = null.  LET methods = return value by CALL GetType().GetMethods WITH all non-public methods that have  IntermediateItem attribute and order by line number position.  FOREACH method in methods  LET attribute = return value by CALL GetCustomAttribute WITH  attributeType is IntermediateItemAttribute and inherit is false.  IF attribute is not null THEN  IF type of BaseIntermediateItem is type of method's returnType THEN  CALL intermediate methods to get result  LET store result in intermediate data with DBPath retrieving from IntermediateItem attribute | | |

{Ref: N/A}

#### BaseIntermediate

GENERIC\_GDD\_CLS\_020\_007:

|  |  |
| --- | --- |
| **Method Name** | BaseIntermediate |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | To initialize a new intermediate data instance. |

{Ref: N/A}

#### BaseIntermediate

GENERIC\_GDD\_CLS\_020\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseIntermediate | | |
| **Arguments** | logger | Logger | |
| Range: | null/valid ILogger |
| repo | Repo | |
| Range: | null/valid IRepository |
| interdata | None | |
| Range: | None |
| basicConfig | Basic configuration | |
| Range: | null/valid IBasicConfiguration |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This is constructor of | | |

{Ref: N/A}

#### Init

GENERIC\_GDD\_CLS\_020\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Init | | |
| **Arguments** | None | | |
| **Return** | N/A | N/A | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Initialize required attribute. | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Parser::Parser

|  |  |
| --- | --- |
| **Location File Name** | Parser.cs |
| **Description** | This is abstract class that is used as template method and is concreted implement by other class. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ParseCDFs | Public | This abstract method is used to parse CDF files include CDF, BSW |
| (2) | ParseTranslationFile | Public | This abstract method is used to parse CDF files include Translation File. |
| (3) | LoadFile | Public | This abstract method is used to load CDF files include CDF, BSW, Translation File. |
| (4) | UpdateBasicConfigurationIfNeeded | Public | This abstract method is used to update basic configuration |
| (5) | RunAll | Public | Outline algorithm parse and validate CDFs in Autosar |

### Fields

### Methods

#### LoadFile

GENERIC\_GDD\_CLS\_021\_001:

|  |  |
| --- | --- |
| **Method Name** | LoadFile |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | This abstract method is used to load CDF files include CDF, BSW, Translation File. |

{Ref: N/A}

#### ParseCDFs

GENERIC\_GDD\_CLS\_021\_002:

|  |  |
| --- | --- |
| **Method Name** | ParseCDFs |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | This abstract method is used to parse CDF files include CDF, BSW |

{Ref: N/A}

#### ParseTranslationFile

GENERIC\_GDD\_CLS\_021\_003:

|  |  |
| --- | --- |
| **Method Name** | ParseTranslationFile |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | This abstract method is used to parse CDF files include Translation File. |

{Ref: N/A}

#### RunAll

GENERIC\_GDD\_CLS\_021\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | RunAll | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Outline algorithm parse and validate CDFs in Autosar  Algorithm: GLOBAL VARIABLE IN:  None   GLOBAL VARIABLE OUT:  None   PRECONDITION:  None   LET CALL LoadFile to load XDocuments from CDF file paths.  LET CALL ParseCDFs to parse all CDF and BSW files.  LET CALL ParseTranslationFile to check required modules and parse translation files.  LET CALL UpdateBasicConfigurationIfNeeded to update information used for Object Factory. | | |

{Ref: N/A}

#### UpdateBasicConfigurationIfNeeded

GENERIC\_GDD\_CLS\_021\_005:

|  |  |
| --- | --- |
| **Method Name** | UpdateBasicConfigurationIfNeeded |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | This abstract method is used to update basic configuration |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Parser::ParserXML

|  |  |
| --- | --- |
| **Location File Name** | ParserXML.cs |
| **Description** | None |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | CdfData | Protected | To save and retrive infomration of CDF files. |
| (3) | RuntimeConfiguration | Protected | To define information processed by command line. |
| (4) | BasicConfiguration | Protected | To store basic information of Gentool. |
| (5) | UserInterface | Protected | To record Error, Warning,Info information and exit Gentool. |
| (6) | CdfFiles | Protected | CDF files |
| (7) | Register | Protected | To parses and stores information of translation and device header files. |
| (8) | ParseConfigStatus | Protected | To evaluate status after parser parse configuration from CDF. |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ParseCDFs | Public | Parse CDF and BSW files |
| (2) | ParseTranslationFile | Public | Parse translation file |
| (3) | LoadFile | Public | Init parse files |
| (4) | UpdateBasicConfigurationIfNeeded | Public | Update basic configuration if needed |
| (5) | ParserXML | Protected | To initialize a new |
| (6) | ParserXML | Protected | This is constructor of |
| (7) | GetDeviceName | Protected | To get the device's name. |
| (8) | ParseAndSaveCDFFile | Protected | To parse and save CDF file into cdfData. |
| (9) | ParseConfiguration | Protected | To parse the configuration from CDF files. |
| (10) | ShouldFilterOutConfiguration | Protected | This method is used to filter vendor in MCAL modules based on -fr option from user input. |
| (11) | BuildAndSaveContainerTree | Protected | Build a containers tree of a module and save in cdfData |
| (12) | BuildSubContainers | Protected | Build the sub containers |
| (13) | ConvertToContainer | Protected | Convert XElement into a Container instance |
| (14) | CheckAutosarVersion | Protected | Check autosar version of BSWMDT File and Module Description File. |
| (15) | CheckParamConfig | Protected | Check required parameter in bsw file. |
| (16) | ParseAndSaveBswConfiguration | Protected | Parse BSW configuration and save into cdfData Tags need parse: IMPLEMENT-SHORT-NAME IMPLEMENT-PROGRAMMING-LANGUAGE IMPLEMENT-SW-VERSION IMPLEMENT-VENDOR-ID IMPLEMENT-AR-RELEASE-VERSION IMPLEMENT-VENDOR-API-INFIX IMPLEMENT-BEHAVIOR-REF IMPLEMENT-UUID DESCRIPTION-SHORT-NAME DESCRIPTION-MODULE-ID DESCRIPTION-UUID IMPLEMENT-VENDOR-SPECIFIC-MODULE-DEF-REFS |
| (17) | CheckERR000025 | Protected | This method check path in bsw mdt with path that defined by Configuration in CDF file |
| (18) | CheckRelationBswmdtAndCdf | Protected | This method check relation between BSW and CDF files. It raise error ERR000048 if not any relation between BSW and CDF |
| (19) | ParseSubElement | Protected | Parse the sub-element |
| (20) | ValidateBswConfigurationDetail | Protected | Validate detail of basic software configuration |
| (21) | ValidateBswConfigurationsGeneral | Protected | Validate rules required for all BSW configuration |
| (22) | ParseThruParameters | Protected | Parse XElement and return a list of parameters |
| (23) | ParseThruReferences | Protected | Parse through References |
| (24) | ParseItemValue | Protected | Parse item value from XElement. |
| (25) | ParseTransFileSection | Protected | Parse translation file section defined by tag TRANSLATION-FILE-PATH |
| (26) | ParseDeviceFileSection | Protected | Parse device file section defined by tag DEVICE-FILE-PATH |
| (27) | GetRootPath | Protected | Get root path of a CDF/BSW file |
| (28) | parseContent | Private | Parse content in bsw file. |
| (29) | convertValueFromString | Private | Convert value of tags in CDF to value that has corresponding data type |
| (30) | convertToBool | Private | Convert value string to value that has bool data type |
| (31) | convertToLong | Private | Convert value string to value that has long data type |
| (32) | convertToDouble | Private | Convert value string to value that has double data type |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_022\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly ParserXML |
| **Range** | None |
| **Description** | None |

#### CdfData

GENERIC\_GDD\_CLS\_022\_002:

|  |  |
| --- | --- |
| **Variable Name** | CdfData |
| **Type** | ICdfData |
| **Range** | None |
| **Description** | To save and retrive infomration of CDF files. |

#### RuntimeConfiguration

GENERIC\_GDD\_CLS\_022\_003:

|  |  |
| --- | --- |
| **Variable Name** | RuntimeConfiguration |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | To define information processed by command line. |

#### BasicConfiguration

GENERIC\_GDD\_CLS\_022\_004:

|  |  |
| --- | --- |
| **Variable Name** | BasicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | To store basic information of Gentool. |

#### UserInterface

GENERIC\_GDD\_CLS\_022\_005:

|  |  |
| --- | --- |
| **Variable Name** | UserInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | To record Error, Warning,Info information and exit Gentool. |

#### CdfFiles

GENERIC\_GDD\_CLS\_022\_006:

|  |  |
| --- | --- |
| **Variable Name** | CdfFiles |
| **Type** | Dictionary< string, XDocument > |
| **Range** | None |
| **Description** | CDF files |

#### Register

GENERIC\_GDD\_CLS\_022\_007:

|  |  |
| --- | --- |
| **Variable Name** | Register |
| **Type** | IRegisterProcessing |
| **Range** | None |
| **Description** | To parses and stores information of translation and device header files. |

#### ParseConfigStatus

GENERIC\_GDD\_CLS\_022\_008:

|  |  |
| --- | --- |
| **Variable Name** | ParseConfigStatus |
| **Type** | List< Tuple< string, ParseConfigStatus |
| **Range** | None |
| **Description** | To evaluate status after parser parse configuration from CDF. |

### Methods

#### LoadFile

GENERIC\_GDD\_CLS\_022\_009:

|  |  |
| --- | --- |
| **Method Name** | LoadFile |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Init parse files  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   FOREACH file path in runtimeConfiguration.CDFFilePaths  LET Load XDocument from the file path to store in cdfFiles  LET Handle error ERR000012 if can NOT read files or files have invalid XML format |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_012>}

{Ref: [3] < CMN\_GDD\_ERR\_012>}

#### ParseCDFs

GENERIC\_GDD\_CLS\_022\_010:

|  |  |
| --- | --- |
| **Method Name** | ParseCDFs |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Parse CDF and BSW files  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   FOREACH file in cdfFiles  IF file is CDF file THEN  LET CALL method parseAndSaveCDFFile to parse all CDF file and save a tree of containers to cdfDatas  LET After process CDF files, call method parseAndSaveBswConfiguration to parse BSW configuration file  LET CALL method validateBswConfigurationsGeneral to validate parsed BSW configuration file |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_040 GENERIC\_TUM\_U2X\_GT\_ERR\_047>}

{Ref: [3] < CMN\_GDD\_ERR\_040 CMN\_GDD\_ERR\_047>}

#### ParseTranslationFile

GENERIC\_GDD\_CLS\_022\_011:

|  |  |
| --- | --- |
| **Method Name** | ParseTranslationFile |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Parse translation file  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET Read translation file from runtimeConfiguration.TranslationFilePath  IF file NOT found or invalid XML format THEN  LET Record error ERR000003 and exit  LET GET device name configured in containers of current module (specfied by basicConfiguration.ModuleName)  IF device name is null or empty THEN  LET Record error ERR000039 and exit  LET CAll method parseTransFileSection to parse TRANSLATION-FILE-PATH section  LET CAll method parseDeviceFileSection to parse DEVICE-FILE-PATH section   IF result of parseTransFileSection or parseDeviceFileSection is FALSE THEN  LET Record error ERR000013 and exit |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_003 GENERIC\_TUM\_U2x\_GT\_ERR\_013 GENERIC\_TUM\_U2x\_GT\_ERR\_039 GENERIC\_TUM\_U2x\_GT\_ERR\_046>}

{Ref: [3] < CMN\_GDD\_ERR\_003 CMN\_GDD\_ERR\_013 CMN\_GDD\_ERR\_039

CMN\_GDD\_ERR\_046>}

#### UpdateBasicConfigurationIfNeeded

GENERIC\_GDD\_CLS\_022\_012:

|  |  |
| --- | --- |
| **Method Name** | UpdateBasicConfigurationIfNeeded |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Update basic configuration if needed  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   FOREACH container in cdfData of current module (specified by basicConfiguration.ModuleName)  IF found parameter DeviceName and its value is NOT null or empty THEN  LET Add device name to collection "result"  IF having device names in collection "result" THEN  LET set collection "result" to basicConfiguration.DeviceNames  LET set default auto version 4.2.2 to basicConfiguration.AutoSARVersion |

{Ref: N/A}

#### BuildAndSaveContainerTree

GENERIC\_GDD\_CLS\_022\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BuildAndSaveContainerTree | | |
| **Arguments** | file | CDF file in XDocument | |
| Range: | None |
| module | Module in XElement | |
| Range: | None |
| config | Configuration of module | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Build a containers tree of a module and save in cdfData  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET get all containers in module by getting all elements inside tag CONTAINERS  FOREACH element in container  LET recursively build a container tree where this container is root and all of its children  LET add container short name and container tree to cdfData | | |

{Ref: N/A}

#### BuildSubContainers

GENERIC\_GDD\_CLS\_022\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BuildSubContainers | | |
| **Arguments** | stack | CDF file in XDocument | |
| Range: | None |
| container | Module in XElement | |
| Range: | None |
| sub\_container | Configuration of module | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Build the sub containers  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET get all containers in module by getting all elements inside tag CONTAINERS  FOREACH element container  LET recursively build a container tree where this container is root and all of its children  LET add container short name and container tree to cdfData | | |

{Ref: N/A}

#### CheckAutosarVersion

GENERIC\_GDD\_CLS\_022\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckAutosarVersion | | |
| **Arguments** | bswConfig | Elements | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Check autosar version of BSWMDT File and Module Description File.  Algorithm: GLOBAL VARIABLE IN:  CdfFiles, UserInterface   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   FOREACH file in CdfFiles.Values  LET isAutoSar = checking AR version between CDF data and BSW file  IF isAutoSar is true and bswConfig.ArReleaseVersion is not null and  bswConfig.ArReleaseVersion is not 4.2.2 THEN  LET raise error ERR000029 | | |

{Ref: N/A}

#### CheckERR000025

GENERIC\_GDD\_CLS\_022\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckERR000025 | | |
| **Arguments** | bsw | BSW Document - XML | |
| Range: | None |
| configs | List of configurations | |
| Range: | None |
| bswConfig | Information of bsw mdt | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This method check path in bsw mdt with path that defined by Configuration in CDF file  Algorithm: GLOBAL VARIABLE IN:  UserInterface   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET get path1 in bsw mdt  LET get path2 in ModuleDescriptionRef cdf file  IF path1 not equals path2 THEN  LET raise error ERR000025 | | |

{Ref: N/A}

#### CheckParamConfig

GENERIC\_GDD\_CLS\_022\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckParamConfig | | |
| **Arguments** | bswConfig | Elements | |
| Range: | None |
| requiredFields | Required fields | |
| Range: | None |
| tagNameToField | Mapping from tag names to fields | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Check required parameter in bsw file.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A    FOREACH parameter is required in bsw file  IF have any value absence THEN  LET raise error ERR000030 | | |

{Ref: N/A}

#### CheckRelationBswmdtAndCdf

GENERIC\_GDD\_CLS\_022\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckRelationBswmdtAndCdf | | |
| **Arguments** | bswConfig | Information of bsw mdt | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This method check relation between BSW and CDF files. It raise error ERR000048 if not any relation between BSW and CDF  Algorithm: GLOBAL VARIABLE IN:  CdfData, UserInterface   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET cdfConfig = CALL CdfData.GetAllInstanceConfigurations WITH BasicConfiguration.ModuleName  IF bswConfig.VendorSpecificModuleDefRef is null THEN  LET raise error ERR000049  ELSE IF cdfConfig is not null and cdfConfig.DefinitionRef not equals bswConfig.VendorSpecificModuleDefRef THEN  LET raise error ERR000048 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_048 GENERIC\_TUM\_U2X\_GT\_ERR\_049>}

{Ref: [3] < CMN\_GDD\_ERR\_049 CMN\_GDD\_ERR\_048>}

#### ConvertToContainer

GENERIC\_GDD\_CLS\_022\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ConvertToContainer | | |
| **Arguments** | elements | Elements | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Convert XElement into a Container instance  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   Generate new uuid by Guid.NewGuid() to replace uuid attribute of elements  Convert XElement into a Container instance with mapping  - UUID attribute to Uuid field  - SHORT-NAME tag to ShortName field  - PARAMETER-VALUES tag to ParameterValues field by calling method parseThruParameters  - REFERENCE-VALUES tag to ReferenceValues field by calling method parseThruReferences | | |

{Ref: N/A}

#### GetDeviceName

GENERIC\_GDD\_CLS\_022\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetDeviceName | | |
| **Arguments** | None | | |
| **Return** | string | The device name | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | To get the device's name.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET result = null.  LET Get all instance Containers in CDF data.  FOREACH container in each instance  IF the value of deviveName is not null or not empty string THEN  LET result = deviveName.  break.  RETURN result. | | |

{Ref: N/A}

#### GetRootPath

GENERIC\_GDD\_CLS\_022\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetRootPath | | |
| **Arguments** | xmlDoc | Xml doc | |
| Range: | None |
| **Return** | string | The root path of xmlDoc | |
| Range: | - |
| **Generated Value** | None | | |
| **Description** | Get root path of a CDF/BSW file  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET get all inner elements of xmlDoc that have name AR-PACKAGE and have inner tag is AR-PACKAGES or ELEMENTS  RETURN root path by concatenating their short name by "/" separator | | |

{Ref: N/A}

#### ParseAndSaveBswConfiguration

GENERIC\_GDD\_CLS\_022\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseAndSaveBswConfiguration | | |
| **Arguments** | bsw | Bsw | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Parse BSW configuration and save into cdfData Tags need parse: IMPLEMENT-SHORT-NAME IMPLEMENT-PROGRAMMING-LANGUAGE IMPLEMENT-SW-VERSION IMPLEMENT-VENDOR-ID IMPLEMENT-AR-RELEASE-VERSION IMPLEMENT-VENDOR-API-INFIX IMPLEMENT-BEHAVIOR-REF IMPLEMENT-UUID DESCRIPTION-SHORT-NAME DESCRIPTION-MODULE-ID DESCRIPTION-UUID IMPLEMENT-VENDOR-SPECIFIC-MODULE-DEF-REFS  Algorithm: Parse bsw configuration, raise ERR000025, ERR000026, ERR000029, ERR000048 or ERR000030 if any | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_025 GENERIC\_TUM\_U2x\_GT\_ERR\_029 GENERIC\_TUM\_U2x\_GT\_ERR\_030>}

{Ref: [3] < CMN\_GDD\_ERR\_025 CMN\_GDD\_ERR\_026 CMN\_GDD\_ERR\_029

CMN\_GDD\_ERR\_030 CMN\_GDD\_ERR\_048>}

#### ParseAndSaveCDFFile

GENERIC\_GDD\_CLS\_022\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseAndSaveCDFFile | | |
| **Arguments** | path | CDF file path | |
| Range: | None |
| file | CDF file in XDocument | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | To parse and save CDF file into cdfData.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET modules = return value by CALL Descendants by file.  FOREACH module in modules, which is wrapped in tag ELEMENTS  LET config = initialize a Configuration.  LET status = return value by CALL ParseConfiguration WITH element is module and config is config.  IF the value of status is success in ParseConfigStatus THEN  LET moduleName = return value by CALL GetElementLastInArray WITH path is DefinitionRef.  IF the MODULE-DESCRIPTION-REF element is not present module specific description file.  LET record error ERR000028.  ELSE  LET count = the number of element by CALL Descendants WITH the name of element is 'CONTAINERS'  IF any element not exists in CONTAINERS element  LET Report error ERR000011.  LET Save mapping module short name - CDF file path into cdfData  LET CALL method buildAndSaveContainerTree to build a container tree of this module and  save into cdfData | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_028 GENERIC\_TUM\_U2x\_GT\_ECU\_002>}

{Ref: [3] < CMN\_GDD\_ERR\_011 CMN\_GDD\_ERR\_028>}

#### ParseConfiguration

GENERIC\_GDD\_CLS\_022\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseConfiguration | | |
| **Arguments** | element | Element | |
| Range: | None |
| config | None | |
| Range: | None |
| **Return** | ParseConfigStatus | The Parse configuration | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | To parse the configuration from CDF files.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET Parse an XElement into a Configuration instance WITH mapping  - UUID attribute to Uuid field  - SHORT-NAME tag to ShortName field  - DEFINITION-REF tag to DefinitionRef field  - MODULE-DESCRIPTION-REF tag to ModuleDescriptionRef field   LET CALL shouldFilterOutConfiguration to check Configuration instance should be filtered out  IF not true THEN  LET Save Configuration instance into cdfData  RETURN Configuration instance  ELSE  RETURN null | | |

{Ref: N/A}

#### ParseDeviceFileSection

GENERIC\_GDD\_CLS\_022\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseDeviceFileSection | | |
| **Arguments** | translationFilePath | Translation file path | |
| Range: | None |
| deviceFileSection | Section DEVICE-FILE-PATH | |
| Range: | None |
| deviceName | Device name | |
| Range: | None |
| **Return** | bool | To check device name should be present in translation file path | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Parse device file section defined by tag DEVICE-FILE-PATH  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   IF deviceFileSection is null THEN  RETURN false  Try to get device header file path from deviceName tag which is a child of deviceFileSection  LET Record error ERR000016 if there are NO children of deviceFileSection and exit  LET Record error ERR000017 if there are NO deviceName tag in deviceFileSection and exit  IF no any error found THEN  CALL register.SaveMacroAddressValue to save all define macro in device header file  RETURN true | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_016 GENERIC\_TUM\_U2X\_GT\_ERR\_017>}

{Ref: [3] < CMN\_GDD\_ERR\_016 CMN\_GDD\_ERR\_017>}

#### ParseItemValue

GENERIC\_GDD\_CLS\_022\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseItemValue | | |
| **Arguments** | element | Elements | |
| Range: | None |
| definitionKey | Definition key | |
| Range: | None |
| valueKey | Value key | |
| Range: | None |
| **Return** | ItemValue | The value from CDF | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Parse item value from XElement.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET Get value from CDF and convert to value that has corresponding data type  LET Convert and RETURN value to corresponding data type | | |

{Ref: N/A}

#### ParseSubElement

GENERIC\_GDD\_CLS\_022\_027:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseSubElement | | |
| **Arguments** | element | Bsw | |
| Range: | None |
| requiredFields | None | |
| Range: | None |
| tagNameToField | None | |
| Range: | None |
| bswConfig | None | |
| Range: | None |
| name | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Parse the sub-element  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   FOREACH subElement in element.Elements  LET implement = Parse for Bsw implementation that is in BSW-IMPLEMENTATION tag  LET description = Parse for Bsw Module Description that is in BSW-MODULE-DESCRIPTION tag  IF subElement.Name.LocalName is not null and subElement.Name.LocalName is SHORT-NAME THEN  LET name = subElement.Value  LET parse values from tags in BSW-IMPLEMENTATION tag  LET parse values from tags in BSW-MODULE-DESCRIPTION tag | | |

{Ref: N/A}

#### ParseThruParameters

GENERIC\_GDD\_CLS\_022\_028:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseThruParameters | | |
| **Arguments** | element | Elements | |
| Range: | None |
| **Return** | IList<ItemValue> | The value references from CDF | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Parse XElement and return a list of parameters  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   FOREACH child item in element  LET Create new ItemValue with fields' value WITH  - DefinitionRef = DEFINITION-REF tag  - Value = VALUE tag  IF VALUE tag = 0 THEN  convert Value to false  IF VALUE tag = 1 THEN  convert Value to true  LET add ItemValue into a collection "result"  RETURN collection "result" | | |

{Ref: N/A}

#### ParseThruReferences

GENERIC\_GDD\_CLS\_022\_029:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseThruReferences | | |
| **Arguments** | element | Elements | |
| Range: | None |
| **Return** | IList<ItemValue> | The value references from CDF | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Parse through References  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   FOREACH child item in element  LET Create new ItemValue with fields' value WITH  - DefinitionRef = DEFINITION-REF tag  - Value = VALUE-REF tag  LET add ItemValue into a collection "result"   RETURN collection "result" | | |

{Ref: N/A}

#### ParseTransFileSection

GENERIC\_GDD\_CLS\_022\_030:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseTransFileSection | | |
| **Arguments** | translationFilePath | Translation file path | |
| Range: | None |
| transFileSection | Section TRANSLATION-FILE-PATH | |
| Range: | None |
| deviceName | Device name | |
| Range: | None |
| **Return** | bool | To check device name should be present in translation file path | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Parse translation file section defined by tag TRANSLATION-FILE-PATH  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   IF transFileSection is null THEN  RETURN false  Try to get translation header file path from deviceName tag which is a child of transFileSection  LET Record error ERR000014 if there are NO children of transFileSection and exit  LET Record error ERR000015 if there are NO deviceName tag in transFileSection and exit  IF no any error found THEN  CALL register.SaveMacroLabelValue to save all define macro in translation header file  RETURN true | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_014 GENERIC\_TUM\_U2X\_GT\_ERR\_015>}

{Ref: [3] < CMN\_GDD\_ERR\_014 CMN\_GDD\_ERR\_015>}

#### ParserXML

GENERIC\_GDD\_CLS\_022\_031:

|  |  |
| --- | --- |
| **Method Name** | ParserXML |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | To initialize a new  Algorithm: N/A |

{Ref: N/A}

#### ParserXML

GENERIC\_GDD\_CLS\_022\_032:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParserXML | | |
| **Arguments** | cdfData | Cdf data | |
| Range: | None |
| runtimeConfiguration | Runtime configuration | |
| Range: | None |
| basicConfiguration | Basic configuration | |
| Range: | None |
| register | Register | |
| Range: | None |
| userInterface | User interface | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This is constructor of  Algorithm: Contructor with:  - CdfData  - RuntimeConfiguration  - BasicConfiguration  - Register  - UserInterface | | |

{Ref: N/A}

#### ShouldFilterOutConfiguration

GENERIC\_GDD\_CLS\_022\_033:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ShouldFilterOutConfiguration | | |
| **Arguments** | config | Config | |
| Range: | None |
| **Return** | bool | true: Ignore data from vendor false: Get data from vendor | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | This method is used to filter vendor in MCAL modules based on -fr option from user input.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET result = true  IF configuration should be filtered by conditions  - moduleName extracted from config.DefinitionRef is NOT one of MCAL modules and  - DefinitionRef value of configuration is NOT started with "/" + runtimeConfiguration.VendorFilter  THEN  result = false  RETURN result | | |

{Ref: N/A}

#### ValidateBswConfigurationDetail

GENERIC\_GDD\_CLS\_022\_034:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidateBswConfigurationDetail | | |
| **Arguments** | bswConfig | Bsw configuration | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Validate detail of basic software configuration  Algorithm: GLOBAL VARIABLE IN:  BasicConfiguration   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   IF (1) bswConfig.DescriptionShortName is null or  (2) bswConfig.DescriptionShortName NOT start with basicConfiguration.ModuleName THEN  LET Record error ERR000026 and exit | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_026>}

#### ValidateBswConfigurationsGeneral

GENERIC\_GDD\_CLS\_022\_035:

|  |  |
| --- | --- |
| **Method Name** | ValidateBswConfigurationsGeneral |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Validate rules required for all BSW configuration  Algorithm: GLOBAL VARIABLE IN:  CdfData, BasicConfiguration   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET BswConfigurations from CdfData WITH DescriptionShortName it not null and DescriptionShortName  startswith ModuleName  IF BswConfigurations is null THEN  LET raise error ERR000036 and exit |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_036>}

{Ref: [3] < CMN\_GDD\_ERR\_036>}

#### convertToBool

GENERIC\_GDD\_CLS\_022\_036:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | convertToBool | | |
| **Arguments** | value | Value | |
| Range: | None |
| **Return** | bool | The number have been converted | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Convert value string to value that has bool data type  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET convert values such as: 0 -> false, 1 -> true | | |

{Ref: N/A}

#### convertToDouble

GENERIC\_GDD\_CLS\_022\_037:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | convertToDouble | | |
| **Arguments** | value | Value | |
| Range: | None |
| **Return** | double | The number have been converted | |
| Range: | double.MinValue..double.MaxValue |
| **Generated Value** | None | | |
| **Description** | Convert value string to value that has double data type  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET convert values such as: NAN, -INF, INF -> double | | |

{Ref: N/A}

#### convertToLong

GENERIC\_GDD\_CLS\_022\_038:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | convertToLong | | |
| **Arguments** | value | Value | |
| Range: | None |
| **Return** | object | The number have been converted | |
| Range: | Int64.MinValue..Int64.MaxValue |
| **Generated Value** | None | | |
| **Description** | Convert value string to value that has long data type  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET convert values such as:  [+|-]?[1-9][0-9]\*  0x[0-9a-f]+  0[0-7]\*  0b[0-1]+ | | |

{Ref: N/A}

#### convertValueFromString

GENERIC\_GDD\_CLS\_022\_039:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | convertValueFromString | | |
| **Arguments** | dataType | None | |
| Range: | None |
| dataValue | None | |
| Range: | None |
| **Return** | object | The number have been converted | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Convert value of tags in CDF to value that has corresponding data type  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET convertMap = dictionary WITH key is Type and value is Func<>  IF convertMap containsKey dataType THEN  RETURN value of dataType in convertMap  ELSE  RETURN dataValue  Convert to value that has corresponding data type | | |

{Ref: N/A}

#### parseContent

GENERIC\_GDD\_CLS\_022\_040:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseContent | | |
| **Arguments** | element |  | |
| Range: | Null/Not Null |
| requiredFields |  | |
| Range: | Null/Not Null |
| tagNameToField |  | |
| Range: | Null/Not Null |
| prefixTag |  | |
| Range: | Null/Not Null |
| bswConfig |  | |
| Range: | Null/Not Null |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Parse content in bsw file.  Algorithm: GLOBAL VARIABLE IN:  BasicConfiguration   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   IF element is not Null THEN  LET attribute = return value by CALL Attribute WITH name is UUID.  FOEACH e in element of element  LET field = null.  IF the return value by CALL tagNameToField.TryGetValue WITH  "prefixTag + LocalName of e" and field THEN  LET property = return value by CALL GetProperty of GetType of bswConfig.  IF attribute is not null THEN  IF prefixTag is IMPLEMENT\_PREFIX THEN  LET ImplementUuid of bswConfig = value of attribute  ELSE  LET DescriptionUuid of bswConfig = value of attribute | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Validation::BaseValidation

|  |  |
| --- | --- |
| **Location File Name** | BaseValidation.cs |
| **Description** | This |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Logger | Protected | To record log information for debugging. |
| (2) | UserInterface | Protected | To record Error, Warning, Info information and exit Gentool. |
| (3) | Repository | Protected | To provide more convenience operations to upper layers. |
| (4) | BasicConfiguration | Protected | To store the basic information of gentool module. |
| (5) | IntermediateData | Protected | To save and retrive data intermediation. |
| (6) | ForceBothPreAndPostValidation | Protected | Force both pre and post validation |
| (7) | ForceAllContinueIfFailed | Protected | Force all continue if failed |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ValidatePostIntermediate | Public | Validate post intermediate |
| (2) | ValidatePreIntermediate | Public | Validate pre intermediate |
| (3) | BaseValidation | Protected | To initialize a new baseValidation data instance. |
| (4) | BaseValidation | Protected | This is constructor of basevalidation. |
| (5) | BaseValidation | Protected | This is constructor of basevalidation. |
| (6) | CheckE\_INT\_INCONSISTENT | Protected | To check module E\_INT\_INCONSISTENT |
| (7) | CheckForMandatoryParameters | Protected | Check mandatory parameters |
| (8) | CheckSameFieldsOfSubstructs | Protected | Check same field of substructs |
| (9) | CheckSameFieldsOfSubstructs | Protected | Check same fields of substructs |
| (10) | CheckRequiredFieldsOfSubstructs | Protected | Check required fields of substructs |
| (11) | CheckRequiredFieldsOfSubstructs | Protected | Check required fields of substructs |
| (12) | Check\_ERR\_SPECIFIC\_01 | Protected | To check number of fields in entity structure name |
| (13) | Check\_ERR\_SPECIFIC\_02 | Protected | To check empty field name in entity structure name |
| (14) | GetAllStructureData | Protected | Get all structure data name, an abstract method to be override by specific module to provide structure data in post build source |
| (15) | CheckRequiredModules | Protected | Check required modules |
| (16) | run | Private | Run validation |
| (17) | addErrorMessage | Private | Process error message for checking mandatory parameters |
| (18) | isMissingContainer | Private | Check the availability of container |
| (19) | checkSameFieldsOfSubstructs | Private | Check same fields of substructs |
| (20) | findFieldsOfAnyLastNodeStruct | Private | Find sample fields of any last node struct |
| (21) | checkRequiredFieldsOfSubstructs | Private | Find sample fields of any last node struct |
| (22) | exitWithType | Private | Exit with type |
| (23) | isExistModuleName | Private | Check if moduleName existed in cdfData |

### Fields

#### Logger

GENERIC\_GDD\_CLS\_023\_001:

|  |  |
| --- | --- |
| **Variable Name** | Logger |
| **Type** | ILogger |
| **Range** | None |
| **Description** | To record log information for debugging. |

#### UserInterface

GENERIC\_GDD\_CLS\_023\_002:

|  |  |
| --- | --- |
| **Variable Name** | UserInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | To record Error, Warning, Info information and exit Gentool. |

#### Repository

GENERIC\_GDD\_CLS\_023\_003:

|  |  |
| --- | --- |
| **Variable Name** | Repository |
| **Type** | IRepository |
| **Range** | None |
| **Description** | To provide more convenience operations to upper layers. |

#### BasicConfiguration

GENERIC\_GDD\_CLS\_023\_004:

|  |  |
| --- | --- |
| **Variable Name** | BasicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | To store the basic information of gentool module. |

#### IntermediateData

GENERIC\_GDD\_CLS\_023\_005:

|  |  |
| --- | --- |
| **Variable Name** | IntermediateData |
| **Type** | IIntermediateData |
| **Range** | None |
| **Description** | To save and retrive data intermediation. |

#### ForceBothPreAndPostValidation

GENERIC\_GDD\_CLS\_023\_006:

|  |  |
| --- | --- |
| **Variable Name** | ForceBothPreAndPostValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | Force both pre and post validation |

#### ForceAllContinueIfFailed

GENERIC\_GDD\_CLS\_023\_007:

|  |  |
| --- | --- |
| **Variable Name** | ForceAllContinueIfFailed |
| **Type** | bool |
| **Range** | None |
| **Description** | Force all continue if failed |

### Methods

#### ValidatePostIntermediate

GENERIC\_GDD\_CLS\_023\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidatePostIntermediate | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Validate post intermediate  Algorithm: LET CALL Info of Logger WITH format is "Run all post validations starting at {0:d/M/yyyy HH:mm:ss}"  and marameter is current date.  LET CALL run WITH runPre is false.  LET CALL Info of Logger WITH format is "Run all post validations ending at {0:d/M/yyyy HH:mm:ss}"  and marameter is current date. | | |

{Ref: N/A}

#### ValidatePreIntermediate

GENERIC\_GDD\_CLS\_023\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidatePreIntermediate | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Validate pre intermediate  Algorithm: LET CALL Info of Logger WITH format is "Run all prevalidations starting at {0:d/M/yyyy HH:mm:ss}"  and marameter is current date.  LET CALL run WITH runPre is false.  LET CALL Info of Logger WITH format is "Run all prevalidations starting at {0:d/M/yyyy HH:mm:ss}"  and marameter is current date. | | |

{Ref: N/A}

#### BaseValidation

GENERIC\_GDD\_CLS\_023\_010:

|  |  |
| --- | --- |
| **Method Name** | BaseValidation |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | To initialize a new baseValidation data instance. |

{Ref: N/A}

#### BaseValidation

GENERIC\_GDD\_CLS\_023\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseValidation | | |
| **Arguments** | logger | Logger | |
| Range: | null/valid ILogger |
| userInterface | User interface | |
| Range: | null/valid IUserInterface |
| repository | Repo | |
| Range: | null/valid IRepository |
| basicConfiguration | Basic configuration | |
| Range: | null/valid IBasicConfiguration |
| intermediateData | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This is constructor of basevalidation. | | |

{Ref: N/A}

#### BaseValidation

GENERIC\_GDD\_CLS\_023\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseValidation | | |
| **Arguments** | logger | Logger | |
| Range: | null/valid ILogger |
| userInterface | User interface | |
| Range: | null/valid IUserInterface |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This is constructor of basevalidation. | | |

{Ref: N/A}

#### CheckE\_INT\_INCONSISTENT

GENERIC\_GDD\_CLS\_023\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckE\_INT\_INCONSISTENT | | |
| **Arguments** | None | | |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | To check module E\_INT\_INCONSISTENT  Algorithm: LET moduleName = BasicConfiguration.ModuleName  LET moduleNameFirstCap = moduleName WITH only first character in upper case  LET intConsistencyCheck = string.Format("{0}InterruptConsistencyCheck", moduleNameFirstCap)  LET eIntInconsistent = string.Format("{0}\_E\_INT\_INCONSISTENT", moduleName.ToUpper())  LET intCheckItem = CALL Repository.GetParameterValue(moduleName, intConsistencyCheck)  IF intCheckItem is null AND value of intCheckItem is true THEN  LET eIntItem = CALL Repository.GetReferenceValue(moduleName, eIntInconsistent)  IF eIntItem is null OR Value of eIntItem is null THEN  Raise ERR000041 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_041>}

{Ref: [3] < CMN\_GDD\_ERR\_041>}

#### CheckForMandatoryParameters

GENERIC\_GDD\_CLS\_023\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckForMandatoryParameters | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| mandatoryParams | Mandatory parameters | |
| Range: | null/valid Hashtable |
| containerDefName |  | |
| Range: | null/valid IList<string> |
| **Return** | List<string> | The error message | |
| Range: | 0/size of List<string> |
| **Generated Value** | None | | |
| **Description** | Check mandatory parameters  Algorithm: LET errorMessages = new List<string>()  IF mandatoryParams is not null THEN  FOREACH defName IN Keys of mandatoryParams  IF mandatoryParams[defName] is List<string> THEN  LET containerList = CALL Repository.GetContainersByDefName(moduleName, defName)  FOREACH container IN containerList  FOREACH param IN paramList  LET paramValue = container.GetParameterValue(param)  LET refValue = container.GetReferenceValue(param)  IF paramValue is null OR refValue is null THEN  LET message = addErrorMessage(moduleName, param, defName, container.Path)  ADD message to errorMessages  RETURN errorMessages | | |

{Ref: N/A}

#### CheckRequiredFieldsOfSubstructs

GENERIC\_GDD\_CLS\_023\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckRequiredFieldsOfSubstructs | | |
| **Arguments** | structName | Struct name | |
| Range: | null/valid string |
| requiredFields | Required fields | |
| Range: | null/valid List<string> |
| **Return** | List<string> | The error message | |
| Range: | 0/size of List<string> |
| **Generated Value** | None | | |
| **Description** | Check required fields of substructs  Algorithm: Default dataPath = "/" + structName  CALL CheckRequiredFieldsOfSubstructs WITH default dataPath, structName and requiredFields  RETURN error list from result of CheckRequiredFieldsOfSubstructs | | |

{Ref: N/A}

#### CheckRequiredFieldsOfSubstructs

GENERIC\_GDD\_CLS\_023\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckRequiredFieldsOfSubstructs | | |
| **Arguments** | dataPath | None | |
| Range: | None |
| structName | Struct name | |
| Range: | null/valid string |
| requiredFields | Required fields | |
| Range: | null/valid List<string> |
| **Return** | List<string> | The error message | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Check required fields of substructs  Algorithm: Get BaseIntermediateItem structData from dataPath  CALL checkRequiredFieldsOfSubstructs WITH structName, structData and requiredFields  RETURN error list from result of checkRequiredFieldsOfSubstructs | | |

{Ref: N/A}

#### CheckRequiredModules

GENERIC\_GDD\_CLS\_023\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckRequiredModules | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Check required modules  Algorithm: FOREACH module name in basicConfiguration.ModuleRequired  CALL method isExistModuleName to check if the module name is existed in input CDF files  IF is not existed THEN  LET record absented module name  LET record error ERR[MSN]003 for absented module names if found. | | |

{Ref: N/A}

#### CheckSameFieldsOfSubstructs

GENERIC\_GDD\_CLS\_023\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckSameFieldsOfSubstructs | | |
| **Arguments** | structName | Struct name | |
| Range: | null/valid string |
| **Return** | List<string> | The error message | |
| Range: | 0/size of List<string> |
| **Generated Value** | None | | |
| **Description** | Check same field of substructs  Algorithm: Default dataPath = "/" + structName  CALL CheckSameFieldsOfSubstructs WITH default dataPath and structName  RETURN error list from result of CheckSameFieldsOfSubstructs | | |

{Ref: N/A}

#### CheckSameFieldsOfSubstructs

GENERIC\_GDD\_CLS\_023\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckSameFieldsOfSubstructs | | |
| **Arguments** | dataPath | None | |
| Range: | None |
| structName | Struct name | |
| Range: | null/valid string |
| **Return** | List<string> | The error message | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Check same fields of substructs  Algorithm: LET get a BaseIntermediateItem structData from dataPath  IF structData is NOT null and structData.Childs is NOT null THEN  - CALL method findFieldsOfAnyLastNodeStruct to get fields of an sample struct  - CALL method checkSameFieldsOfSubstructs to check all substructs in structData has the same number of  fields with the sample struct  - RETURN error found by checkSameFieldsOfSubstructs  Else  RETURN empty error list | | |

{Ref: N/A}

#### Check\_ERR\_SPECIFIC\_01

GENERIC\_GDD\_CLS\_023\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Check\_ERR\_SPECIFIC\_01 | | |
| **Arguments** | None | | |
| **Return** | return a |  | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | To check number of fields in entity structure name  Algorithm: Handle error ERRXXX001 | | |

{Ref: N/A}

#### Check\_ERR\_SPECIFIC\_02

GENERIC\_GDD\_CLS\_023\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Check\_ERR\_SPECIFIC\_02 | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | To check empty field name in entity structure name  Algorithm: LET result = initialize a ValidationResult WITH Code is two and ModuleId is specific module id.  LET errorMessages = initialize a list of string.  LET requiredStructData = return value by CALL GetAllStructureData.  IF the value of requiredStructData is not Null THEN  FOREACH structName in each key of requiredStructData  LET add the value by CALL CheckRequiredFieldsOfSubstructs WITH  structuName and requiredStructData[structName] to errorMessages.  IF the number of element of errorMessages is larger than zero THEN  LET Message of result is errorMessages.  LET Type of result is ERROR.  RETURN result. | | |

{Ref: N/A}

#### GetAllStructureData

GENERIC\_GDD\_CLS\_023\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAllStructureData | | |
| **Arguments** | None | | |
| **Return** | return null | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get all structure data name, an abstract method to be override by specific module to provide structure data in post build source  Algorithm: Specific module have to override to provide structure data | | |

{Ref: N/A}

#### addErrorMessage

GENERIC\_GDD\_CLS\_023\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | addErrorMessage | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| param | Parameter | |
| Range: | null/valid string |
| defName | Definition name | |
| Range: | null/valid string |
| path | Path of conainer | |
| Range: | null/valid string |
| **Return** | string | The error message | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Process error message for checking mandatory parameters  Algorithm: LET stringBuilder = initialize a StringBuilder.  LET fileName = return value by CALL GetFullPath of IOWrapper WITH  path is the return value by CALL GetCdfFileName WITH moduleName is moduleName.  LET CALL AppendFormat of stringBuilder WITH  format is "The parameter '{0}' in the container " and arg0 is param.  LET CALL AppendFormat of stringBuilder WITH  format is "'{0}' should be configured.\n" and arg0 is param.  LET CALL AppendFormat of stringBuilder WITH  format is "File Name: {0}\n" and arg0 is param.  LET CALL AppendFormat of stringBuilder WITH  format is "Path: {0}" and arg0 is param.  RETURN stringBuilder | | |

{Ref: N/A}

#### checkRequiredFieldsOfSubstructs

GENERIC\_GDD\_CLS\_023\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkRequiredFieldsOfSubstructs | | |
| **Arguments** | rootStructName | Struct name | |
| Range: | null/valid string |
| structData | None | |
| Range: | None |
| requiredFields | None | |
| Range: | None |
| **Return** | string | The error message | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Find sample fields of any last node struct  Algorithm: Recurively scan substructs in structData  IF a substruct is last node (all items are fields, NOT inner substruct) THEN  IF substruct NOT have values for all requiredFields THEN  LET add an error to collection "result"  RETURN error collection "result" | | |

{Ref: N/A}

#### checkSameFieldsOfSubstructs

GENERIC\_GDD\_CLS\_023\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkSameFieldsOfSubstructs | | |
| **Arguments** | rootStructName | Root struct name | |
| Range: | null/valid string |
| structData | Struct name | |
| Range: | null/valid string |
| fields | Fields | |
| Range: | null/valid List<string> |
| **Return** | string | The error message | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Check same fields of substructs  Algorithm: Recurively scan substructs in structData  IF a substruct is last node (all items are fields, NOT inner substruct) THEN  IF substruct NOT have the same number of field with sample fields list THEN  LET add a error to collection "result"  RETURN error collection "result" | | |

{Ref: N/A}

#### exitWithType

GENERIC\_GDD\_CLS\_023\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | exitWithType | | |
| **Arguments** | type | Error message type | |
| Range: | null/valid string |
| **Return** | return a void | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Exit with type  Algorithm: LET exitCode =  EXIT\_DUE\_INPUT\_FILES\_ERROR if type = ERROR  EXIT\_SUCCESSFUL\_WITH\_WARNINGS if type = WARN  EXIT\_SUCCESSFUL otherwise   CALL userInterface.Exit WITH exitCode | | |

{Ref: N/A}

#### findFieldsOfAnyLastNodeStruct

GENERIC\_GDD\_CLS\_023\_027:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | findFieldsOfAnyLastNodeStruct | | |
| **Arguments** | structData | Struct name | |
| Range: | null/valid string |
| **Return** | string | The error message | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Find sample fields of any last node struct  Algorithm: Recurively scan substructs in structData  IF a substruct is last node (all items are fields, NOT inner substruct) THEN  RETURN the substruct fields list | | |

{Ref: N/A}

#### isExistModuleName

GENERIC\_GDD\_CLS\_023\_028:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | isExistModuleName | | |
| **Arguments** | moduleName | Module name | |
| Range: | None |
| **Return** | bool | Status of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check if moduleName existed in cdfData  Algorithm: LET get all modules from cdfData by CALL method cdfData.GetModules()  IF found a module name that matchs pattern {modulename}{number}\* THEN  RETURN true  ELSE  RETURN false | | |

{Ref: N/A}

#### isMissingContainer

GENERIC\_GDD\_CLS\_023\_029:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | isMissingContainer | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| defName | Definition name | |
| Range: | null/valid string |
| containerDefName |  | |
| Range: | null/valid IList<string> |
| **Return** | bool | The result of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check the availability of container  Algorithm: FOREACH defName in mandatoryParams  LET get list of mandatory parameters in mandatoryParams for this def name  LET get list of containers by this def name  FOREACH container in containerDefName  IF not all the mandatory parameters have value in each container THEN  LET add an error for each missing parameter to collection "result"  RETURN collection "result" | | |

{Ref: N/A}

#### run

GENERIC\_GDD\_CLS\_023\_030:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | run | | |
| **Arguments** | runPre | Pre-validation | |
| Range: | true/false |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Run validation  Algorithm: LET methods = collection of all validation methods that are decorated by ValidationRuleAttribute  FOREACH method IN methods  LET attribute = ValidationRuleAttribute of method  IF PreIntermediationValidation of attribute is runPre AND EnableMethod of attribute is true THEN  LET result = return value of method  IF result is not null AND result is not Error THEN  FOREACH message IN result.Message  Print message and corresponding error code | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Validation::ValidationResult

|  |  |
| --- | --- |
| **Location File Name** | ValidationResult.cs |
| **Description** | This class contain result information of validation such code error, error message, module id, type, success, isError. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | MessageType | Public | This enum is used to declare type of validation result |
| (2) | Code | Public | To define code for error/wrn/inf. |
| (3) | Message | Public | Message list to store the message for error/wrn/inf. |
| (4) | ModuleId | Public | Specific module id, which was defined by AUTOSAR. |
| (5) | Type | Public | Type of the message is error/wrn/inf. |
| (6) | Success | Public | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | IsError | Public | Check if this is an error result |
| (2) | ValidationResult | Public | None |
| (3) | ValidationResult | Public | None |
| (4) | ValidationResult | Public | None |
| (5) | AddMessage | Public | Add new message constructed from message format |

### Fields

#### MessageType

GENERIC\_GDD\_CLS\_024\_001:

|  |  |
| --- | --- |
| **Variable Name** | MessageType |
| **Type** | None |
| **Range** | None |
| **Description** | This enum is used to declare type of validation result |

#### Code

GENERIC\_GDD\_CLS\_024\_002:

|  |  |
| --- | --- |
| **Variable Name** | Code |
| **Type** | int |
| **Range** | None |
| **Description** | To define code for error/wrn/inf. |

#### Message

GENERIC\_GDD\_CLS\_024\_003:

|  |  |
| --- | --- |
| **Variable Name** | Message |
| **Type** | List< string > |
| **Range** | None |
| **Description** | Message list to store the message for error/wrn/inf. |

#### ModuleId

GENERIC\_GDD\_CLS\_024\_004:

|  |  |
| --- | --- |
| **Variable Name** | ModuleId |
| **Type** | int |
| **Range** | None |
| **Description** | Specific module id, which was defined by AUTOSAR. |

#### Type

GENERIC\_GDD\_CLS\_024\_005:

|  |  |
| --- | --- |
| **Variable Name** | Type |
| **Type** | MessageType |
| **Range** | None |
| **Description** | Type of the message is error/wrn/inf. |

#### Success

GENERIC\_GDD\_CLS\_024\_006:

|  |  |
| --- | --- |
| **Variable Name** | Success |
| **Type** | ValidationResult |
| **Range** | None |
| **Description** | None |

### Methods

#### AddMessage

GENERIC\_GDD\_CLS\_024\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AddMessage | | |
| **Arguments** | msgFormat | Message format | |
| Range: | None |
| param | Parameters to fill in message format | |
| Range: | None |
| **Return** | return a void | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Add new message constructed from message format  Algorithm: Add new message constructed from format and parameters to a list | | |

{Ref: N/A}

#### IsError

GENERIC\_GDD\_CLS\_024\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsError | | |
| **Arguments** | None | | |
| **Return** | return a bool | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Check if this is an error result  Algorithm: IF Type is NOT SUCCESS THEN  RETURN true  Else  RETURN false | | |

{Ref: N/A}

#### ValidationResult

GENERIC\_GDD\_CLS\_024\_009:

|  |  |
| --- | --- |
| **Method Name** | ValidationResult |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** |  |

{Ref: N/A}

#### ValidationResult

GENERIC\_GDD\_CLS\_024\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidationResult | | |
| **Arguments** | code | Code | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: N/A}

#### ValidationResult

GENERIC\_GDD\_CLS\_024\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidationResult | | |
| **Arguments** | type | Type | |
| Range: | None |
| code | Code | |
| Range: | None |
| moduleId | Module id | |
| Range: | None |
| msgFormat | Message format | |
| Range: | None |
| param | Parameters to fill in message format | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Business::Validation::ValidationRuleAttribute

|  |  |
| --- | --- |
| **Location File Name** | ValidationRuleAttribute.cs |
| **Description** | This class decorates methods which will be invoked to validate rules |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | PreIntermediationValidation | Public | Flag to validate the process of Gentool is pre-intermetiate. |
| (2) | PostIntermediationValidation | Public | Flag to validate the process of Gentool is post-intermetiate. |
| (3) | ContinueIfFailed | Public | Flag to validate the process of Gentool, continue with any failure. |
| (4) | Order | Public | None |
| (5) | EnableMethod | Public | Flag to enable debugger method |
| (6) | preIntermediationValidation | Private | Flag to validate the process of Gentool is pre-intermetiate. |
| (7) | postIntermediationValidation | Private | Flag to validate the process of Gentool is post-intermetiate. |
| (8) | continueIfFailed | Private | Flag to validate the process of Gentool, continue with any failure. |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ValidationRuleAttribute | Public | None |
| (2) | ValidationRuleAttribute | Public | None |

### Fields

#### PreIntermediationValidation

GENERIC\_GDD\_CLS\_025\_001:

|  |  |
| --- | --- |
| **Variable Name** | PreIntermediationValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | Flag to validate the process of Gentool is pre-intermetiate. |

#### PostIntermediationValidation

GENERIC\_GDD\_CLS\_025\_002:

|  |  |
| --- | --- |
| **Variable Name** | PostIntermediationValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | Flag to validate the process of Gentool is post-intermetiate. |

#### ContinueIfFailed

GENERIC\_GDD\_CLS\_025\_003:

|  |  |
| --- | --- |
| **Variable Name** | ContinueIfFailed |
| **Type** | bool |
| **Range** | None |
| **Description** | Flag to validate the process of Gentool, continue with any failure. |

#### Order

GENERIC\_GDD\_CLS\_025\_004:

|  |  |
| --- | --- |
| **Variable Name** | Order |
| **Type** | float |
| **Range** | None |
| **Description** | None |

#### EnableMethod

GENERIC\_GDD\_CLS\_025\_005:

|  |  |
| --- | --- |
| **Variable Name** | EnableMethod |
| **Type** | bool |
| **Range** | None |
| **Description** | Flag to enable debugger method |

#### preIntermediationValidation

GENERIC\_GDD\_CLS\_025\_006:

|  |  |
| --- | --- |
| **Variable Name** | preIntermediationValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | Flag to validate the process of Gentool is pre-intermetiate. |

#### postIntermediationValidation

GENERIC\_GDD\_CLS\_025\_007:

|  |  |
| --- | --- |
| **Variable Name** | postIntermediationValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | Flag to validate the process of Gentool is post-intermetiate. |

#### continueIfFailed

GENERIC\_GDD\_CLS\_025\_008:

|  |  |
| --- | --- |
| **Variable Name** | continueIfFailed |
| **Type** | bool |
| **Range** | None |
| **Description** | Flag to validate the process of Gentool, continue with any failure. |

### Methods

#### ValidationRuleAttribute

GENERIC\_GDD\_CLS\_025\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidationRuleAttribute | | |
| **Arguments** | order | Provide order that is in method order | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: N/A}

#### ValidationRuleAttribute

GENERIC\_GDD\_CLS\_025\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidationRuleAttribute | | |
| **Arguments** | preIntermediationValidation | Pre intermediation validation | |
| Range: | None |
| postIntermediationValidation | Post intermediation validation | |
| Range: | None |
| continueIfFailed | Continue if failed | |
| Range: | None |
| enableMethod | Flag to enable debugger method | |
| Range: | None |
| order | Provide order that is in method order | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::ConsoleWrapper

|  |  |
| --- | --- |
| **Location File Name** | ConsoleWrapper.cs |
| **Description** | This class is used to wrap .NET methods of System.Console classes. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | ui | Private | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | WriteLine | Public | Call .NET API to write to console, all exceptions .NET API to console are raise at here. |

### Fields

#### ui

GENERIC\_GDD\_CLS\_026\_001:

|  |  |
| --- | --- |
| **Variable Name** | ui |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | None |

### Methods

#### WriteLine

GENERIC\_GDD\_CLS\_026\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | WriteLine | | |
| **Arguments** | value | string value | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Call .NET API to write to console, all exceptions .NET API to console are raise at here.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Console.WriteLine to write to console. | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::EnvironmentWrapper

|  |  |
| --- | --- |
| **Location File Name** | EnvironmentWrapper.cs |
| **Description** | This class is used to wrap .NET methods of System.Environment classes. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Exit | Public | Call .NET API to exit application, all exceptions method Environment.Exit() are raise at here. |

### Fields

### Methods

#### Exit

GENERIC\_GDD\_CLS\_027\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Exit | | |
| **Arguments** | exitCode | int value | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Call .NET API to exit application, all exceptions method Environment.Exit() are raise at here.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Environment.Exit to exit application. | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::IOWrapper

|  |  |
| --- | --- |
| **Location File Name** | IOWrapper.cs |
| **Description** | This class is used to wrap .NET methods of System.IO classes. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | ui | Private | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | CreateDirectory | Public | Handle kind of .NET exceptions incase create directory |
| (2) | GetFiles | Public | Handle kind of .NET exceptions incase get files |
| (3) | DirectoryExists | Public | This method call .NET API to check if a directory exists |
| (4) | GetCurrentDirectory | Public | This method call .NET API to get current directory of executing application |
| (5) | FileExists | Public | Call .NET API to check if a file exists. |
| (6) | ReadAllLines | Public | Handle kind of .NET exceptions incase read content of a file |
| (7) | GetFullPath | Public | Handle kind of .NET exceptions incase get the absolute path |
| (8) | StreamWrite | Public | Handle kind of .NET exceptions incase write content to streamWriter |
| (9) | StreamWriteLine | Public | Handle kind of .NET exceptions incase write content to streamWriter |
| (10) | LoadXML | Public | Handle kind of .NET exceptions incase load XML file. |
| (11) | GetFileName | Public | Handle kind of .NET exceptions incase get file name and extension path string |
| (12) | exit | Private | Call to exit program |

### Fields

#### ui

GENERIC\_GDD\_CLS\_028\_001:

|  |  |
| --- | --- |
| **Variable Name** | ui |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | None |

### Methods

#### CreateDirectory

GENERIC\_GDD\_CLS\_028\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CreateDirectory | | |
| **Arguments** | path | string path | |
| Range: | None |
| exitCode | int? value | |
| Range: | Null or not |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase create directory  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Directory.CreateDirectory(path) to create a new directory.  IF exception(s) is(are) thrown  LET record the error and exit application  Handle .NET exceptions:  UnauthorizedAccessException: record error ERR000050  PathTooLongException: record error ERR000051  IOException: record error ERR000018  DirectoryNotFoundException: record error ERR000018  ArgumentException: record error ERR000018 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_018 GENERIC\_TUM\_U2X\_GT\_ERR\_050 GENERIC\_TUM\_U2X\_GT\_ERR\_051>}

{Ref: [3] <CMN\_GDD\_ERR\_050 CMN\_GDD\_ERR\_051 CMN\_GDD\_ERR\_018>}

#### DirectoryExists

GENERIC\_GDD\_CLS\_028\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | DirectoryExists | | |
| **Arguments** | directory | string path | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | bool | The status of directory exists or not | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | This method call .NET API to check if a directory exists  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Directory.Exists(directory) to check if a directory exists  IF exception is thrown THEN  LET record ERR000050 and exit application | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_050>}

#### FileExists

GENERIC\_GDD\_CLS\_028\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | FileExists | | |
| **Arguments** | file | string file | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | bool | Status of file exists or not | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Call .NET API to check if a file exists.  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API File.Exists(file) to get current directory of executing application.  IF exception is thrown THEN\  LET record ERR000050 and exit application | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_050>}

#### GetCurrentDirectory

GENERIC\_GDD\_CLS\_028\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetCurrentDirectory | | |
| **Arguments** | exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | string | The path of current directory | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | This method call .NET API to get current directory of executing application  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Directory.GetCurrentDirectory() to get current directory of executing application.  IF exception is thrown THEN  LET record ERR000050 and exit application | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_050>}

#### GetFileName

GENERIC\_GDD\_CLS\_028\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetFileName | | |
| **Arguments** | fileName | string file | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | string | The name of file | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase get file name and extension path string  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Path.GetFileName(fileName) to get the file name  and extension of the specified path string.  IF exception(s) is(are) thrown THEN  LET record the error and exit application  Handle .NET exceptions:  ArgumentException: record error ERR000053 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_053>}

{Ref: [3] < CMN\_GDD\_ERR\_053>}

#### GetFiles

GENERIC\_GDD\_CLS\_028\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetFiles | | |
| **Arguments** | path | string path | |
| Range: | Null or not |
| searchPattern | string path | |
| Range: | TopDirectoryOnly AllDirectories |
| searchOption | int? exitCode | |
| Range: | Null or not |
| exitCode | None | |
| Range: | None |
| **Return** | string[] | The list of file at current path | |
| Range: | - |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase get files  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Directory.GetFiles(path, searchPattern, searchOption) to search file names by pattern.  IF exception(s) is(are) thrown THEN  LET record the error and exit application  Handle .NET exceptions:  UnauthorizedAccessException: record error ERR000050  PathTooLongException: record error ERR000051  IOException: record error ERR000018 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_050 GENERIC\_TUM\_U2X\_GT\_ERR\_051 GENERIC\_TUM\_U2X\_GT\_ERR\_053>}

#### GetFullPath

GENERIC\_GDD\_CLS\_028\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetFullPath | | |
| **Arguments** | path | string path | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | string | The absolute path for the specified path string | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase get the absolute path  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Path.GetFullPath(path) to returns the absolute path for the specified path string  IF exception(s) is(are) thrown THEN  LETrecord the error and exit application  Handle .NET exceptions:  SecurityException: record error ERR000050  PathTooLongException: record error ERR000051 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_050 GENERIC\_TUM\_U2X\_GT\_ERR\_051>}

#### LoadXML

GENERIC\_GDD\_CLS\_028\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | LoadXML | | |
| **Arguments** | file | string file | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | XDocument | The content of XML file | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase load XML file.  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API XDocument.Load(file) to load a XML file.  IF exception(s) is(are) thrown THEN  LET record the error and exit application  Handle .NET exceptions:  FormatException: record error ERR000012  XmlException: record error ERR000012  UnauthorizedAccessException: record error ERR000050 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_012 GENERIC\_TUM\_U2X\_GT\_ERR\_050>}

#### ReadAllLines

GENERIC\_GDD\_CLS\_028\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ReadAllLines | | |
| **Arguments** | file | string path | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | string[] | Content of file | |
| Range: | - |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase read content of a file  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API File.ReadAllLines(file) to read content of a file and return  an array of string in which a string is a line.  IF exception(s) is(are) thrown THEN  LET record the error and exit application  Handle .NET exceptions:  UnauthorizedAccessException: record error ERR000050  SecurityException: record error ERR000050  IOException: record error ERR000053 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_050 GENERIC\_TUM\_U2X\_GT\_ERR\_053>}

{Ref: [3] <CMN\_GDD\_ERR\_053>}

#### StreamWrite

GENERIC\_GDD\_CLS\_028\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StreamWrite | | |
| **Arguments** | streamWriter | StreamWriter streamWriter | |
| Range: | Null or not |
| content | string content | |
| Range: | Null or not |
| path | string path | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase write content to streamWriter  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API streamWriter.Write(content) to write content to streamWriter  WHICH in turn will write to a file.  Content is NOT appended a new line character at the end.  IF exception(s) is(are) thrown THEN  LET record the error and exit application  Handle .NET exceptions:  IOException: record error ERR000018 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_053>}

#### StreamWriteLine

GENERIC\_GDD\_CLS\_028\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StreamWriteLine | | |
| **Arguments** | streamWriter | StreamWriter streamWriter | |
| Range: | Null or not |
| content | string content | |
| Range: | Null or not |
| path | string path | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase write content to streamWriter  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API streamWriter.Write(content) to write content to streamWriter  which in turn will write to a file  Content is appended a new line character at the end.  IF exception(s) is(are) thrown THEN  LET record the error and exit application  Handle .NET exceptions:  IOException: record error ERR000018 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_053>}

#### exit

GENERIC\_GDD\_CLS\_028\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | exit | | |
| **Arguments** | exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Call to exit program  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   This method receive parameter as exit code value from calling method.  IF parameter does not provide THEN  LET exit program without status code. | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::ReflectionWrapper

|  |  |
| --- | --- |
| **Location File Name** | ReflectionWrapper.cs |
| **Description** | This class is used to wrap .NET methods of System.Reflection classes |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | ui | Private | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | LoadFile | Public | Handle kind of .NET exceptions incase load the contents of an assembly file. |
| (2) | LoadFrom | Public | Handle kind of .NET exceptions incase load an assembly given its file name or path. |
| (3) | exit | Private | Call to exit program |

### Fields

#### ui

GENERIC\_GDD\_CLS\_029\_001:

|  |  |
| --- | --- |
| **Variable Name** | ui |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | None |

### Methods

#### LoadFile

GENERIC\_GDD\_CLS\_029\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | LoadFile | | |
| **Arguments** | path | string file | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | Assembly | Assembly of file | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase load the contents of an assembly file.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Assembly.LoadFile(path) to load the contents of an assembly file on the specified path.  IF exception(s) is(are) thrown THEN  LET record the error and exit application  Handle .NET exceptions:  FileLoadException: record error ERR000055  BadImageFormatException: record error ERR000056 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_055 GENERIC\_TUM\_U2X\_GT\_ERR\_056>}

{Ref: [3] < CMN\_GDD\_ERR\_055 CMN\_GDD\_ERR\_056>}

#### LoadFrom

GENERIC\_GDD\_CLS\_029\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | LoadFrom | | |
| **Arguments** | path | string file | |
| Range: | Null or not |
| exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | Assembly | Assembly of path | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase load an assembly given its file name or path.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL .NET API Assembly.LoadFrom(path) to load an assembly  LET given its file name or path.  IF exception(s) is(are) thrown THEN  LET record the error and exit application  Handle .NET exceptions:  - FileLoadException: record error ERR000055  - BadImageFormatException: record error ERR000056  - SecurityException: record error ERR000050  - PathTooLongException: record error ERR000051 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_050 GENERIC\_TUM\_U2X\_GT\_ERR\_051 GENERIC\_TUM\_U2X\_GT\_ERR\_055 GENERIC\_TUM\_U2X\_GT\_ERR\_056>}

{Ref: [3] <CMN\_GDD\_ERR\_051 CMN\_GDD\_ERR\_050 CMN\_GDD\_ERR\_055

CMN\_GDD\_ERR\_056>}

#### exit

GENERIC\_GDD\_CLS\_029\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | exit | | |
| **Arguments** | exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Call to exit program  Algorithm: GLOBAL VARIABLE IN:  ui   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   This method receive parameter as exit code value from calling method.  IF parameter does not provide THEN  LET exit program without status code. | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::Logger::DebugLogger

|  |  |
| --- | --- |
| **Location File Name** | DebugLogger.cs |
| **Description** | This class to record log information for debugging. Now it's a placeholder and will be supported in the future. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Debug | Public | Print debug information constructed by format and parameters |
| (2) | Info | Public | Print information constructed by format and parameters |
| (3) | DebugLogger | Private | Debug logger constructor |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_030\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly DebugLogger |
| **Range** | None |
| **Description** | None |

### Methods

#### Debug

GENERIC\_GDD\_CLS\_030\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Debug | | |
| **Arguments** | format | Format | |
| Range: | None |
| parameters | Parameters | |
| Range: | None |
| **Return** | return a void | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Print debug information constructed by format and parameters  Algorithm: Print debug information constructed by format and parameters | | |

{Ref: N/A}

#### Info

GENERIC\_GDD\_CLS\_030\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Info | | |
| **Arguments** | format | Format | |
| Range: | None |
| parameters | Parameters | |
| Range: | None |
| **Return** | return a void | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Print information constructed by format and parameters  Algorithm: Print information constructed by format and parameters | | |

{Ref: N/A}

#### DebugLogger

GENERIC\_GDD\_CLS\_030\_004:

|  |  |
| --- | --- |
| **Method Name** | DebugLogger |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Debug logger constructor |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactory

|  |  |
| --- | --- |
| **Location File Name** | ObjectFactory.cs |
| **Description** | This class which is used to bind an interface to concrete classes. Binding to which concrete classes depend on |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | ModuleName | Public | A string to store module name |
| (2) | SupportedDevices | Public | None |
| (3) | gentoolExecutableTypes | Private | Gentool executable type |
| (4) | registeredTypes | Private | Types that are loaded from DLLs |
| (5) | cache | Private | A cache of initialized instances for interfaces |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ResgisterAssemblyTypes | Public | Register all types from files specified by assemblyFilePaths |
| (2) | GetInstance< T > | Public | Get an instance object with interface is T |
| (3) | GetInstances< T > | Public | Get all instance objects with interface is T |
| (4) | ClearCache | Public | Remove all instance in cache because it need to clear static data each completed instance in case multi-instance |
| (5) | ObjectFactory | Private | Object factory constructor |
| (6) | createInstance | Private | Create object instance from a concrete type |
| (7) | checkConditionType | Private | Check type to meet Autosar and Device conditions |
| (8) | getCacheObjectsOfInterface | Private | Get cache object instance of specified interface |
| (9) | parseOtherConditions | Private | Parse other conditions of |

### Fields

#### ModuleName

GENERIC\_GDD\_CLS\_031\_001:

|  |  |
| --- | --- |
| **Variable Name** | ModuleName |
| **Type** | string |
| **Range** | None |
| **Description** | A string to store module name |

#### SupportedDevices

GENERIC\_GDD\_CLS\_031\_002:

|  |  |
| --- | --- |
| **Variable Name** | SupportedDevices |
| **Type** | List< string > |
| **Range** | None |
| **Description** | None |

#### gentoolExecutableTypes

GENERIC\_GDD\_CLS\_031\_003:

|  |  |
| --- | --- |
| **Variable Name** | gentoolExecutableTypes |
| **Type** | HashSet< Type > |
| **Range** | None |
| **Description** | Gentool executable type |

#### registeredTypes

GENERIC\_GDD\_CLS\_031\_004:

|  |  |
| --- | --- |
| **Variable Name** | registeredTypes |
| **Type** | Dictionary< Type, List< Type > > |
| **Range** | None |
| **Description** | Types that are loaded from DLLs |

#### cache

GENERIC\_GDD\_CLS\_031\_005:

|  |  |
| --- | --- |
| **Variable Name** | cache |
| **Type** | Dictionary< Type, List< CacheObject > > |
| **Range** | None |
| **Description** | A cache of initialized instances for interfaces |

### Methods

#### ClearCache

GENERIC\_GDD\_CLS\_031\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ClearCache | | |
| **Arguments** | type | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Remove all instance in cache because it need to clear static data each completed instance in case multi-instance  Algorithm: GLOBAL VARIABLE IN:  cache   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF cache is contain key type  LET remove type | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ECU\_003 GENERIC\_TUM\_U2x\_GT\_ECU\_004>}

#### GetInstance< T >

GENERIC\_GDD\_CLS\_031\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetInstance< T > | | |
| **Arguments** | None | | |
| **Return** | T | Instance object with interface T | |
| Range: | - |
| **Generated Value** | None | | |
| **Description** | Get an instance object with interface is T  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL getCacheObjectsOfInterface to get all instance objects with interface is T  IF found THEN  RETURN first object  ELSE  Exit Gentool | | |

{Ref: N/A}

#### GetInstances< T >

GENERIC\_GDD\_CLS\_031\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetInstances< T > | | |
| **Arguments** | None | | |
| **Return** | T[] | ALL Instance object with interface T | |
| Range: | - |
| **Generated Value** | None | | |
| **Description** | Get all instance objects with interface is T  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET CALL getCacheObjectsOfInterface to get all instance objects with interface is T  IF found THEN  RETURN first object  ELSE  Exit Gentool | | |

{Ref: N/A}

#### ResgisterAssemblyTypes

GENERIC\_GDD\_CLS\_031\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ResgisterAssemblyTypes | | |
| **Arguments** | assemblyFilePaths | Assembly file paths | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Register all types from files specified by assemblyFilePaths  Algorithm: GLOBAL VARIABLE IN:  registeredTypes   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   FOREACH filePath in assemblyFilePaths  LET specificDLL = CALL LoadFile WITH filePath and exit code = 2  IF specificDLL is not null  LET types = list of the types defined in this assembly which has defined ObjectFactoryAttribute  FOREACH type in types  LET interfaceTypes = load all interface  FOREACH interfaceType in interfaceTypes  IF registeredTypes is not contain key interfaceType  LET Store all interface  FOREACH type in list key of registeredTypes  FOREACH type in types  LET attribute = type.GetCustomAttribute<ObjectFactoryAttribute>  IF attribute is not null and att.Device is not null and att.Device is not Equals '\*'  LET Store all devices on class that is used to compare with device in CDFs. | | |

{Ref: N/A}

#### ObjectFactory

GENERIC\_GDD\_CLS\_031\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ObjectFactory | | |
| **Arguments** | None | | |
| **Return** | Type | The type | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Object factory constructor  Algorithm: GLOBAL VARIABLE IN:  registeredTypes, gentoolExecutableTypes   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A  LET registeredTypes = initialize new Dictionary WITH key is Type and value are list of Type // Implementation: GENERIC\_GDD\_CLS\_031\_004  LET gentoolExecutableTypes = initialize new HashSet of Type // Implementation: GENERIC\_GDD\_CLS\_031\_003  FOREACH assembly in all assembly MCALConfGen  LET types = list of the types defined in this assembly which has defined ObjectFactoryAttribute  FOREACH type in types  LET interfaceTypes = load all interface  FOREACH interfaceType in interfaceTypes  IF registeredTypes is not contain interfaceType  LET value of interfaceType in dictionary registeredTypes = new list of type // Implementation: GENERIC\_GDD\_CLS\_031\_004  LET registeredTypes[interfaceType].Add(type)  LET gentoolExecutableTypes.Add(type)   End | | |

{Ref: N/A}

#### checkConditionType

GENERIC\_GDD\_CLS\_031\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkConditionType | | |
| **Arguments** | concreteType | Concrete type | |
| Range: | None |
| allRelatedTypes | All related types | |
| Range: | None |
| interfaceType | Interface | |
| Range: | None |
| **Return** | bool | The status when check Autosar and Device conditions | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check type to meet Autosar and Device conditions  Algorithm: GLOBAL VARIABLE IN:  gentoolExecutableTypes   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF (AutoSarVersion == "\*" or AutoSarVersion == basicConfig.AutoSARVersion) and  (Device == "\*" or basicConfig.DeviceNames contain Device) THEN  RETURN true  ELSE  RETURN false | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_057>}

{Ref: [3] < CMN\_GDD\_ERR\_057>}

#### createInstance

GENERIC\_GDD\_CLS\_031\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | createInstance | | |
| **Arguments** | concreteType | Concrete type | |
| Range: | None |
| **Return** | Object | The object from a concrete type | |
| Range: | - |
| **Generated Value** | None | | |
| **Description** | Create object instance from a concrete type  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET ctor = CALL GetConstructor WITH Type.EmptyTy"Instance", BindingFlags.Public | BindingFlags.Static)pes  LET getInstance = CALL GetField WITH "Instance", BindingFlags.Public | BindingFlags.Static  IF ctor is not null  LET result = ctor.Invoke(null)  ELSE IF getInstance is not null  result = getInstance.GetValue(null)  RETURN result | | |

{Ref: N/A}

#### getCacheObjectsOfInterface

GENERIC\_GDD\_CLS\_031\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getCacheObjectsOfInterface | | |
| **Arguments** | interfaceType | Interface | |
| Range: | None |
| **Return** | List<CacheObject> | Cache object instance of specified interface | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Get cache object instance of specified interface  Algorithm: GLOBAL VARIABLE IN:  cache, registeredTypes   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET cacheObjects = NULL  IF key of cache contain interfaceType and cache[interfaceType].Count > 0  LET cacheObjects = cache[interfaceType]  ELSE IF key of cache not contain interfaceType and key of registeredTypes contain interfaceType  LET concreteTypes = registeredTypes[interfaceType]  LET cacheObjects = initialize new List<CacheObject>  FOREACH type in concreteTypes (which has the interface in registeredTypes)  LET create a new instance and add to cache  RETURN object instances from cache | | |

{Ref: N/A}

#### parseOtherConditions

GENERIC\_GDD\_CLS\_031\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseOtherConditions | | |
| **Arguments** | otherConditions | Other conditions | |
| Range: | None |
| **Return** | return Dictionary of conditions, key is parameter/reference name and value is expected value | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Parse other conditions of  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF otherConditions is empty string THEN  RETURN empty dictionary  ELSE  LET Split otherConditions into multiple conditions (separator is ;)  FOREACH condition  LET Split condition by separator '=' and add to result dictionary  END  RETURN result dictionary | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_057>}

## Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactory::CacheObject

|  |  |
| --- | --- |
| **Location File Name** | ObjectFactory.cs |
| **Description** | Save instance of type to re-used (lazy loading) |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | Intance |
| (2) | TypeOf | Public | Type of |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | CacheObject | Public | Cache object constructor |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_032\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | Object |
| **Range** | None |
| **Description** | Intance |

#### TypeOf

GENERIC\_GDD\_CLS\_032\_002:

|  |  |
| --- | --- |
| **Variable Name** | TypeOf |
| **Type** | Type |
| **Range** | None |
| **Description** | Type of |

### Methods

#### CacheObject

GENERIC\_GDD\_CLS\_032\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CacheObject | | |
| **Arguments** | typeOf | Type | |
| Range: | None |
| instance | Object instance | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Cache object constructor  Algorithm: This is constructor | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactoryAttribute

|  |  |
| --- | --- |
| **Location File Name** | ObjectFactoryAttribute.cs |
| **Description** | This class contains attribute to be resolved by |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | AutoSarVersion | Public | Auto sar version |
| (2) | Device | Public | Device |
| (3) | Version | Public | Version |
| (4) | ModuleNames | Public | Module Name |
| (5) | OtherConditions | Public | Other conditions Format: |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ObjectFactoryAttribute | Public | Object factory attribute constructor |

### Fields

#### AutoSarVersion

GENERIC\_GDD\_CLS\_033\_001:

|  |  |
| --- | --- |
| **Variable Name** | AutoSarVersion |
| **Type** | string |
| **Range** | None |
| **Description** | Auto sar version |

#### Device

GENERIC\_GDD\_CLS\_033\_002:

|  |  |
| --- | --- |
| **Variable Name** | Device |
| **Type** | string |
| **Range** | None |
| **Description** | Device |

#### Version

GENERIC\_GDD\_CLS\_033\_003:

|  |  |
| --- | --- |
| **Variable Name** | Version |
| **Type** | string |
| **Range** | None |
| **Description** | Version |

#### ModuleNames

GENERIC\_GDD\_CLS\_033\_004:

|  |  |
| --- | --- |
| **Variable Name** | ModuleNames |
| **Type** | string |
| **Range** | None |
| **Description** | Module Name |

#### OtherConditions

GENERIC\_GDD\_CLS\_033\_005:

|  |  |
| --- | --- |
| **Variable Name** | OtherConditions |
| **Type** | string |
| **Range** | None |
| **Description** | Other conditions Format: |

### Methods

#### ObjectFactoryAttribute

GENERIC\_GDD\_CLS\_033\_006:

|  |  |
| --- | --- |
| **Method Name** | ObjectFactoryAttribute |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Object factory attribute constructor |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::UserInterface::UserInterface

|  |  |
| --- | --- |
| **Location File Name** | UserInterface.cs |
| **Description** | This class that implements |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | ErrorCount | Public | Error count |
| (3) | WarningCount | Public | Warning count |
| (4) | runtimeConfig | Private | Runtime config |
| (5) | basicConfig | Private | Basic config |
| (6) | swLog | Private | Stream writer log |
| (7) | logContent | Private | Log content |
| (8) | errorCount | Private | Error count |
| (9) | warningCount | Private | Warning count |
| (10) | inExitMode | Private | in Exit Mode |
| (11) | exitCodeDic | Private | exit Code Dictionary |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Error | Public | Record error |
| (2) | Warn | Public | Report warning |
| (3) | Info | Public | Report info |
| (4) | OpenFile | Public | Open file from path |
| (5) | Exit | Public | Exit Gentool without parameter |
| (6) | Exit | Public | Exit Gentool with parameter exit code |
| (7) | ErrorFileNotFound | Public | Handle file not found error |
| (8) | ErrorModuleNotFound | Public | Handle module not found error |
| (9) | InfoErrorCode | Public | Record error based on error code |
| (10) | getDateTime | Private | Get current date time with format "dd MMM yyyy - hh:mm:ss" |
| (11) | getCallerClass | Private | Get caller class |
| (12) | report | Private | Report error, warning and info |
| (13) | textWrap | Private | Wrap text |
| (14) | writeLog | Private | Write content to log file |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_034\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly UserInterface |
| **Range** | None |
| **Description** | None |

#### ErrorCount

GENERIC\_GDD\_CLS\_034\_002:

|  |  |
| --- | --- |
| **Variable Name** | ErrorCount |
| **Type** | int |
| **Range** | None |
| **Description** | Error count |

#### WarningCount

GENERIC\_GDD\_CLS\_034\_003:

|  |  |
| --- | --- |
| **Variable Name** | WarningCount |
| **Type** | int |
| **Range** | None |
| **Description** | Warning count |

#### runtimeConfig

GENERIC\_GDD\_CLS\_034\_004:

|  |  |
| --- | --- |
| **Variable Name** | runtimeConfig |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | Runtime config |

#### basicConfig

GENERIC\_GDD\_CLS\_034\_005:

|  |  |
| --- | --- |
| **Variable Name** | basicConfig |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | Basic config |

#### swLog

GENERIC\_GDD\_CLS\_034\_006:

|  |  |
| --- | --- |
| **Variable Name** | swLog |
| **Type** | StreamWriter |
| **Range** | None |
| **Description** | Stream writer log |

#### logContent

GENERIC\_GDD\_CLS\_034\_007:

|  |  |
| --- | --- |
| **Variable Name** | logContent |
| **Type** | StringBuilder |
| **Range** | None |
| **Description** | Log content |

#### errorCount

GENERIC\_GDD\_CLS\_034\_008:

|  |  |
| --- | --- |
| **Variable Name** | errorCount |
| **Type** | int |
| **Range** | None |
| **Description** | Error count |

#### warningCount

GENERIC\_GDD\_CLS\_034\_009:

|  |  |
| --- | --- |
| **Variable Name** | warningCount |
| **Type** | int |
| **Range** | None |
| **Description** | Warning count |

#### inExitMode

GENERIC\_GDD\_CLS\_034\_010:

|  |  |
| --- | --- |
| **Variable Name** | inExitMode |
| **Type** | bool |
| **Range** | None |
| **Description** | in Exit Mode |

#### exitCodeDic

GENERIC\_GDD\_CLS\_034\_011:

|  |  |
| --- | --- |
| **Variable Name** | exitCodeDic |
| **Type** | Dictionary< Type, int > |
| **Range** | None |
| **Description** | exit Code Dictionary |

### Methods

#### Error

GENERIC\_GDD\_CLS\_034\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Error | | |
| **Arguments** | moduleId | Module id | |
| Range: | None |
| errorCode | Error code | |
| Range: | None |
| message | Message | |
| Range: | None |
| parameters | Parameters | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Record error  Algorithm: GLOBAL VARIABLE IN:  errorCount   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Increase number of errors and call report method  CALL report method WITH moduleId, errorCode, "ERR", message, parameters | | |

{Ref: N/A}

#### ErrorFileNotFound

GENERIC\_GDD\_CLS\_034\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ErrorFileNotFound | | |
| **Arguments** | fileName | File name | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Handle file not found error  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Record error ERR000001  LET CALL Exit method with exitCode EXIT\_DUE\_CMD\_ERROR | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_001>}

{Ref: [3] < CMN\_GDD\_ERR\_001>}

#### ErrorModuleNotFound

GENERIC\_GDD\_CLS\_034\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ErrorModuleNotFound | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Handle module not found error  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Record error ERR000031  LET Record info INF000005  LET CALL infoErrorCode method to record info based on error code EXIT\_DUE\_CMD\_ERROR  LET CALL Environment.Exit with 1 | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_ERR\_031>}

{Ref: [3] < CMN\_GDD\_ERR\_031>}

#### Exit

GENERIC\_GDD\_CLS\_034\_015:

|  |  |
| --- | --- |
| **Method Name** | Exit |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Exit Gentool without parameter  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Identify exitcode from caller class  IF CALL class NOT found in exitCodeDic THEN  LET exitcode = EXIT\_SUCCESSFUL  LET CALL Exit method with parameter exitcode |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_004 GENERIC\_TUM\_U2X\_GT\_INF\_005>}

#### Exit

GENERIC\_GDD\_CLS\_034\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Exit | | |
| **Arguments** | exitCode | Exit code | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Exit Gentool with parameter exit code  Algorithm: GLOBAL VARIABLE IN:  runtimeConfig, swLog, basicConfig   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Use basicConfig.OutputFolder if runtimeConfig.FolderOutput NOT existed  IF runtimeConfig.DryRun is false THEN  LET Write to log file  LET Handle error ERR000004 when can NOT write log file  End   LET Record info INF000005  LET CALL infoErrorCode method to record info based on exitCode  IF exitCode = EXIT\_SUCCESSFUL THEN  LET Environment.Exit with 0  ELSE  LET Environment.Exit with 1 | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_004 GENERIC\_TUM\_U2X\_GT\_INF\_005>}

{Ref: [3] < CMN\_GDD\_ERR\_004 CMN\_GDD\_ERR\_005>}

#### Info

GENERIC\_GDD\_CLS\_034\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Info | | |
| **Arguments** | moduleId | Module id | |
| Range: | None |
| errorCode | Error code | |
| Range: | None |
| message | Message | |
| Range: | None |
| parameters | Parameters | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Report info  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   CALL report method WITH moduleId, errorCode, "INF", message, parameters | | |

{Ref: N/A}

#### InfoErrorCode

GENERIC\_GDD\_CLS\_034\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | InfoErrorCode | | |
| **Arguments** | errorcode | Errorcode | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Record error based on error code  Algorithm: GLOBAL VARIABLE IN:  WarningCount   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF errorcode iS:  - EXIT\_DUE\_CMD\_ERROR THEN LET record info INF000008  - EXIT\_SUCCESSFUL and EXIT\_SUCCESSFUL\_WITH\_WARNINGS  + IF WarningCount > 0 THEN LET record info INF000008  + ELSE LET record info INF000006  - EXIT\_DUE\_INPUT\_FILES\_ERROR THEN LET record info INF000009  - EXIT\_DUE\_STRUCTURE\_GENERATION\_ERROR THEN LET record info INF000010  - EXIT\_DUE\_OVERWRITTEN\_FILE\_ERROR THEN LET record info INF000011 | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_INF\_006 GENERIC\_TUM\_U2x\_GT\_INF\_007 GENERIC\_TUM\_U2x\_GT\_INF\_008 GENERIC\_TUM\_U2x\_GT\_INF\_009, GENERIC\_TUM\_U2x\_GT\_INF\_010 GENERIC\_TUM\_U2x\_GT\_INF\_011 GENERIC\_TUM\_U2x\_GT\_INF\_012 GENERIC\_TUM\_U2x\_GT\_INF\_014>}

#### OpenFile

GENERIC\_GDD\_CLS\_034\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | OpenFile | | |
| **Arguments** | path | Path | |
| Range: | None |
| **Return** | StreamWriter | The string wrapped | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Open file from path  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Open file from path  LET CALL Info WITH (0, 4, String.Format(Resources.INF000004, StringUtils.GetRelativePath(path),  getDateTime())) TO Record info INF000004 | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_INF\_004>}

#### Warn

GENERIC\_GDD\_CLS\_034\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Warn | | |
| **Arguments** | moduleId | Module id | |
| Range: | None |
| errorCode | Error code | |
| Range: | None |
| message | Message | |
| Range: | None |
| parameters | Parameters | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Report warning  Algorithm: GLOBAL VARIABLE IN:  warningCount   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Increase number of warnings and call report method  CALL report method WITH moduleId, errorCode, "WRN", message, parameters | | |

{Ref: N/A}

#### getCallerClass

GENERIC\_GDD\_CLS\_034\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getCallerClass | | |
| **Arguments** | None | | |
| **Return** | Type | The type | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Get caller class  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET ret WITH type is Type and initialize value is null  LET stack = initialize new StackTrace  FOR index from 0 to length of stack.GetFrames  LET type = stack.GetFrame(index).GetMethod  IF type.DeclaringType.GetInterfaces().FirstOrDefault is not null THEN  LET ret = type.DeclaringType.GetInterfaces().FirstOrDefault  RETURN ret | | |

{Ref: N/A}

#### getDateTime

GENERIC\_GDD\_CLS\_034\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getDateTime | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get current date time with format "dd MMM yyyy - hh:mm:ss"  Algorithm: Get current date time with format "dd MMM yyyy - hh:mm:ss" | | |

{Ref: N/A}

#### report

GENERIC\_GDD\_CLS\_034\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | report | | |
| **Arguments** | moduleId | Module id | |
| Range: | None |
| errorCode | Error code | |
| Range: | None |
| msgType | Message type: ERR, WRN and INF | |
| Range: | None |
| msg | Message format | |
| Range: | None |
| parameters | Parameters | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Report error, warning and info  Algorithm: GLOBAL VARIABLE IN:  logContent   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   Infomration = msgType + moduleId + errorCode + message which is constructed from format and parameters  IF Infomration length is larger than 80 THEN  LET Split Infomration into multiple lines, each of which has maximum length is 80  End  LET wite information to logContent and Console | | |

{Ref: N/A}

#### textWrap

GENERIC\_GDD\_CLS\_034\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | textWrap | | |
| **Arguments** | prefix | Prefix | |
| Range: | None |
| inputString | Input string | |
| Range: | None |
| width | Width | |
| Range: | None |
| **Return** | string | The string wrapped | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Wrap text  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET outStr = ""  LET newLinePrefix = ""  LET outlines = initialize list of string  IF prefix is not null THEN  prefix += ""  LET inputString.Trim()  LET newLinePrefix = String.Concat(Enumerable.Repeat(" ", prefix.Length))  LET outStr = prefix  LET words = Split Infomration into multiple lines  FOREACH word in words  IF length of word + length of outStr less than outStr THEN  LET outStr += word + " "  ELSE  LET outlines.Add(outStr);  LET outStr = Environment.NewLine + newLinePrefix + word + " "  RETURN String.Join(" ", outlines) + outStr | | |

{Ref: N/A}

#### writeLog

GENERIC\_GDD\_CLS\_034\_025:

|  |  |
| --- | --- |
| **Method Name** | writeLog |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Write content to log file  Algorithm: GLOBAL VARIABLE IN:  basicConfig, swLog, LogFilePath, runtimeConfig   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF 0 less than length of logContent and swLog is not null  LET str = String.Format(logContent.ToString(), basicConfig.ModuleName.ToUpper(), getDateTime(),  String.Join(Environment.NewLine + "#\t", runtimeConfig.CDFFilePaths.ToArray())  LET IOWrapper.StreamWriteLine(swLog, str, runtimeConfig.LogFilePath |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::CrossCutting::Util::Constant

|  |  |
| --- | --- |
| **Location File Name** | Constant.cs |
| **Description** | This class define common constant and is used by other class. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | ON | Public | ON |
| (2) | OFF | Public | OFF |
| (3) | STD\_ON | Public | StdOn |
| (4) | STD\_OFF | Public | StdOff |
| (5) | ECUC\_CONTAINER\_VALUE | Public | Ecuc container value |
| (6) | CONTAINERS | Public | Container |
| (7) | SUB\_CONTAINERS | Public | Subcontainer |
| (8) | ECUC\_MODULE\_CONFIGURATION\_VALUES | Public | Ecuc module configuration value |
| (9) | BSW\_IMPLEMENTATION | Public | Basic software implementation |
| (10) | BSW\_MODULE\_DESCRIPTION | Public | Basic software module description |
| (11) | TRANSLATION\_FILE\_PATH | Public | Translation file path |
| (12) | DEVICE\_FILE\_PATH | Public | Device file path |
| (13) | ONE | Public | ONE |
| (14) | ZERO | Public | ZERO |
| (15) | TRUE | Public | TRUE |
| (16) | FALSE | Public | FALSE |
| (17) | IOREG | Public | IOREG |
| (18) | OPTION | Public | OPTION |
| (19) | INPUT\_FILE | Public | Input file |
| (20) | AR\_RELEASE\_VERSION | Public | Ar release version |
| (21) | SW\_VERSION | Public | Ar software version |
| (22) | VENDOR\_ID | Public | Vendor id |
| (23) | MODULE\_ID | Public | Module id |
| (24) | AR\_PACKAGE | Public | Ar package |
| (25) | AR\_PACKAGES | Public | Ar packages |
| (26) | ELEMENTS | Public | Elements |
| (27) | SHORT\_NAME | Public | Short name |
| (28) | ECUC\_VALUE\_COLLECTION | Public | Ecuc value collection |
| (29) | DEVICE\_NAME\_E2M\_E2xFCC1 | Public | Device name E2M\_E2xFCC1 |
| (30) | DEVICE\_NAME\_E2M | Public | Device name E2M |
| (31) | DEVICE\_NAME\_E2L | Public | Device name E2L |
| (32) | EXIT\_DUE\_CMD\_ERROR | Public | Exit due to command line error |
| (33) | EXIT\_SUCCESSFUL | Public | Exit successful |
| (34) | EXIT\_SUCCESSFUL\_WITH\_WARNINGS | Public | Exit successful with warnings |
| (35) | EXIT\_DUE\_INPUT\_FILES\_ERROR | Public | Exit due to input files error |
| (36) | EXIT\_DUE\_STRUCTURE\_GENERATION\_ERROR | Public | Exit due to structure generation error |
| (37) | EXIT\_DUE\_OVERWRITTEN\_FILE\_ERROR | Public | Exit due to overwritten file error |
| (38) | EXIT\_DUE\_CRASH | Public | Exit due to overwritten file error |
| (39) | EXIT\_DUE\_UNCOMPATIBLE\_DLL | Public | Exit due to overwritten file error |
| (40) | EXIT\_DUE\_DLL\_NOT\_FOUND | Public | Exit due to overwritten file error |
| (41) | DEVICE\_NAME\_E2GH | Public | Device name E2GH |
| (42) | DEVICE\_NAME\_E2GH\_E2xFCC2 | Public | Device name E2GH\_E2xFCC2 |
| (43) | DEVICE\_NAME\_E2GUH | Public | Device name E2GUH |
| (44) | DEVICE\_NAME\_E2GUH\_E2xFCC2 | Public | Device name E2GUH\_E2xFCC2 |
| (45) | DEVICE\_NAME\_E2H | Public | Device name E2H |
| (46) | DEVICE\_NAME\_E2H\_E2xFCC2 | Public | Device name E2H\_E2xFCC2 |
| (47) | DEVICE\_NAME\_E2UH | Public | Device name E2UH |
| (48) | DEVICE\_NAME\_E2UH\_E2xFCC2 | Public | Device name E2UH\_E2xFCC2 |
| (49) | DEVICE\_NAME\_ALL | Public | Device name of all. |
| (50) | AR\_VERSION\_4\_2\_2 | Public | AutoSar Version 4.2.2 |
| (51) | AR\_VERSION\_ALL | Public | AutoSar Version of all support |
| (52) | VERSION\_1\_0\_0 | Public | Version 1.0.0 |
| (53) | VERSION\_ALL | Public | Version of all support |
| (54) | DEVICE\_NAME\_E2H\_E2xFCC2\_468 | Public | Device name E2H\_E2xFCC2 with package 468 pins |
| (55) | DEVICE\_NAME\_E2H\_E2xFCC2\_373 | Public | Device name E2H\_E2xFCC2 with package 373 pins |
| (56) | DEVICE\_NAME\_E2M\_373 | Public | Device name E2M with package 373 pins |
| (57) | DEVICE\_NAME\_E2M\_292 | Public | Device name E2M with package 292 pins |
| (58) | DEVICE\_NAME\_E2M\_E2xFCC1\_373 | Public | Device name E2M\_E2xFCC1 with package 373 pins |
| (59) | DEVICE\_NAME\_E2M\_E2xFCC1\_292 | Public | Device name E2M\_E2xFCC1 with package 292 pins |
| (60) | FILTER | Public | FILTER |
| (61) | DEFINITION\_REF | Public | DEFINITION REF |
| (62) | VALUE\_REF | Public | VALUE REF |
| (63) | DEST | Public | DEST |
| (64) | MODULE\_DESCRIPTION\_REF | Public | Module description ref |
| (65) | UUID | Public | UUID |
| (66) | PARAMETER\_VALUES | Public | Parameter Values |
| (67) | REFERENCE\_VALUES | Public | Reference Values |
| (68) | VENDOR\_API\_INFIX | Public | Vendor API infix |
| (69) | STR\_ON | Public | STR\_ON |
| (70) | STR\_OFF | Public | STR\_OFF |
| (71) | OUTPUT\_PATH | Public | OUTPUT PATH |
| (72) | FILTER\_NAME | Public | FILTER NAME |
| (73) | NAN | Public | NAN |
| (74) | DOUBLE\_MIN\_VALUE | Public | Min value float point |
| (75) | DOUBLE\_MAX\_VALUE | Public | Max value float point |
| (76) | VALUE | Public | Value string |
| (77) | ECUC\_BOOLEAN\_PARAM\_DEF | Public | Mark for ECUC BOOLEAN PARAM DEF attribute CDF |
| (78) | ECUC\_INTEGER\_PARAM\_DEF | Public | Mark for ECUC\_INTEGER\_PARAM\_DEF attribute CDF |
| (79) | ECUC\_FLOAT\_PARAM\_DEF | Public | Mark for ECUC\_FLOAT\_PARAM\_DEF attribute CDF |
| (80) | ECUC\_STRING\_PARAM\_DEF | Public | Mark for ECUC\_STRING\_PARAM\_DEF attribute CDF |
| (81) | ECUC\_LINKER\_SYMBOL\_DEF | Public | Mark for ECUC\_LINKER\_SYMBOL\_DEF attribute CDF |
| (82) | ECUC\_FUNCTION\_NAME\_DEF | Public | Mark for ECUC\_FUNCTION\_NAME\_DEF attribute CDF |
| (83) | ECUC\_ENUMERATION\_PARAM\_DEF | Public | Mark for ECUC\_ENUMERATION\_PARAM\_DEF attribute CDF |
| (84) | ECUC\_ENUMERATION\_LITERAL\_DEF | Public | Mark for ECUC\_ENUMERATION\_LITERAL\_DEF attribute CDF |
| (85) | ECUC\_SYMBOLIC\_NAME\_REFERENCE\_DEF | Public | Mark for ECUC\_SYMBOLIC\_NAME\_REFERENCE\_DEF attribute CDF |
| (86) | ECUC\_URI\_REFERENCE\_DEF | Public | Mark for ECUC\_URI\_REFERENCE\_DEF attribute CDF |
| (87) | ECUC\_ADD\_INFO\_PARAM\_DEF | Public | Mark for ECUC\_ADD\_INFO\_PARAM\_DEF attribute CDF |
| (88) | INVALID\_ATTRIBUTE | Public | Invalid Attribute |
| (89) | DEVICE\_NAME\_E2GH\_468 | Public | Device name E2GH with package 468 pins |
| (90) | DEVICE\_NAME\_E2GH\_373 | Public | Device name E2GH with package 373 pins |
| (91) | DEVICE\_NAME\_E2GH\_E2xFCC2\_468 | Public | Device name E2GH\_E2xFCC2 with package 468 pins |
| (92) | DEVICE\_NAME\_E2GH\_E2xFCC2\_373 | Public | Device name E2GH\_E2xFCC2 with package 373 pins |
| (93) | DEVICE\_NAME\_E2H\_468 | Public | Device name E2H with package 468 pins |
| (94) | DEVICE\_NAME\_E2H\_373 | Public | Device name E2H with package 373 pins |
| (95) | INT\_TWO | Public | None |
| (96) | INT\_THREE | Public | None |
| (97) | INT\_FOUR | Public | FOUR numeric |
| (98) | INSTANCE\_INDEX | Public | Instance index |
| (99) | MAX\_LINE\_LENGTH | Public | Max line length |
| (100) | INDENT\_SECTION | Public | Indent section length |
| (101) | OUTPUT | Public | OUTPUT |
| (102) | IMPLEMENT\_PREFIX | Public | Implement Prefix |
| (103) | DESCRIPTION\_PREFIX | Public | Implement Prefix |
| (104) | DEVICE\_NAME\_U2A\_EVA | Public | Implement Prefix |
| (105) | CDF\_EXTENSION | Public | CDF Extension |
| (106) | TRANSLATION\_FILE\_EXTENSION | Public | Translation Extension |
| (107) | NULL | Public | NULL |
| (108) | NULL\_PTR | Public | NULL\_PTR |
| (109) | ON\_OFF\_PARAMS | Public | ON\_OFF\_PARAMS |
| (110) | DEM\_EVENT\_PARAMS | Public | DEM\_EVENT\_PARAMS |
| (111) | DEM\_CONF\_DEM\_EVENT\_PARAMETER | Public | DEM\_CONF\_DEM\_EVENT\_PARAMETER |
| (112) | MODULE\_NAME | Public | MODULE\_NAME |
| (113) | MICROFAMILY\_NAME | Public | MICROFAMILY\_NAME |
| (114) | MODULE\_REQUIRED | Public | MODULE\_REQUIRED |
| (115) | PROJECT\_TITLE | Public | PROJECT\_TITLE |
| (116) | MODULE\_TOOL\_ID | Public | MODULE\_TOOL\_ID |
| (117) | VERSION\_INFORMATION\_DATA | Public | VERSION INFORMATION DATA |
| (118) | CFG\_AR\_RELEASE\_MAJOR\_VERSION | Public | CFG AR RELEASE MAJOR VERSION |
| (119) | CFG\_AR\_RELEASE\_MINOR\_VERSION | Public | CFG AR RELEASE MINOR VERSION |
| (120) | CFG\_AR\_RELEASE\_REVISION\_VERSION | Public | CFG AR RELEASE REVISION VERSION |
| (121) | CFG\_SW\_MAJOR\_VERSION | Public | CFG SW MAJOR VERSION |
| (122) | CFG\_SW\_MINOR\_VERSION | Public | CFG SW MINOR VERSION |
| (123) | AR\_RELEASE\_MAJOR\_VERSION\_VALUE | Public | AR RELEASE MAJOR VERSION VALUE |
| (124) | AR\_RELEASE\_MINOR\_VERSION\_VALUE | Public | AR RELEASE MINOR VERSION VALUE |
| (125) | AR\_RELEASE\_REVISION\_VERSION\_VALUE | Public | AR RELEASE REVISION VERSION VALUE |
| (126) | SW\_MAJOR\_VERSION\_VALUE | Public | SW MAJOR VERSION VALUE |
| (127) | SW\_MINOR\_VERSION\_VALUE | Public | SW MINOR VERSION VALUE |
| (128) | SW\_PATCH\_VERSION\_VALUE | Public | SW PATCH VERSION VALUE |
| (129) | VENDOR\_ID\_VALUE | Public | VENDOR ID VALUE |
| (130) | MODULE\_ID\_VALUE | Public | MODULE ID VALUE |
| (131) | CBK\_AR\_RELEASE\_MAJOR\_VERSION | Public | CBK AR RELEASE MAJOR VERSION |
| (132) | CBK\_AR\_RELEASE\_MINOR\_VERSION | Public | CBK AR RELEASE MINOR VERSION |
| (133) | CBK\_AR\_RELEASE\_REVISION\_VERSION | Public | CBK AR RELEASE REVISION VERSION |
| (134) | CBK\_SW\_MAJOR\_VERSION | Public | CBK SW MAJOR VERSION |
| (135) | CBK\_SW\_MINOR\_VERSION | Public | CBK SW MINOR VERSION |
| (136) | PBCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION | Public | PBCFG C AR RELEASE MAJOR VERSION |
| (137) | PBCFG\_C\_AR\_RELEASE\_MINOR\_VERSION | Public | PBCFG C AR RELEASE MINOR VERSION |
| (138) | PBCFG\_C\_AR\_RELEASE\_REVISION\_VERSION | Public | PBCFG C AR RELEASE REVISION VERSION |
| (139) | PBCFG\_C\_SW\_MAJOR\_VERSION | Public | PBCFG C SW MAJOR VERSION |
| (140) | PBCFG\_C\_SW\_MINOR\_VERSION | Public | PBCFG C SW MINOR VERSION |
| (141) | VERSION\_INFORMATION\_CBK | Public | VERSION INFORMATION CBK |
| (142) | VERSION\_INFORMATION\_PBCFG | Public | VERSION INFORMATION PBCFG |
| (143) | CALL\_BACK | Public | CALL BACK |
| (144) | POST\_BUILD | Public | POST BUILD |
| (145) | VERSION\_INFORMATION\_CFG | Public | None |
| (146) | LCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION | Public | None |
| (147) | LCFG\_C\_AR\_RELEASE\_MINOR\_VERSION | Public | None |
| (148) | LCFG\_C\_AR\_RELEASE\_REVISION\_VERSION | Public | None |
| (149) | LCFG\_C\_SW\_MAJOR\_VERSION | Public | None |
| (150) | LCFG\_C\_SW\_MINOR\_VERSION | Public | None |
| (151) | LINK\_TIME | Public | None |
| (152) | VERSION\_INFORMATION\_LT | Public | None |
| (153) | BoolDataTypeMarked | Public | None |
| (154) | FloatDataTypeMarked | Public | None |
| (155) | VendorIdPosition | Public | None |
| (156) | ModuleIdPosition | Public | None |
| (157) | SwMajorVersionPosition | Public | None |
| (158) | SwMinorVersionPosition | Public | None |
| (159) | INT\_ZERO | Public | None |
| (160) | INT\_ONE | Public | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| - | - | - | - |

### Fields

#### ON

GENERIC\_GDD\_CLS\_035\_001:

|  |  |
| --- | --- |
| **Variable Name** | ON |
| **Type** | const bool |
| **Range** | None |
| **Description** | ON |

#### OFF

GENERIC\_GDD\_CLS\_035\_002:

|  |  |
| --- | --- |
| **Variable Name** | OFF |
| **Type** | const bool |
| **Range** | None |
| **Description** | OFF |

#### STD\_ON

GENERIC\_GDD\_CLS\_035\_003:

|  |  |
| --- | --- |
| **Variable Name** | STD\_ON |
| **Type** | const string |
| **Range** | None |
| **Description** | StdOn |

#### STD\_OFF

GENERIC\_GDD\_CLS\_035\_004:

|  |  |
| --- | --- |
| **Variable Name** | STD\_OFF |
| **Type** | const string |
| **Range** | None |
| **Description** | StdOff |

#### ECUC\_CONTAINER\_VALUE

GENERIC\_GDD\_CLS\_035\_005:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_CONTAINER\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | Ecuc container value |

#### CONTAINERS

GENERIC\_GDD\_CLS\_035\_006:

|  |  |
| --- | --- |
| **Variable Name** | CONTAINERS |
| **Type** | const string |
| **Range** | None |
| **Description** | Container |

#### SUB\_CONTAINERS

GENERIC\_GDD\_CLS\_035\_007:

|  |  |
| --- | --- |
| **Variable Name** | SUB\_CONTAINERS |
| **Type** | const string |
| **Range** | None |
| **Description** | Subcontainer |

#### ECUC\_MODULE\_CONFIGURATION\_VALUES

GENERIC\_GDD\_CLS\_035\_008:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_MODULE\_CONFIGURATION\_VALUES |
| **Type** | const string |
| **Range** | None |
| **Description** | Ecuc module configuration value |

#### BSW\_IMPLEMENTATION

GENERIC\_GDD\_CLS\_035\_009:

|  |  |
| --- | --- |
| **Variable Name** | BSW\_IMPLEMENTATION |
| **Type** | const string |
| **Range** | None |
| **Description** | Basic software implementation |

#### BSW\_MODULE\_DESCRIPTION

GENERIC\_GDD\_CLS\_035\_010:

|  |  |
| --- | --- |
| **Variable Name** | BSW\_MODULE\_DESCRIPTION |
| **Type** | const string |
| **Range** | None |
| **Description** | Basic software module description |

#### TRANSLATION\_FILE\_PATH

GENERIC\_GDD\_CLS\_035\_011:

|  |  |
| --- | --- |
| **Variable Name** | TRANSLATION\_FILE\_PATH |
| **Type** | const string |
| **Range** | None |
| **Description** | Translation file path |

#### DEVICE\_FILE\_PATH

GENERIC\_GDD\_CLS\_035\_012:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_FILE\_PATH |
| **Type** | const string |
| **Range** | None |
| **Description** | Device file path |

#### ONE

GENERIC\_GDD\_CLS\_035\_013:

|  |  |
| --- | --- |
| **Variable Name** | ONE |
| **Type** | const string |
| **Range** | None |
| **Description** | ONE |

#### ZERO

GENERIC\_GDD\_CLS\_035\_014:

|  |  |
| --- | --- |
| **Variable Name** | ZERO |
| **Type** | const string |
| **Range** | None |
| **Description** | ZERO |

#### TRUE

GENERIC\_GDD\_CLS\_035\_015:

|  |  |
| --- | --- |
| **Variable Name** | TRUE |
| **Type** | const string |
| **Range** | None |
| **Description** | TRUE |

#### FALSE

GENERIC\_GDD\_CLS\_035\_016:

|  |  |
| --- | --- |
| **Variable Name** | FALSE |
| **Type** | const string |
| **Range** | None |
| **Description** | FALSE |

#### IOREG

GENERIC\_GDD\_CLS\_035\_017:

|  |  |
| --- | --- |
| **Variable Name** | IOREG |
| **Type** | const string |
| **Range** | None |
| **Description** | IOREG |

#### OPTION

GENERIC\_GDD\_CLS\_035\_018:

|  |  |
| --- | --- |
| **Variable Name** | OPTION |
| **Type** | const string |
| **Range** | None |
| **Description** | OPTION |

#### INPUT\_FILE

GENERIC\_GDD\_CLS\_035\_019:

|  |  |
| --- | --- |
| **Variable Name** | INPUT\_FILE |
| **Type** | const string |
| **Range** | None |
| **Description** | Input file |

#### AR\_RELEASE\_VERSION

GENERIC\_GDD\_CLS\_035\_020:

|  |  |
| --- | --- |
| **Variable Name** | AR\_RELEASE\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | Ar release version |

#### SW\_VERSION

GENERIC\_GDD\_CLS\_035\_021:

|  |  |
| --- | --- |
| **Variable Name** | SW\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | Ar software version |

#### VENDOR\_ID

GENERIC\_GDD\_CLS\_035\_022:

|  |  |
| --- | --- |
| **Variable Name** | VENDOR\_ID |
| **Type** | const string |
| **Range** | None |
| **Description** | Vendor id |

#### MODULE\_ID

GENERIC\_GDD\_CLS\_035\_023:

|  |  |
| --- | --- |
| **Variable Name** | MODULE\_ID |
| **Type** | const string |
| **Range** | None |
| **Description** | Module id |

#### AR\_PACKAGE

GENERIC\_GDD\_CLS\_035\_024:

|  |  |
| --- | --- |
| **Variable Name** | AR\_PACKAGE |
| **Type** | const string |
| **Range** | None |
| **Description** | Ar package |

#### AR\_PACKAGES

GENERIC\_GDD\_CLS\_035\_025:

|  |  |
| --- | --- |
| **Variable Name** | AR\_PACKAGES |
| **Type** | const string |
| **Range** | None |
| **Description** | Ar packages |

#### ELEMENTS

GENERIC\_GDD\_CLS\_035\_026:

|  |  |
| --- | --- |
| **Variable Name** | ELEMENTS |
| **Type** | const string |
| **Range** | None |
| **Description** | Elements |

#### SHORT\_NAME

GENERIC\_GDD\_CLS\_035\_027:

|  |  |
| --- | --- |
| **Variable Name** | SHORT\_NAME |
| **Type** | const string |
| **Range** | None |
| **Description** | Short name |

#### ECUC\_VALUE\_COLLECTION

GENERIC\_GDD\_CLS\_035\_028:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_VALUE\_COLLECTION |
| **Type** | const string |
| **Range** | None |
| **Description** | Ecuc value collection |

#### DEVICE\_NAME\_E2M\_E2xFCC1

GENERIC\_GDD\_CLS\_035\_029:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2M\_E2xFCC1 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2M\_E2xFCC1 |

#### DEVICE\_NAME\_E2M

GENERIC\_GDD\_CLS\_035\_030:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2M |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2M |

#### DEVICE\_NAME\_E2L

GENERIC\_GDD\_CLS\_035\_031:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2L |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2L |

#### EXIT\_DUE\_CMD\_ERROR

GENERIC\_GDD\_CLS\_035\_032:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_DUE\_CMD\_ERROR |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit due to command line error |

#### EXIT\_SUCCESSFUL

GENERIC\_GDD\_CLS\_035\_033:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_SUCCESSFUL |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit successful |

#### EXIT\_SUCCESSFUL\_WITH\_WARNINGS

GENERIC\_GDD\_CLS\_035\_034:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_SUCCESSFUL\_WITH\_WARNINGS |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit successful with warnings |

#### EXIT\_DUE\_INPUT\_FILES\_ERROR

GENERIC\_GDD\_CLS\_035\_035:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_DUE\_INPUT\_FILES\_ERROR |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit due to input files error |

#### EXIT\_DUE\_STRUCTURE\_GENERATION\_ERROR

GENERIC\_GDD\_CLS\_035\_036:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_DUE\_STRUCTURE\_GENERATION\_ERROR |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit due to structure generation error |

#### EXIT\_DUE\_OVERWRITTEN\_FILE\_ERROR

GENERIC\_GDD\_CLS\_035\_037:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_DUE\_OVERWRITTEN\_FILE\_ERROR |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit due to overwritten file error |

#### EXIT\_DUE\_CRASH

GENERIC\_GDD\_CLS\_035\_038:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_DUE\_CRASH |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit due to overwritten file error |

#### EXIT\_DUE\_UNCOMPATIBLE\_DLL

GENERIC\_GDD\_CLS\_035\_039:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_DUE\_UNCOMPATIBLE\_DLL |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit due to overwritten file error |

#### EXIT\_DUE\_DLL\_NOT\_FOUND

GENERIC\_GDD\_CLS\_035\_040:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_DUE\_DLL\_NOT\_FOUND |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit due to overwritten file error |

#### DEVICE\_NAME\_E2GH

GENERIC\_GDD\_CLS\_035\_041:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2GH |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2GH |

#### DEVICE\_NAME\_E2GH\_E2xFCC2

GENERIC\_GDD\_CLS\_035\_042:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2GH\_E2xFCC2 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2GH\_E2xFCC2 |

#### DEVICE\_NAME\_E2GUH

GENERIC\_GDD\_CLS\_035\_043:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2GUH |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2GUH |

#### DEVICE\_NAME\_E2GUH\_E2xFCC2

GENERIC\_GDD\_CLS\_035\_044:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2GUH\_E2xFCC2 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2GUH\_E2xFCC2 |

#### DEVICE\_NAME\_E2H

GENERIC\_GDD\_CLS\_035\_045:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2H |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2H |

#### DEVICE\_NAME\_E2H\_E2xFCC2

GENERIC\_GDD\_CLS\_035\_046:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2H\_E2xFCC2 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2H\_E2xFCC2 |

#### DEVICE\_NAME\_E2UH

GENERIC\_GDD\_CLS\_035\_047:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2UH |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2UH |

#### DEVICE\_NAME\_E2UH\_E2xFCC2

GENERIC\_GDD\_CLS\_035\_048:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2UH\_E2xFCC2 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2UH\_E2xFCC2 |

#### DEVICE\_NAME\_ALL

GENERIC\_GDD\_CLS\_035\_049:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_ALL |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name of all. |

#### AR\_VERSION\_4\_2\_2

GENERIC\_GDD\_CLS\_035\_050:

|  |  |
| --- | --- |
| **Variable Name** | AR\_VERSION\_4\_2\_2 |
| **Type** | const string |
| **Range** | None |
| **Description** | AutoSar Version 4.2.2 |

#### AR\_VERSION\_ALL

GENERIC\_GDD\_CLS\_035\_051:

|  |  |
| --- | --- |
| **Variable Name** | AR\_VERSION\_ALL |
| **Type** | const string |
| **Range** | None |
| **Description** | AutoSar Version of all support |

#### VERSION\_1\_0\_0

GENERIC\_GDD\_CLS\_035\_052:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_1\_0\_0 |
| **Type** | const string |
| **Range** | None |
| **Description** | Version 1.0.0 |

#### VERSION\_ALL

GENERIC\_GDD\_CLS\_035\_053:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_ALL |
| **Type** | const string |
| **Range** | None |
| **Description** | Version of all support |

#### DEVICE\_NAME\_E2H\_E2xFCC2\_468

GENERIC\_GDD\_CLS\_035\_054:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2H\_E2xFCC2\_468 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2H\_E2xFCC2 with package 468 pins |

#### DEVICE\_NAME\_E2H\_E2xFCC2\_373

GENERIC\_GDD\_CLS\_035\_055:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2H\_E2xFCC2\_373 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2H\_E2xFCC2 with package 373 pins |

#### DEVICE\_NAME\_E2M\_373

GENERIC\_GDD\_CLS\_035\_056:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2M\_373 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2M with package 373 pins |

#### DEVICE\_NAME\_E2M\_292

GENERIC\_GDD\_CLS\_035\_057:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2M\_292 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2M with package 292 pins |

#### DEVICE\_NAME\_E2M\_E2xFCC1\_373

GENERIC\_GDD\_CLS\_035\_058:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2M\_E2xFCC1\_373 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2M\_E2xFCC1 with package 373 pins |

#### DEVICE\_NAME\_E2M\_E2xFCC1\_292

GENERIC\_GDD\_CLS\_035\_059:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2M\_E2xFCC1\_292 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2M\_E2xFCC1 with package 292 pins |

#### FILTER

GENERIC\_GDD\_CLS\_035\_060:

|  |  |
| --- | --- |
| **Variable Name** | FILTER |
| **Type** | const string |
| **Range** | None |
| **Description** | FILTER |

#### DEFINITION\_REF

GENERIC\_GDD\_CLS\_035\_061:

|  |  |
| --- | --- |
| **Variable Name** | DEFINITION\_REF |
| **Type** | const string |
| **Range** | None |
| **Description** | DEFINITION REF |

#### VALUE\_REF

GENERIC\_GDD\_CLS\_035\_062:

|  |  |
| --- | --- |
| **Variable Name** | VALUE\_REF |
| **Type** | const string |
| **Range** | None |
| **Description** | VALUE REF |

#### DEST

GENERIC\_GDD\_CLS\_035\_063:

|  |  |
| --- | --- |
| **Variable Name** | DEST |
| **Type** | const string |
| **Range** | None |
| **Description** | DEST |

#### MODULE\_DESCRIPTION\_REF

GENERIC\_GDD\_CLS\_035\_064:

|  |  |
| --- | --- |
| **Variable Name** | MODULE\_DESCRIPTION\_REF |
| **Type** | const string |
| **Range** | None |
| **Description** | Module description ref |

#### UUID

GENERIC\_GDD\_CLS\_035\_065:

|  |  |
| --- | --- |
| **Variable Name** | UUID |
| **Type** | const string |
| **Range** | None |
| **Description** | UUID |

#### PARAMETER\_VALUES

GENERIC\_GDD\_CLS\_035\_066:

|  |  |
| --- | --- |
| **Variable Name** | PARAMETER\_VALUES |
| **Type** | const string |
| **Range** | None |
| **Description** | Parameter Values |

#### REFERENCE\_VALUES

GENERIC\_GDD\_CLS\_035\_067:

|  |  |
| --- | --- |
| **Variable Name** | REFERENCE\_VALUES |
| **Type** | const string |
| **Range** | None |
| **Description** | Reference Values |

#### VENDOR\_API\_INFIX

GENERIC\_GDD\_CLS\_035\_068:

|  |  |
| --- | --- |
| **Variable Name** | VENDOR\_API\_INFIX |
| **Type** | const string |
| **Range** | None |
| **Description** | Vendor API infix |

#### STR\_ON

GENERIC\_GDD\_CLS\_035\_069:

|  |  |
| --- | --- |
| **Variable Name** | STR\_ON |
| **Type** | const string |
| **Range** | None |
| **Description** | STR\_ON |

#### STR\_OFF

GENERIC\_GDD\_CLS\_035\_070:

|  |  |
| --- | --- |
| **Variable Name** | STR\_OFF |
| **Type** | const string |
| **Range** | None |
| **Description** | STR\_OFF |

#### OUTPUT\_PATH

GENERIC\_GDD\_CLS\_035\_071:

|  |  |
| --- | --- |
| **Variable Name** | OUTPUT\_PATH |
| **Type** | const string |
| **Range** | None |
| **Description** | OUTPUT PATH |

#### FILTER\_NAME

GENERIC\_GDD\_CLS\_035\_072:

|  |  |
| --- | --- |
| **Variable Name** | FILTER\_NAME |
| **Type** | const string |
| **Range** | None |
| **Description** | FILTER NAME |

#### NAN

GENERIC\_GDD\_CLS\_035\_073:

|  |  |
| --- | --- |
| **Variable Name** | NAN |
| **Type** | const string |
| **Range** | None |
| **Description** | NAN |

#### DOUBLE\_MIN\_VALUE

GENERIC\_GDD\_CLS\_035\_074:

|  |  |
| --- | --- |
| **Variable Name** | DOUBLE\_MIN\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | Min value float point |

#### DOUBLE\_MAX\_VALUE

GENERIC\_GDD\_CLS\_035\_075:

|  |  |
| --- | --- |
| **Variable Name** | DOUBLE\_MAX\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | Max value float point |

#### VALUE

GENERIC\_GDD\_CLS\_035\_076:

|  |  |
| --- | --- |
| **Variable Name** | VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | Value string |

#### ECUC\_BOOLEAN\_PARAM\_DEF

GENERIC\_GDD\_CLS\_035\_077:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_BOOLEAN\_PARAM\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC BOOLEAN PARAM DEF attribute CDF |

#### ECUC\_INTEGER\_PARAM\_DEF

GENERIC\_GDD\_CLS\_035\_078:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_INTEGER\_PARAM\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_INTEGER\_PARAM\_DEF attribute CDF |

#### ECUC\_FLOAT\_PARAM\_DEF

GENERIC\_GDD\_CLS\_035\_079:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_FLOAT\_PARAM\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_FLOAT\_PARAM\_DEF attribute CDF |

#### ECUC\_STRING\_PARAM\_DEF

GENERIC\_GDD\_CLS\_035\_080:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_STRING\_PARAM\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_STRING\_PARAM\_DEF attribute CDF |

#### ECUC\_LINKER\_SYMBOL\_DEF

GENERIC\_GDD\_CLS\_035\_081:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_LINKER\_SYMBOL\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_LINKER\_SYMBOL\_DEF attribute CDF |

#### ECUC\_FUNCTION\_NAME\_DEF

GENERIC\_GDD\_CLS\_035\_082:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_FUNCTION\_NAME\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_FUNCTION\_NAME\_DEF attribute CDF |

#### ECUC\_ENUMERATION\_PARAM\_DEF

GENERIC\_GDD\_CLS\_035\_083:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_ENUMERATION\_PARAM\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_ENUMERATION\_PARAM\_DEF attribute CDF |

#### ECUC\_ENUMERATION\_LITERAL\_DEF

GENERIC\_GDD\_CLS\_035\_084:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_ENUMERATION\_LITERAL\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_ENUMERATION\_LITERAL\_DEF attribute CDF |

#### ECUC\_SYMBOLIC\_NAME\_REFERENCE\_DEF

GENERIC\_GDD\_CLS\_035\_085:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_SYMBOLIC\_NAME\_REFERENCE\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_SYMBOLIC\_NAME\_REFERENCE\_DEF attribute CDF |

#### ECUC\_URI\_REFERENCE\_DEF

GENERIC\_GDD\_CLS\_035\_086:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_URI\_REFERENCE\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_URI\_REFERENCE\_DEF attribute CDF |

#### ECUC\_ADD\_INFO\_PARAM\_DEF

GENERIC\_GDD\_CLS\_035\_087:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_ADD\_INFO\_PARAM\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC\_ADD\_INFO\_PARAM\_DEF attribute CDF |

#### INVALID\_ATTRIBUTE

GENERIC\_GDD\_CLS\_035\_088:

|  |  |
| --- | --- |
| **Variable Name** | INVALID\_ATTRIBUTE |
| **Type** | const int |
| **Range** | None |
| **Description** | Invalid Attribute |

#### DEVICE\_NAME\_E2GH\_468

GENERIC\_GDD\_CLS\_035\_089:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2GH\_468 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2GH with package 468 pins |

#### DEVICE\_NAME\_E2GH\_373

GENERIC\_GDD\_CLS\_035\_090:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2GH\_373 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2GH with package 373 pins |

#### DEVICE\_NAME\_E2GH\_E2xFCC2\_468

GENERIC\_GDD\_CLS\_035\_091:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2GH\_E2xFCC2\_468 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2GH\_E2xFCC2 with package 468 pins |

#### DEVICE\_NAME\_E2GH\_E2xFCC2\_373

GENERIC\_GDD\_CLS\_035\_092:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2GH\_E2xFCC2\_373 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2GH\_E2xFCC2 with package 373 pins |

#### DEVICE\_NAME\_E2H\_468

GENERIC\_GDD\_CLS\_035\_093:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2H\_468 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2H with package 468 pins |

#### DEVICE\_NAME\_E2H\_373

GENERIC\_GDD\_CLS\_035\_094:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_E2H\_373 |
| **Type** | const string |
| **Range** | None |
| **Description** | Device name E2H with package 373 pins |

#### INT\_TWO

GENERIC\_GDD\_CLS\_035\_095:

|  |  |
| --- | --- |
| **Variable Name** | INT\_TWO |
| **Type** | const int |
| **Range** | None |
| **Description** | None |

#### INT\_THREE

GENERIC\_GDD\_CLS\_035\_096:

|  |  |
| --- | --- |
| **Variable Name** | INT\_THREE |
| **Type** | const int |
| **Range** | None |
| **Description** | None |

#### INT\_FOUR

GENERIC\_GDD\_CLS\_035\_097:

|  |  |
| --- | --- |
| **Variable Name** | INT\_FOUR |
| **Type** | const int |
| **Range** | None |
| **Description** | FOUR numeric |

#### INSTANCE\_INDEX

GENERIC\_GDD\_CLS\_035\_098:

|  |  |
| --- | --- |
| **Variable Name** | INSTANCE\_INDEX |
| **Type** | const string |
| **Range** | None |
| **Description** | Instance index |

#### MAX\_LINE\_LENGTH

GENERIC\_GDD\_CLS\_035\_099:

|  |  |
| --- | --- |
| **Variable Name** | MAX\_LINE\_LENGTH |
| **Type** | const int |
| **Range** | None |
| **Description** | Max line length |

#### INDENT\_SECTION

GENERIC\_GDD\_CLS\_035\_100:

|  |  |
| --- | --- |
| **Variable Name** | INDENT\_SECTION |
| **Type** | const int |
| **Range** | None |
| **Description** | Indent section length |

#### OUTPUT

GENERIC\_GDD\_CLS\_035\_101:

|  |  |
| --- | --- |
| **Variable Name** | OUTPUT |
| **Type** | const string |
| **Range** | None |
| **Description** | OUTPUT |

#### IMPLEMENT\_PREFIX

GENERIC\_GDD\_CLS\_035\_102:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_PREFIX |
| **Type** | const string |
| **Range** | None |
| **Description** | Implement Prefix |

#### DESCRIPTION\_PREFIX

GENERIC\_GDD\_CLS\_035\_103:

|  |  |
| --- | --- |
| **Variable Name** | DESCRIPTION\_PREFIX |
| **Type** | const string |
| **Range** | None |
| **Description** | Implement Prefix |

#### DEVICE\_NAME\_U2A\_EVA

GENERIC\_GDD\_CLS\_035\_104:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_U2A\_EVA |
| **Type** | const string |
| **Range** | None |
| **Description** | Implement Prefix |

#### CDF\_EXTENSION

GENERIC\_GDD\_CLS\_035\_105:

|  |  |
| --- | --- |
| **Variable Name** | CDF\_EXTENSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CDF Extension |

#### TRANSLATION\_FILE\_EXTENSION

GENERIC\_GDD\_CLS\_035\_106:

|  |  |
| --- | --- |
| **Variable Name** | TRANSLATION\_FILE\_EXTENSION |
| **Type** | const string |
| **Range** | None |
| **Description** | Translation Extension |

#### NULL

GENERIC\_GDD\_CLS\_035\_107:

|  |  |
| --- | --- |
| **Variable Name** | NULL |
| **Type** | const string |
| **Range** | None |
| **Description** | NULL |

#### NULL\_PTR

GENERIC\_GDD\_CLS\_035\_108:

|  |  |
| --- | --- |
| **Variable Name** | NULL\_PTR |
| **Type** | const string |
| **Range** | None |
| **Description** | NULL\_PTR |

#### ON\_OFF\_PARAMS

GENERIC\_GDD\_CLS\_035\_109:

|  |  |
| --- | --- |
| **Variable Name** | ON\_OFF\_PARAMS |
| **Type** | const string |
| **Range** | None |
| **Description** | ON\_OFF\_PARAMS |

#### DEM\_EVENT\_PARAMS

GENERIC\_GDD\_CLS\_035\_110:

|  |  |
| --- | --- |
| **Variable Name** | DEM\_EVENT\_PARAMS |
| **Type** | const string |
| **Range** | None |
| **Description** | DEM\_EVENT\_PARAMS |

#### DEM\_CONF\_DEM\_EVENT\_PARAMETER

GENERIC\_GDD\_CLS\_035\_111:

|  |  |
| --- | --- |
| **Variable Name** | DEM\_CONF\_DEM\_EVENT\_PARAMETER |
| **Type** | const string |
| **Range** | None |
| **Description** | DEM\_CONF\_DEM\_EVENT\_PARAMETER |

#### MODULE\_NAME

GENERIC\_GDD\_CLS\_035\_112:

|  |  |
| --- | --- |
| **Variable Name** | MODULE\_NAME |
| **Type** | const string |
| **Range** | None |
| **Description** | MODULE\_NAME |

#### MICROFAMILY\_NAME

GENERIC\_GDD\_CLS\_035\_113:

|  |  |
| --- | --- |
| **Variable Name** | MICROFAMILY\_NAME |
| **Type** | const string |
| **Range** | None |
| **Description** | MICROFAMILY\_NAME |

#### MODULE\_REQUIRED

GENERIC\_GDD\_CLS\_035\_114:

|  |  |
| --- | --- |
| **Variable Name** | MODULE\_REQUIRED |
| **Type** | const string |
| **Range** | None |
| **Description** | MODULE\_REQUIRED |

#### PROJECT\_TITLE

GENERIC\_GDD\_CLS\_035\_115:

|  |  |
| --- | --- |
| **Variable Name** | PROJECT\_TITLE |
| **Type** | const string |
| **Range** | None |
| **Description** | PROJECT\_TITLE |

#### MODULE\_TOOL\_ID

GENERIC\_GDD\_CLS\_035\_116:

|  |  |
| --- | --- |
| **Variable Name** | MODULE\_TOOL\_ID |
| **Type** | const string |
| **Range** | None |
| **Description** | MODULE\_TOOL\_ID |

#### VERSION\_INFORMATION\_DATA

GENERIC\_GDD\_CLS\_035\_117:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_DATA |
| **Type** | const string |
| **Range** | None |
| **Description** | VERSION INFORMATION DATA |

#### CFG\_AR\_RELEASE\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_035\_118:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_AR\_RELEASE\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CFG AR RELEASE MAJOR VERSION |

#### CFG\_AR\_RELEASE\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_035\_119:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_AR\_RELEASE\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CFG AR RELEASE MINOR VERSION |

#### CFG\_AR\_RELEASE\_REVISION\_VERSION

GENERIC\_GDD\_CLS\_035\_120:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_AR\_RELEASE\_REVISION\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CFG AR RELEASE REVISION VERSION |

#### CFG\_SW\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_035\_121:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_SW\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CFG SW MAJOR VERSION |

#### CFG\_SW\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_035\_122:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_SW\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CFG SW MINOR VERSION |

#### AR\_RELEASE\_MAJOR\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_035\_123:

|  |  |
| --- | --- |
| **Variable Name** | AR\_RELEASE\_MAJOR\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | AR RELEASE MAJOR VERSION VALUE |

#### AR\_RELEASE\_MINOR\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_035\_124:

|  |  |
| --- | --- |
| **Variable Name** | AR\_RELEASE\_MINOR\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | AR RELEASE MINOR VERSION VALUE |

#### AR\_RELEASE\_REVISION\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_035\_125:

|  |  |
| --- | --- |
| **Variable Name** | AR\_RELEASE\_REVISION\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | AR RELEASE REVISION VERSION VALUE |

#### SW\_MAJOR\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_035\_126:

|  |  |
| --- | --- |
| **Variable Name** | SW\_MAJOR\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | SW MAJOR VERSION VALUE |

#### SW\_MINOR\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_035\_127:

|  |  |
| --- | --- |
| **Variable Name** | SW\_MINOR\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | SW MINOR VERSION VALUE |

#### SW\_PATCH\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_035\_128:

|  |  |
| --- | --- |
| **Variable Name** | SW\_PATCH\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | SW PATCH VERSION VALUE |

#### VENDOR\_ID\_VALUE

GENERIC\_GDD\_CLS\_035\_129:

|  |  |
| --- | --- |
| **Variable Name** | VENDOR\_ID\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | VENDOR ID VALUE |

#### MODULE\_ID\_VALUE

GENERIC\_GDD\_CLS\_035\_130:

|  |  |
| --- | --- |
| **Variable Name** | MODULE\_ID\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | MODULE ID VALUE |

#### CBK\_AR\_RELEASE\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_035\_131:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_AR\_RELEASE\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CBK AR RELEASE MAJOR VERSION |

#### CBK\_AR\_RELEASE\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_035\_132:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_AR\_RELEASE\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CBK AR RELEASE MINOR VERSION |

#### CBK\_AR\_RELEASE\_REVISION\_VERSION

GENERIC\_GDD\_CLS\_035\_133:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_AR\_RELEASE\_REVISION\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CBK AR RELEASE REVISION VERSION |

#### CBK\_SW\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_035\_134:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_SW\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CBK SW MAJOR VERSION |

#### CBK\_SW\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_035\_135:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_SW\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | CBK SW MINOR VERSION |

#### PBCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_035\_136:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | PBCFG C AR RELEASE MAJOR VERSION |

#### PBCFG\_C\_AR\_RELEASE\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_035\_137:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_AR\_RELEASE\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | PBCFG C AR RELEASE MINOR VERSION |

#### PBCFG\_C\_AR\_RELEASE\_REVISION\_VERSION

GENERIC\_GDD\_CLS\_035\_138:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_AR\_RELEASE\_REVISION\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | PBCFG C AR RELEASE REVISION VERSION |

#### PBCFG\_C\_SW\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_035\_139:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_SW\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | PBCFG C SW MAJOR VERSION |

#### PBCFG\_C\_SW\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_035\_140:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_SW\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | PBCFG C SW MINOR VERSION |

#### VERSION\_INFORMATION\_CBK

GENERIC\_GDD\_CLS\_035\_141:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_CBK |
| **Type** | const string |
| **Range** | None |
| **Description** | VERSION INFORMATION CBK |

#### VERSION\_INFORMATION\_PBCFG

GENERIC\_GDD\_CLS\_035\_142:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_PBCFG |
| **Type** | const string |
| **Range** | None |
| **Description** | VERSION INFORMATION PBCFG |

#### CALL\_BACK

GENERIC\_GDD\_CLS\_035\_143:

|  |  |
| --- | --- |
| **Variable Name** | CALL\_BACK |
| **Type** | const string |
| **Range** | None |
| **Description** | CALL BACK |

#### POST\_BUILD

GENERIC\_GDD\_CLS\_035\_144:

|  |  |
| --- | --- |
| **Variable Name** | POST\_BUILD |
| **Type** | const string |
| **Range** | None |
| **Description** | POST BUILD |

#### VERSION\_INFORMATION\_CFG

GENERIC\_GDD\_CLS\_035\_145:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_CFG |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

#### LCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_035\_146:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

#### LCFG\_C\_AR\_RELEASE\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_035\_147:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_AR\_RELEASE\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

#### LCFG\_C\_AR\_RELEASE\_REVISION\_VERSION

GENERIC\_GDD\_CLS\_035\_148:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_AR\_RELEASE\_REVISION\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

#### LCFG\_C\_SW\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_035\_149:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_SW\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

#### LCFG\_C\_SW\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_035\_150:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_SW\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

#### LINK\_TIME

GENERIC\_GDD\_CLS\_035\_151:

|  |  |
| --- | --- |
| **Variable Name** | LINK\_TIME |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

#### VERSION\_INFORMATION\_LT

GENERIC\_GDD\_CLS\_035\_152:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_LT |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

#### BoolDataTypeMarked

GENERIC\_GDD\_CLS\_035\_153:

|  |  |
| --- | --- |
| **Variable Name** | BoolDataTypeMarked |
| **Type** | readonly string |
| **Range** | None |
| **Description** | None |

#### FloatDataTypeMarked

GENERIC\_GDD\_CLS\_035\_154:

|  |  |
| --- | --- |
| **Variable Name** | FloatDataTypeMarked |
| **Type** | readonly string |
| **Range** | None |
| **Description** | None |

#### VendorIdPosition

GENERIC\_GDD\_CLS\_035\_155:

|  |  |
| --- | --- |
| **Variable Name** | VendorIdPosition |
| **Type** | readonly int |
| **Range** | None |
| **Description** | None |

#### ModuleIdPosition

GENERIC\_GDD\_CLS\_035\_156:

|  |  |
| --- | --- |
| **Variable Name** | ModuleIdPosition |
| **Type** | readonly int |
| **Range** | None |
| **Description** | None |

#### SwMajorVersionPosition

GENERIC\_GDD\_CLS\_035\_157:

|  |  |
| --- | --- |
| **Variable Name** | SwMajorVersionPosition |
| **Type** | readonly int |
| **Range** | None |
| **Description** | None |

#### SwMinorVersionPosition

GENERIC\_GDD\_CLS\_035\_158:

|  |  |
| --- | --- |
| **Variable Name** | SwMinorVersionPosition |
| **Type** | readonly int |
| **Range** | None |
| **Description** | None |

#### INT\_ZERO

GENERIC\_GDD\_CLS\_035\_159:

|  |  |
| --- | --- |
| **Variable Name** | INT\_ZERO |
| **Type** | readonly int |
| **Range** | None |
| **Description** | None |

#### INT\_ONE

GENERIC\_GDD\_CLS\_035\_160:

|  |  |
| --- | --- |
| **Variable Name** | INT\_ONE |
| **Type** | readonly int |
| **Range** | None |
| **Description** | None |

### Methods

## Renesas::Generator::MCALConfGen::CrossCutting::Util::StringUtils

|  |  |
| --- | --- |
| **Location File Name** | StringUtils.cs |
| **Description** | This class contains the util functions for processing string. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | FindString | Public | Find substring in text that match a regular expression |
| (2) | GetElementLastInArray | Public | Get last element when spliting path into substrings by delimeter |
| (3) | ParseHexToInt | Public | Convert hex string to integer |
| (4) | ParseHexToUInt | Public | Convert hex string to unsigned integer |
| (5) | GetStringFromTemplate | Public | Replace placeholders in a string |
| (6) | IsHex | Public | Check if string is in hexa format |
| (7) | GetRelativePath | Public | Get relative path from current directory |
| (8) | IsInvalidName | Public | Check invalid name |
| (9) | CompareShortName | Public | Compare 2 short name for sorting |
| (10) | IsNullPointer | Public | Check whether value is null pointer or not |
| (11) | IsNameSyntaxAsC | Public | Check whether variable name follow to C coding rule or not. |
| (12) | replaceStringByTemplate | Private | Replace placeholder in a string |
| (13) | appendDirectorySeparatorChar | Private | Append directory separator "\" to a path |
| (14) | parseShortName | Private | Parse short name in CDF to 2 parts: name and surfix number |

### Fields

### Methods

#### CompareShortName

GENERIC\_GDD\_CLS\_036\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CompareShortName | | |
| **Arguments** | shrName1 | short name for comparation | |
| Range: | None |
| shrName2 | short name for comparation | |
| Range: | None |
| **Return** | int | Indicates the lexical relationship between the two short name | |
| Range: | Less than 0 | 0 | Greater than 0 |
| **Generated Value** | None | | |
| **Description** | Compare 2 short name for sorting  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET name1, name2 = ""  LET num1, num2 = 0  LET name1, name2, num1, num2 = CALL parseShortName WITH name1, name2, num1, num2  IF name1 == name2  RETURN num1.compare(num2)  ELSE  RETURN name1.compare(name2) | | |

{Ref: N/A}

#### FindString

GENERIC\_GDD\_CLS\_036\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | FindString | | |
| **Arguments** | text | Text | |
| Range: | None |
| expression | Expression | |
| Range: | None |
| **Return** | string | The string is found in the test | |
| Range: | Null/!=Null |
| **Generated Value** | None | | |
| **Description** | Find substring in text that match a regular expression  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF text or expr is NULL or empty THEN  RETURN emtpy  ELSE  IF found 1 or more than 1 of matched substring THEN  RETURN the last found item  ELSE  RETURN empty | | |

{Ref: N/A}

#### GetElementLastInArray

GENERIC\_GDD\_CLS\_036\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetElementLastInArray | | |
| **Arguments** | path | Path | |
| Range: | None |
| delimeter | Delimeter | |
| Range: | None |
| **Return** | string | The last string is found in the path | |
| Range: | Null/!=Null |
| **Generated Value** | None | | |
| **Description** | Get last element when spliting path into substrings by delimeter  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF path is not null THEN  LET arr = Split path into substrings by delimeter.  IF Length off arr > 1 THEN  RETURN last substring.  ELSE  RETURN empty  RETURN empty | | |

{Ref: N/A}

#### GetRelativePath

GENERIC\_GDD\_CLS\_036\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetRelativePath | | |
| **Arguments** | toPath | To path | |
| Range: | None |
| **Return** | string | The relative path | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Get relative path from current directory  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET fromPath = CALL GetCurrentDirectory  LET fromUri = initialize new Uri WITH CALL appendDirectorySeparatorChar WITH fromPath  LET toPath = CALL GetFullPath WITH toPath  LET toUri = initialize new Uri WITH CALL appendDirectorySeparatorChar WITH toPath  LET ret = ""  IF fromUri.Scheme != toUri.Scheme THEN  LET ret = toPath  ELSE  LET relativeUri = fromUri.MakeRelativeUri WITH toUri  LET relativePath = CALL UnescapeDataString WITH relativeUri  IF relativePath is not null THEN  LET relativePath = '.' + DirectorySeparatorChar  IF toUri.Scheme == Uri.UriSchemeFile THEN  LET relativePath = REPLACE AltDirectorySeparatorChar by DirectorySeparatorChar in relativePath  LET ret = relativePath  RETURN ret | | |

{Ref: N/A}

#### GetStringFromTemplate

GENERIC\_GDD\_CLS\_036\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetStringFromTemplate | | |
| **Arguments** | inputString | Input string | |
| Range: | None |
| dictionary | Mapping placeholders and their values | |
| Range: | None |
| **Return** | string | The string has been replace | |
| Range: | Null/!=Null |
| **Generated Value** | None | | |
| **Description** | Replace placeholders in a string  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   FOR each placeholder in dictionary  LET Replace placeholder in inputString by its value defined in dictionary  END | | |

{Ref: N/A}

#### IsHex

GENERIC\_GDD\_CLS\_036\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsHex | | |
| **Arguments** | value | Value | |
| Range: | None |
| **Return** | bool | The status of string is in hexa format or not | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check if string is in hexa format  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Using regular expression to check if string is in hexa format | | |

{Ref: N/A}

#### IsInvalidName

GENERIC\_GDD\_CLS\_036\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsInvalidName | | |
| **Arguments** | name | Name | |
| Range: | None |
| **Return** | bool | The status of name (invalid or not) | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check invalid name  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET result = false  IF name is null or empty or name starts with "-" or name contains "\*", "?" or "|" THEN  LET result = true  RETURN result | | |

{Ref: N/A}

#### IsNameSyntaxAsC

GENERIC\_GDD\_CLS\_036\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsNameSyntaxAsC | | |
| **Arguments** | name | A variable name | |
| Range: | None |
| **Return** | bool | The status of value | |
| Range: | Less than 0 | 0 | Greater than 0 |
| **Generated Value** | None | | |
| **Description** | Check whether variable name follow to C coding rule or not.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET ret = false  LET regex = new Regex("^[a-zA-Z][a-zA-Z0-9\_]\*$")  LET cLanguageKeywordList = list of key word C#  IF name is not null and cLanguageKeywordList not contain name  LET ret = reg.Match(name).Success  RETURN ret | | |

{Ref: N/A}

#### IsNullPointer

GENERIC\_GDD\_CLS\_036\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsNullPointer | | |
| **Arguments** | value | A value of parameter | |
| Range: | None |
| **Return** | bool | The status of value | |
| Range: | Less than 0 | 0 | Greater than 0 |
| **Generated Value** | None | | |
| **Description** | Check whether value is null pointer or not  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF string value is null, null\_ptr, empty THEN  RETURN true  ELSE  RETURN false | | |

{Ref: N/A}

#### ParseHexToInt

GENERIC\_GDD\_CLS\_036\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseHexToInt | | |
| **Arguments** | inputString | Input string | |
| Range: | None |
| **Return** | int | The value with type is Int | |
| Range: | 0..Int32.Max |
| **Generated Value** | None | | |
| **Description** | Convert hex string to integer  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF inputString is not null THEN  LET inputString = Upper case inputString  LET inputString = Remove all "0x", "0X", "U" and "L" characters from inputString  LET Using C# util function to convert inputString to integer value  RETURN value | | |

{Ref: N/A}

#### ParseHexToUInt

GENERIC\_GDD\_CLS\_036\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseHexToUInt | | |
| **Arguments** | inputString | Input string | |
| Range: | None |
| **Return** | uint | The value with type is UInt | |
| Range: | 0..Int32.Max |
| **Generated Value** | None | | |
| **Description** | Convert hex string to unsigned integer  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF inputString is not null THEN  LET inputString = Upper case inputString  LET inputString = Remove all "0x", "0X", "U" and "L" characters from inputString  LET Using C# util function to convert inputString to unsigned integer value  RETURN value | | |

{Ref: N/A}

#### appendDirectorySeparatorChar

GENERIC\_GDD\_CLS\_036\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | appendDirectorySeparatorChar | | |
| **Arguments** | path | Path | |
| Range: | None |
| **Return** | string | The string has been added separator '\' | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Append directory separator "\" to a path  Algorithm: GLOBAL VARIABLE IN:  Path   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   IF path doesn't have extension at the end AND  path doesn't have directory separator "\" at the end THEN  LET Append directory separator "\" to a path | | |

{Ref: N/A}

#### parseShortName

GENERIC\_GDD\_CLS\_036\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseShortName | | |
| **Arguments** | shrName | short name in CDF | |
| Range: | None |
| name | name of parameter | |
| Range: | None |
| number | surfix number | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Parse short name in CDF to 2 parts: name and surfix number  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET match = match shrName WITH regex @"(\d+)$"  IF match is Success  LET number = first element of match.Groups  LET name = replace number by "" in shrName  ELSE  LET number = 0  LET name = shrName | | |

{Ref: N/A}

#### replaceStringByTemplate

GENERIC\_GDD\_CLS\_036\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | replaceStringByTemplate | | |
| **Arguments** | inputString | Dictionary <string | |
| Range: | None |
| dictionary | Dictionary | |
| Range: | None |
| **Return** | string | The string has been aligned | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Replace placeholder in a string  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET ret = ""  LET temp = REPLACE "$" by "" in inputString and split by ":"  LET item = Get first element in array temp  LET spaces = Get second element in array temp  LET alignment = GET third element in array temp  LET suffix = GET four element in array temp  LET spaces = spaces - length of suffix  IF alignment is "C" THEN  LET haftSpaces = (spaces - text.Length) / 2  LET ret = " "\*haftSpaces + text + " "\*((spaces - text.Length) - haftSpaces)  ELSE IF alignment is "L" THEN  LET format = String.Format("{{0,-{0}}}", spaces)  LET ret = String.Format(format, text)  ELSE  LET string format = String.Format("{{0,{0}}}", spaces)  LET ret = String.Format(format, text)  RETURN ret | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::BasicConfiguration::BasicConfiguration

|  |  |
| --- | --- |
| **Location File Name** | BasicConfiguration.cs |
| **Description** | This class implements |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | ModuleName | Public | Module name |
| (3) | MicroFamilyName | Public | Micro family name |
| (4) | AutoSARVersion | Public | AutoSAR version |
| (5) | DeviceNames | Public | Device names |
| (6) | VendorId | Public | Device names |
| (7) | ExeDirectory | Public | Executable directory |
| (8) | ModuleId | Public | Module id |
| (9) | OutputFolder | Public | Output folder |
| (10) | ExecutionName | Public | Execution name |
| (11) | ModuleRequired | Public | Modules required |
| (12) | ProjectTitle | Public | Project title |
| (13) | InstanceIndex | Public | Instance index |
| (14) | HasInstanceSetting | Public | Enable/Disable multiple instance |
| (15) | InstanceOutType | Public | Output type: Files or Folders |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ToInstanceValue | Public | Get index instance if has any instance setting |
| (2) | AppendInstanceToFileName | Public | Append instance to file name |
| (3) | BasicConfiguration | Private | Basic configuration constructor |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_037\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly BasicConfiguration |
| **Range** | None |
| **Description** | None |

#### ModuleName

GENERIC\_GDD\_CLS\_037\_002:

|  |  |
| --- | --- |
| **Variable Name** | ModuleName |
| **Type** | string |
| **Range** | None |
| **Description** | Module name |

#### MicroFamilyName

GENERIC\_GDD\_CLS\_037\_003:

|  |  |
| --- | --- |
| **Variable Name** | MicroFamilyName |
| **Type** | string |
| **Range** | None |
| **Description** | Micro family name |

#### AutoSARVersion

GENERIC\_GDD\_CLS\_037\_004:

|  |  |
| --- | --- |
| **Variable Name** | AutoSARVersion |
| **Type** | string |
| **Range** | None |
| **Description** | AutoSAR version |

#### DeviceNames

GENERIC\_GDD\_CLS\_037\_005:

|  |  |
| --- | --- |
| **Variable Name** | DeviceNames |
| **Type** | List< string > |
| **Range** | None |
| **Description** | Device names |

#### VendorId

GENERIC\_GDD\_CLS\_037\_006:

|  |  |
| --- | --- |
| **Variable Name** | VendorId |
| **Type** | int |
| **Range** | None |
| **Description** | Device names |

#### ExeDirectory

GENERIC\_GDD\_CLS\_037\_007:

|  |  |
| --- | --- |
| **Variable Name** | ExeDirectory |
| **Type** | string |
| **Range** | None |
| **Description** | Executable directory |

#### ModuleId

GENERIC\_GDD\_CLS\_037\_008:

|  |  |
| --- | --- |
| **Variable Name** | ModuleId |
| **Type** | int |
| **Range** | None |
| **Description** | Module id |

#### OutputFolder

GENERIC\_GDD\_CLS\_037\_009:

|  |  |
| --- | --- |
| **Variable Name** | OutputFolder |
| **Type** | string |
| **Range** | None |
| **Description** | Output folder |

#### ExecutionName

GENERIC\_GDD\_CLS\_037\_010:

|  |  |
| --- | --- |
| **Variable Name** | ExecutionName |
| **Type** | string |
| **Range** | None |
| **Description** | Execution name |

#### ModuleRequired

GENERIC\_GDD\_CLS\_037\_011:

|  |  |
| --- | --- |
| **Variable Name** | ModuleRequired |
| **Type** | List< string > |
| **Range** | None |
| **Description** | Modules required |

#### ProjectTitle

GENERIC\_GDD\_CLS\_037\_012:

|  |  |
| --- | --- |
| **Variable Name** | ProjectTitle |
| **Type** | string |
| **Range** | None |
| **Description** | Project title |

#### InstanceIndex

GENERIC\_GDD\_CLS\_037\_013:

|  |  |
| --- | --- |
| **Variable Name** | InstanceIndex |
| **Type** | int |
| **Range** | None |
| **Description** | Instance index |

#### HasInstanceSetting

GENERIC\_GDD\_CLS\_037\_014:

|  |  |
| --- | --- |
| **Variable Name** | HasInstanceSetting |
| **Type** | bool |
| **Range** | None |
| **Description** | Enable/Disable multiple instance |

#### InstanceOutType

GENERIC\_GDD\_CLS\_037\_015:

|  |  |
| --- | --- |
| **Variable Name** | InstanceOutType |
| **Type** | InstanceOutputType |
| **Range** | None |
| **Description** | Output type: Files or Folders |

### Methods

#### AppendInstanceToFileName

GENERIC\_GDD\_CLS\_037\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AppendInstanceToFileName | | |
| **Arguments** | name | Name of file name | |
| Range: | None |
| **Return** | string: | return string after a appending instance to file name | |
| Range: | string.Empty/Valid string |
| **Generated Value** | None | | |
| **Description** | Append instance to file name  Algorithm: LET define result  IF InstanceOutType is equal to InstanceOutputType.FILES THEN  LET file = CALLl Path.GetFileNameWithoutExtension WITH name  LET exts = CALL Path.GetExtension WITH name  LET CALL string.Format WITH "{0}{1}{2}", file, CALL ToInstanceValue, extss  Return result | | |

{Ref: N/A}

#### ToInstanceValue

GENERIC\_GDD\_CLS\_037\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ToInstanceValue | | |
| **Arguments** | None | | |
| **Return** | string: | return instance value type of string | |
| Range: | string.Empty/Valid string |
| **Generated Value** | None | | |
| **Description** | Get index instance if has any instance setting  Algorithm: LET result = string.Empty  IF has any instance setting THEN  LET result = "\_" + CALL InstanceIndex.ToString  Return result | | |

{Ref: N/A}

#### BasicConfiguration

GENERIC\_GDD\_CLS\_037\_018:

|  |  |
| --- | --- |
| **Method Name** | BasicConfiguration |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Basic configuration constructor |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::CDFData::BaseRegisterProcessing

|  |  |
| --- | --- |
| **Location File Name** | BaseRegisterProcessing.cs |
| **Description** | This |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | macroLabelValue | Private | This field store macros in translation file |
| (3) | macroAddressValue | Private | This field store macros in device file |
| (4) | logger | Private | This field isn't used, it is reserved |
| (5) | userInterface | Private | This field report error and exit |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetMacroAddressValue | Public | Get macro address value |
| (2) | GetMacroLabelValue | Public | Get macro label value |
| (3) | IsRegisterExist | Public | Is register exist |
| (4) | SaveMacroAddressValue | Public | Parse input file and save macro address value |
| (5) | SaveMacroLabelValue | Public | Parse file and save macro label value |
| (6) | BaseRegisterProcessing | Protected | Base register processing constructor |
| (7) | BaseRegisterProcessing | Protected | Base register processing constructor |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_038\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly BaseRegisterProcessing |
| **Range** | None |
| **Description** | None |

#### macroLabelValue

GENERIC\_GDD\_CLS\_038\_002:

|  |  |
| --- | --- |
| **Variable Name** | macroLabelValue |
| **Type** | IDictionary< string, string > |
| **Range** | None |
| **Description** | This field store macros in translation file |

#### macroAddressValue

GENERIC\_GDD\_CLS\_038\_003:

|  |  |
| --- | --- |
| **Variable Name** | macroAddressValue |
| **Type** | IDictionary< string, string > |
| **Range** | None |
| **Description** | This field store macros in device file |

#### logger

GENERIC\_GDD\_CLS\_038\_004:

|  |  |
| --- | --- |
| **Variable Name** | logger |
| **Type** | ILogger |
| **Range** | None |
| **Description** | This field isn't used, it is reserved |

#### userInterface

GENERIC\_GDD\_CLS\_038\_005:

|  |  |
| --- | --- |
| **Variable Name** | userInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | This field report error and exit |

### Methods

#### GetMacroAddressValue

GENERIC\_GDD\_CLS\_038\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetMacroAddressValue | | |
| **Arguments** | macroValue | Macro value | |
| Range: | Null/valid string |
| **Return** | string | The macro address values | |
| Range: | empty/valid string |
| **Generated Value** | None | | |
| **Description** | Get macro address value  Algorithm: LET address = string empty  IF macroValue is contained in macroAddressValue THEN  LET address = value in macroAddressValue with key is macroValue  ELSE  LET record error ERR000023 into userInterface and exit  IF address is null or empty THEN  LET record error ERR000024 into userInterface and exit  ELSE IF address isn't hex format THEN  LET record error ERR000038 into userInterface and exit  RETURN address | | |

{Ref: [3] <CMN\_GDD\_ERR\_023, CMN\_GDD\_ERR\_024, CMN\_GDD\_ERR\_038>}

#### GetMacroLabelValue

GENERIC\_GDD\_CLS\_038\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetMacroLabelValue | | |
| **Arguments** | macroDefinition | Macro definition | |
| Range: | null/valid string |
| **Return** | string | The macro label value | |
| Range: | empty/valid string |
| **Generated Value** | None | | |
| **Description** | Get macro label value  Algorithm: LET macroValue = string empty  IF macroDefinition is contained in macroLabelValue THEN  LET macroValue = value in macroLabelValue with key is macroDefinition  ELSE  LET record error ERR000021 into userInterface and exit  IF macroValue is null or empty THEN  LET record error ERR000022 into userInterface and exit  RETURN macroValue | | |

{Ref: [1] <GENERIC\_TUM\_U2X\_GT\_ERR\_021 GENERIC\_TUM\_U2X\_GT\_ERR\_022>}

{Ref: [3] < CMN\_GDD\_ERR\_021, CMN\_GDD\_ERR\_022>}

#### IsRegisterExist

GENERIC\_GDD\_CLS\_038\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsRegisterExist | | |
| **Arguments** | macroDefinition | Macro definition | |
| Range: | null/valid string |
| **Return** | bool | The status of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Is register exist  Algorithm: LET ret = false  IF macroDefinition is contained in macroLabelValue THEN  LET ret = true  RETURN ret | | |

{Ref: N/A}

#### SaveMacroAddressValue

GENERIC\_GDD\_CLS\_038\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveMacroAddressValue | | |
| **Arguments** | filesInput | File name | |
| Range: | null/valid string |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Parse input file and save macro address value  Algorithm: IF file NOT existed THEN  LET record error ERR000001 into userInterface and exit  Else  FOREACH line IN file  IF line isn't a comment THEN  LET parse macro value and address  Add macro value and address to collection macroAddressValue  End  End | | |

{Ref: [3] <CMN\_GDD\_ERR\_001, CMN\_GDD\_ERR\_037>}

#### SaveMacroLabelValue

GENERIC\_GDD\_CLS\_038\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveMacroLabelValue | | |
| **Arguments** | fileName | File name | |
| Range: | None |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Parse file and save macro label value  Algorithm: If file NOT existed  Record error ERR000001 and exit  Else  For each line in file  If line NOT a comment  Parse macro label and value  Add macro label and value to collection macroLabelValue  End  End   If NO item found in collection macroLabelValue  Record error ERR000020 and exit  End | | |

{Ref: [3] <CMN\_GDD\_ERR\_001, CMN\_GDD\_ERR\_020>}

#### BaseRegisterProcessing

GENERIC\_GDD\_CLS\_038\_011:

|  |  |
| --- | --- |
| **Method Name** | BaseRegisterProcessing |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Base register processing constructor |

{Ref: N/A}

#### BaseRegisterProcessing

GENERIC\_GDD\_CLS\_038\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseRegisterProcessing | | |
| **Arguments** | logger | Logger | |
| Range: | Null/Valid value |
| userInterface | User interface | |
| Range: | Null/Valid value |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base register processing constructor | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::CDFData::CdfDataDictionary

|  |  |
| --- | --- |
| **Location File Name** | CdfDataDictionary.cs |
| **Description** | This |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | dataContext | Private | A 2-levels dictionary current context which store multiple instances data Instance short name 1 ------ |
| (3) | currentInstanceIndex | Private | None |
| (4) | moduleInstances | Private | Mapping between 1 moduleName ------ |
| (5) | configurations | Private | This field store Configurations |
| (6) | cdfFiles | Private | This field store CDF files |
| (7) | bswConfigs | Private | This field store Basic software configuration |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | SaveContainer | Public | Add a container of a module 2-levels dictionary dataContext Instance short name 1 ------ |
| (2) | GetCurrentInstanceContainers | Public | Get all containers of a module |
| (3) | GetAllInstanceContainers | Public | Get all containers of a module Get all instance based on module name Get all container in each instance |
| (4) | SaveConfiguration | Public | Save configuration |
| (5) | GetCurrentInstanceConfiguration | Public | Get configuration by module name |
| (6) | GetAllInstanceConfigurations | Public | Get configuration by module name With each instance, get all configuration based on module name |
| (7) | SaveBswConfiguration | Public | Save bsw configuration |
| (8) | GetBswConfiguration | Public | Get the first bsw configuration if module name does not input |
| (9) | GetBswConfigurations | Public | Get all bsw configurations |
| (10) | GetModules | Public | Get all modules' name from dataContext |
| (11) | SaveCdfFiles | Public | Save CDF files |
| (12) | GetCdfFile | Public | Get CDF file |
| (13) | GetContainerByUUID | Public | Get container by UUID |
| (14) | PrepareInstance | Public | Update current instance index |
| (15) | CdfDataDictionary | Private | Cdf data dictionary constructor |
| (16) | getContainerByShortName | Private | Get container by short name |
| (17) | addInstanceIfNeeded | Private | Add instance if needed |
| (18) | isCurrentInstanceAvailable | Private | Checking instance is current instance |
| (19) | getCurrentInstanceKey | Private | Get current instance key |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_039\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly CdfDataDictionary |
| **Range** | None |
| **Description** | None |

#### dataContext

GENERIC\_GDD\_CLS\_039\_002:

|  |  |
| --- | --- |
| **Variable Name** | dataContext |
| **Type** | Dictionary< string, Dictionary< string, Container |
| **Range** | None |
| **Description** | A 2-levels dictionary current context which store multiple instances data Instance short name 1 ------ |

#### currentInstanceIndex

GENERIC\_GDD\_CLS\_039\_003:

|  |  |
| --- | --- |
| **Variable Name** | currentInstanceIndex |
| **Type** | int |
| **Range** | None |
| **Description** | None |

#### moduleInstances

GENERIC\_GDD\_CLS\_039\_004:

|  |  |
| --- | --- |
| **Variable Name** | moduleInstances |
| **Type** | Dictionary< string, List< string > > |
| **Range** | None |
| **Description** | Mapping between 1 moduleName ------ |

#### configurations

GENERIC\_GDD\_CLS\_039\_005:

|  |  |
| --- | --- |
| **Variable Name** | configurations |
| **Type** | Dictionary< string, Configuration |
| **Range** | None |
| **Description** | This field store Configurations |

#### cdfFiles

GENERIC\_GDD\_CLS\_039\_006:

|  |  |
| --- | --- |
| **Variable Name** | cdfFiles |
| **Type** | Dictionary< string, string > |
| **Range** | None |
| **Description** | This field store CDF files |

#### bswConfigs

GENERIC\_GDD\_CLS\_039\_007:

|  |  |
| --- | --- |
| **Variable Name** | bswConfigs |
| **Type** | Dictionary< string, BswConfig |
| **Range** | None |
| **Description** | This field store Basic software configuration |

### Methods

#### GetAllInstanceConfigurations

GENERIC\_GDD\_CLS\_039\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAllInstanceConfigurations | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | IList<Configuration> | The list of module configurations | |
| Range: | empty/valid list Configuration |
| **Generated Value** | None | | |
| **Description** | Get configuration by module name With each instance, get all configuration based on module name  Algorithm: LET result = list Configuration empty  IF moduleInstances contains moduleName AND value of moduleInstances with key moduleName isn't null THEN  FOREACH each IN value of moduleInstances with key moduleName is a list string  LET result add value of configurations with Key is each  RETURN result | | |

{Ref: N/A}

#### GetAllInstanceContainers

GENERIC\_GDD\_CLS\_039\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAllInstanceContainers | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | Dictionary<string, IList<Container>> | The collection of containers | |
| Range: | empty/valid Dictionary |
| **Generated Value** | None | | |
| **Description** | Get all containers of a module Get all instance based on module name Get all container in each instance  Algorithm: LET result = Dictionary empty  IF moduleInstances contains moduleName AND value of moduleInstances with key moduleName isn't null THEN  FOREACH key IN value of moduleInstances with key moduleName is a list string  IF containerList isn't null THEN  LET containerList = value of dataContext with key  FOREACH container IN containerList  LET add container and childs of container into eachInstance  LET result add data with Key is key, Value is eachInstance  RETURN result | | |

{Ref: N/A}

#### GetBswConfiguration

GENERIC\_GDD\_CLS\_039\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetBswConfiguration | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | BswConfig | Basic software configuration info | |
| Range: | null/valid value |
| **Generated Value** | None | | |
| **Description** | Get the first bsw configuration if module name does not input  Algorithm: IF module is null or empty THEN  IF have any config in bswConfigs THEN  RETURN first config  Else  IF found moduleName in bswConfigs THEN  RETURN first found  RETURN null | | |

{Ref: N/A}

#### GetBswConfigurations

GENERIC\_GDD\_CLS\_039\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetBswConfigurations | | |
| **Arguments** | None | | |
| **Return** | BswConfig | List of basic software configuration info | |
| Range: | null/valid value |
| **Generated Value** | None | | |
| **Description** | Get all bsw configurations  Algorithm: RETURN all bsw configurations of bswConfigs | | |

{Ref: N/A}

#### GetCdfFile

GENERIC\_GDD\_CLS\_039\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetCdfFile | | |
| **Arguments** | shortName | Short name | |
| Range: | null/valid string |
| **Return** | string | Cdf file | |
| Range: | empty/valid value |
| **Generated Value** | None | | |
| **Description** | Get CDF file  Algorithm: IF shortName is NOT null or empty THEN  IF found short name exists in cdfFiles THEN  RETURN first found  RETURN empty | | |

{Ref: N/A}

#### GetContainerByUUID

GENERIC\_GDD\_CLS\_039\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainerByUUID | | |
| **Arguments** | moduleName | Short name | |
| Range: | null/valid string |
| uuid | Uuid | |
| Range: | null/valid string |
| **Return** | Container | The container | |
| Range: | null/valid value |
| **Generated Value** | None | | |
| **Description** | Get container by UUID  Algorithm: IF any of containers under moduleName has Uuid that equal to uuid THEN  RETURN first found container  RETURN null | | |

{Ref: N/A}

#### GetCurrentInstanceConfiguration

GENERIC\_GDD\_CLS\_039\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetCurrentInstanceConfiguration | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | Configuration | The module configuration | |
| Range: | null/valid value |
| **Generated Value** | None | | |
| **Description** | Get configuration by module name  Algorithm: IF found moduleName (or "{moduleName}d\*$") in configurations THEN  RETURN first found  Return null | | |

{Ref: N/A}

#### GetCurrentInstanceContainers

GENERIC\_GDD\_CLS\_039\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetCurrentInstanceContainers | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | IList<Container> | The list of container | |
| Range: | empty/valid list Container |
| **Generated Value** | None | | |
| **Description** | Get all containers of a module  Algorithm: IF moduleName is NOT null or empty THEN  LET containerList = get all containers that are right under moduleName (or "{moduleName}d\*$")  to add to a containers collection "result"  FOREACH container IN containerList  LET get all of its children containers to add to "result"  Return "result"  Else  FOREACH module name IN dataContext  LET containerList = Get all containers that are right under moduleName (or "{moduleName}d\*$")  to add to a containers collection "result"  FOREACH container IN containerList  LET get all of its children containers to add to "result"  RETURN "result" | | |

{Ref: N/A}

#### GetModules

GENERIC\_GDD\_CLS\_039\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetModules | | |
| **Arguments** | None | | |
| **Return** | IList<string> | List of module name | |
| Range: | null/valid value |
| **Generated Value** | None | | |
| **Description** | Get all modules' name from dataContext  Algorithm: RETURN list key of moduleInstances | | |

{Ref: N/A}

#### PrepareInstance

GENERIC\_GDD\_CLS\_039\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | PrepareInstance | | |
| **Arguments** | instanceIndex | Instance Index | |
| Range: | None |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Update current instance index  Algorithm: LET currentInstanceIndex = instanceIndex | | |

{Ref: N/A}

#### SaveBswConfiguration

GENERIC\_GDD\_CLS\_039\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveBswConfiguration | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| config | The module configuration | |
| Range: | null/valid Configuration |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Save bsw configuration  Algorithm: IF moduleName is NOT null or empty and config is NOT null THEN  LET bswConfigs = Add config with key is moduleName | | |

{Ref: N/A}

#### SaveCdfFiles

GENERIC\_GDD\_CLS\_039\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveCdfFiles | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| shortName | Short name | |
| Range: | null/valid string |
| fileName | File name | |
| Range: | null/valid string |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Save CDF files  Algorithm: IF both moduleName, shortName and fileName are NOT null or empty THEN  LET cdfFiles = Add fileName with key is moduleName  LET cdfFiles = Add fileName with key is shortName in case multi-instance module | | |

{Ref: N/A}

#### SaveConfiguration

GENERIC\_GDD\_CLS\_039\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveConfiguration | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| key | Key | |
| Range: | null/valid string |
| config | The module configuration | |
| Range: | null/valid Configuration |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Save configuration  Algorithm: If moduleName is NOT null or empty and config is NOT null THEN  LET configurations = Save config with key moduleName | | |

{Ref: N/A}

#### SaveContainer

GENERIC\_GDD\_CLS\_039\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveContainer | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| key | Container | |
| Range: | null/valid Container |
| container | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Add a container of a module 2-levels dictionary dataContext Instance short name 1 ------  Algorithm: IF moduleName, key and container isn't null or empty THEN  IF dataContext contains key THEN  IF value of dataContext with key whose isn't contains sort name of container THEN  LET store container into dataContext  ELSE  LET store container into dataContext  CALL addInstanceIfNeeded WITH moduleName, key | | |

{Ref: N/A}

#### CdfDataDictionary

GENERIC\_GDD\_CLS\_039\_022:

|  |  |
| --- | --- |
| **Method Name** | CdfDataDictionary |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Cdf data dictionary constructor |

{Ref: N/A}

#### addInstanceIfNeeded

GENERIC\_GDD\_CLS\_039\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | addInstanceIfNeeded | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| instanceShortname | Instance short name | |
| Range: | null/valid string |
| **Return** | void | N/A | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Add instance if needed  Algorithm: IF moduleInstances contains moduleName THEN  IF shortNames contains instanceShortname  LET add instanceShortname into shortNames  ELSE  LET value of moduleInstances with Key is moduleName = instanceShortname | | |

{Ref: N/A}

#### getContainerByShortName

GENERIC\_GDD\_CLS\_039\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getContainerByShortName | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| shortName | Short name | |
| Range: | null/valid string |
| **Return** | Container | The container | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Get container by short name  Algorithm: LET containerList = return value of CALL GetCurrentInstanceContainers WITH moduleName  RETURN first container in containerList WHERE it has short name is shortName | | |

{Ref: N/A}

#### getCurrentInstanceKey

GENERIC\_GDD\_CLS\_039\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getCurrentInstanceKey | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | string | Instance key | |
| Range: | null/valid string |
| **Generated Value** | None | | |
| **Description** | Get current instance key  Algorithm: IF moduleInstances contains moduleName  IF keys isn't null THEN  IF size of keys is greater than currentInstanceIndex  LET result = value of keys with Key is currentInstanceIndex  ELSE  LET result = first elemant of keys  RETURN result | | |

{Ref: N/A}

#### isCurrentInstanceAvailable

GENERIC\_GDD\_CLS\_039\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | isCurrentInstanceAvailable | | |
| **Arguments** | None | | |
| **Return** | bool | The status of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Checking instance is current instance  Algorithm: IF currentInstanceIndex >=0 THEN  RETURN 1  ELSE  RETURN 0 | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::CDFData::Items::BswConfig

|  |  |
| --- | --- |
| **Location File Name** | BswConfig.cs |
| **Description** | This class to store basic software configuration information. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | ImplementUuid | Public | This field store Uuid of basic software configuration |
| (2) | ImplementSortName | Public | This field store Sort Name of basic software configuration |
| (3) | ProgrammingLanguage | Public | This field store Programming language of basic software configuration |
| (4) | SwVersion | Public | This field store software Version of basic software configuration |
| (5) | VendorId | Public | This field store Vendor Id of basic software configuration |
| (6) | ArReleaseVersion | Public | This field store Autosar Release Version of basic software configuration |
| (7) | BehaviorRef | Public | This field store Behavior reference of basic software configuration |
| (8) | VendorApiInfix | Public | This field store Vendor api infix of basic software configuration |
| (9) | DescriptionUuid | Public | This field store Description uuid of basic software configuration |
| (10) | DescriptionShortName | Public | This field store Description short name of basic software configuration |
| (11) | ModuleId | Public | This field store Module id of basic software configuration |
| (12) | VendorSpecificModuleDefRef | Public | This field store Describe information Vendor Specific Module DefRef |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| - | - | - | - |

### Fields

#### ImplementUuid

GENERIC\_GDD\_CLS\_040\_001:

|  |  |
| --- | --- |
| **Variable Name** | ImplementUuid |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Uuid of basic software configuration |

#### ImplementSortName

GENERIC\_GDD\_CLS\_040\_002:

|  |  |
| --- | --- |
| **Variable Name** | ImplementSortName |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Sort Name of basic software configuration |

#### ProgrammingLanguage

GENERIC\_GDD\_CLS\_040\_003:

|  |  |
| --- | --- |
| **Variable Name** | ProgrammingLanguage |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Programming language of basic software configuration |

#### SwVersion

GENERIC\_GDD\_CLS\_040\_004:

|  |  |
| --- | --- |
| **Variable Name** | SwVersion |
| **Type** | string |
| **Range** | None |
| **Description** | This field store software Version of basic software configuration |

#### VendorId

GENERIC\_GDD\_CLS\_040\_005:

|  |  |
| --- | --- |
| **Variable Name** | VendorId |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Vendor Id of basic software configuration |

#### ArReleaseVersion

GENERIC\_GDD\_CLS\_040\_006:

|  |  |
| --- | --- |
| **Variable Name** | ArReleaseVersion |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Autosar Release Version of basic software configuration |

#### BehaviorRef

GENERIC\_GDD\_CLS\_040\_007:

|  |  |
| --- | --- |
| **Variable Name** | BehaviorRef |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Behavior reference of basic software configuration |

#### VendorApiInfix

GENERIC\_GDD\_CLS\_040\_008:

|  |  |
| --- | --- |
| **Variable Name** | VendorApiInfix |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Vendor api infix of basic software configuration |

#### DescriptionUuid

GENERIC\_GDD\_CLS\_040\_009:

|  |  |
| --- | --- |
| **Variable Name** | DescriptionUuid |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Description uuid of basic software configuration |

#### DescriptionShortName

GENERIC\_GDD\_CLS\_040\_010:

|  |  |
| --- | --- |
| **Variable Name** | DescriptionShortName |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Description short name of basic software configuration |

#### ModuleId

GENERIC\_GDD\_CLS\_040\_011:

|  |  |
| --- | --- |
| **Variable Name** | ModuleId |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Module id of basic software configuration |

#### VendorSpecificModuleDefRef

GENERIC\_GDD\_CLS\_040\_012:

|  |  |
| --- | --- |
| **Variable Name** | VendorSpecificModuleDefRef |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Describe information Vendor Specific Module DefRef |

### Methods

## Renesas::Generator::MCALConfGen::Data::CDFData::Items::Configuration

|  |  |
| --- | --- |
| **Location File Name** | Configuration.cs |
| **Description** | This class to store basic formation of a CDF tag ECUC-MODULE-CONFIGURATION-VALUES. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Uuid | Public | This field store Uuid of a ECUC-MODULE-CONFIGURATION-VALUES |
| (2) | ShortName | Public | This field store Short name of a ECUC-MODULE-CONFIGURATION-VALUES |
| (3) | DefinitionRef | Public | This field store Definition reference of a ECUC-MODULE-CONFIGURATION-VALUES |
| (4) | ModuleDescriptionRef | Public | This field store Module description reference of a ECUC-MODULE-CONFIGURATION-VALUES |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| - | - | - | - |

### Fields

#### Uuid

GENERIC\_GDD\_CLS\_041\_001:

|  |  |
| --- | --- |
| **Variable Name** | Uuid |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Uuid of a ECUC-MODULE-CONFIGURATION-VALUES |

#### ShortName

GENERIC\_GDD\_CLS\_041\_002:

|  |  |
| --- | --- |
| **Variable Name** | ShortName |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Short name of a ECUC-MODULE-CONFIGURATION-VALUES |

#### DefinitionRef

GENERIC\_GDD\_CLS\_041\_003:

|  |  |
| --- | --- |
| **Variable Name** | DefinitionRef |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Definition reference of a ECUC-MODULE-CONFIGURATION-VALUES |

#### ModuleDescriptionRef

GENERIC\_GDD\_CLS\_041\_004:

|  |  |
| --- | --- |
| **Variable Name** | ModuleDescriptionRef |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Module description reference of a ECUC-MODULE-CONFIGURATION-VALUES |

### Methods

## Renesas::Generator::MCALConfGen::Data::CDFData::Items::Container

|  |  |
| --- | --- |
| **Location File Name** | Container.cs |
| **Description** | This class that stores information of a CDF on the tag ECUC-CONTAINER-VALUE |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Uuid | Public | This field store Uuid of a ECUC-CONTAINER-VALUE |
| (2) | ShortName | Public | This field store Short name of a ECUC-CONTAINER-VALUE |
| (3) | DefinitionRef | Public | This field store Definition reference of a ECUC-CONTAINER-VALUE |
| (4) | ParameterValues | Public | This field store Parameter values of a ECUC-CONTAINER-VALUE |
| (5) | ReferenceValues | Public | This field store Reference values of a ECUC-CONTAINER-VALUE |
| (6) | Parent | Public | This field store Parent of a ECUC-CONTAINER-VALUE |
| (7) | Path | Public | This field store Path of a ECUC-CONTAINER-VALUE |
| (8) | Childs | Public | This field store List of Childs of a ECUC-CONTAINER-VALUE |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Equals | Public | Check if this object and target are equal |
| (2) | GetHashCode | Public | Get hash code of container |
| (3) | GetParameterValue | Public | Get a parameter value from name |
| (4) | GetParameterValues | Public | Get all parameters' values from name |
| (5) | GetReferenceValue | Public | Get a reference value from name |
| (6) | GetReferenceValues | Public | Get all references' values from name |
| (7) | GetDescendant | Public | Get all descendant containers (including this container) |
| (8) | GetChildsByDefName | Public | Get direct children by def name |
| (9) | GetChildByDefName | Public | Get a child by def name |
| (10) | Compare | Public | Compare 2 containers by their def names and short names |

### Fields

#### Uuid

GENERIC\_GDD\_CLS\_042\_001:

|  |  |
| --- | --- |
| **Variable Name** | Uuid |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Uuid of a ECUC-CONTAINER-VALUE |

#### ShortName

GENERIC\_GDD\_CLS\_042\_002:

|  |  |
| --- | --- |
| **Variable Name** | ShortName |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Short name of a ECUC-CONTAINER-VALUE |

#### DefinitionRef

GENERIC\_GDD\_CLS\_042\_003:

|  |  |
| --- | --- |
| **Variable Name** | DefinitionRef |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Definition reference of a ECUC-CONTAINER-VALUE |

#### ParameterValues

GENERIC\_GDD\_CLS\_042\_004:

|  |  |
| --- | --- |
| **Variable Name** | ParameterValues |
| **Type** | IList< ItemValue |
| **Range** | None |
| **Description** | This field store Parameter values of a ECUC-CONTAINER-VALUE |

#### ReferenceValues

GENERIC\_GDD\_CLS\_042\_005:

|  |  |
| --- | --- |
| **Variable Name** | ReferenceValues |
| **Type** | IList< ItemValue |
| **Range** | None |
| **Description** | This field store Reference values of a ECUC-CONTAINER-VALUE |

#### Parent

GENERIC\_GDD\_CLS\_042\_006:

|  |  |
| --- | --- |
| **Variable Name** | Parent |
| **Type** | Container |
| **Range** | None |
| **Description** | This field store Parent of a ECUC-CONTAINER-VALUE |

#### Path

GENERIC\_GDD\_CLS\_042\_007:

|  |  |
| --- | --- |
| **Variable Name** | Path |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Path of a ECUC-CONTAINER-VALUE |

#### Childs

GENERIC\_GDD\_CLS\_042\_008:

|  |  |
| --- | --- |
| **Variable Name** | Childs |
| **Type** | List< Container |
| **Range** | None |
| **Description** | This field store List of Childs of a ECUC-CONTAINER-VALUE |

### Methods

#### Compare

GENERIC\_GDD\_CLS\_042\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Compare | | |
| **Arguments** | con1 | None | |
| Range: | None |
| con2 | None | |
| Range: | None |
| **Return** | int | Status of checking | |
| Range: | 0/1 |
| **Generated Value** | None | | |
| **Description** | Compare 2 containers by their def names and short names  Algorithm: If both are null  Return 0  Else If container1 is null && container2 NOT null  Return -1  Else If container1 NOT null && container2 is null  Return 1  Else  If container1.DefinitionRef and container2.DefinitionRef NOT equal  Return string comparision result of container1.ShortName and container2.ShortName  Else  Return string comparision result of container1.DefinitionRef and container2.DefinitionRef  End | | |

{Ref: N/A}

#### Equals

GENERIC\_GDD\_CLS\_042\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Equals | | |
| **Arguments** | target | Target object | |
| Range: | null/valid object |
| **Return** | bool | The status of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check if this object and target are equal  Algorithm: LET ret = false  LET targetContainer = target as instance of Container  IF targetContainer is null  LET ret = false  ELSE  LET ret = Uuid and ShortName of current object equals to values of targetContainer  RETURN ret | | |

{Ref: N/A}

#### GetChildByDefName

GENERIC\_GDD\_CLS\_042\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetChildByDefName | | |
| **Arguments** | defName | Def name | |
| Range: | null/valid string |
| sortOption | Sort option | |
| Range: | 0/valid integer |
| **Return** | IList<Container> | List of containers | |
| Range: | empty/valid value |
| **Generated Value** | None | | |
| **Description** | Get a child by def name  Algorithm: Get a list of child by def name by calling method GetChildsByDefName  If not found  Return null  Else  Return first child | | |

{Ref: N/A}

#### GetChildsByDefName

GENERIC\_GDD\_CLS\_042\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetChildsByDefName | | |
| **Arguments** | defName | Def name | |
| Range: | null/valid string |
| sortOption | Sort option | |
| Range: | 0/valid integer |
| **Return** | IList<Container> | List of containers | |
| Range: | empty/valid value |
| **Generated Value** | None | | |
| **Description** | Get direct children by def name  Algorithm: Scan all direct children of this container  Collect a list of children where their DefinitionRef end with "/" + defName  End   Short list if sortOption = 1   Return list | | |

{Ref: N/A}

#### GetDescendant

GENERIC\_GDD\_CLS\_042\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetDescendant | | |
| **Arguments** | None | | |
| **Return** | IList<Container> | List of containers | |
| Range: | empty/valid value |
| **Generated Value** | None | | |
| **Description** | Get all descendant containers (including this container)  Algorithm: LET ret = init list of Container  LET containerStack = init stack of Container  CALL containerStack.Push(this) to insert current object at top of containerStack  WHILE Count of containerStack is larger than 0  LET con = CALL containerStack.Pop() to get top Container of containerStack  CALL ret.Add WITH con to add con to ret | | |

{Ref: N/A}

#### GetHashCode

GENERIC\_GDD\_CLS\_042\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetHashCode | | |
| **Arguments** | None | | |
| **Return** | int | hash code | |
| Range: | 0/valid int |
| **Generated Value** | None | | |
| **Description** | Get hash code of container  Algorithm: LET hashCode = Constant.INT\_ZERO  IF Uuid is null or empty  hashCode = CALL Uuid.GetHashCode() to get hash code of Uuid  ELSE  hashCode = CALL base.GetHashCode() to get default hash code  RETURN hashCode | | |

{Ref: N/A}

#### GetParameterValue

GENERIC\_GDD\_CLS\_042\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetParameterValue | | |
| **Arguments** | paramName | Param name | |
| Range: | null/valid string |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Get a parameter value from name  Algorithm: LET item = return value of scanning ParameterValues of this container where its definition reference equals to paramName   RETURN item | | |

{Ref: N/A}

#### GetParameterValues

GENERIC\_GDD\_CLS\_042\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetParameterValues | | |
| **Arguments** | paramName | Param name | |
| Range: | null/valid string |
| **Return** | IList<ItemValue> | List of parameter values | |
| Range: | empty/valid value |
| **Generated Value** | None | | |
| **Description** | Get all parameters' values from name  Algorithm: LET paramList = list of return value of scanning ParameterValues of this container where its definition reference equals to paramName   RETURN paramList | | |

{Ref: N/A}

#### GetReferenceValue

GENERIC\_GDD\_CLS\_042\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetReferenceValue | | |
| **Arguments** | paramName | Param name | |
| Range: | null/valid string |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Get a reference value from name  Algorithm: LET item = return value of scanning ReferenceValues of this container where its definition reference equals to paramName   RETURN item | | |

{Ref: N/A}

#### GetReferenceValues

GENERIC\_GDD\_CLS\_042\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetReferenceValues | | |
| **Arguments** | paramName | Param name | |
| Range: | null/valid string |
| **Return** | IList<ItemValue> | List of reference values | |
| Range: | empty/valid value |
| **Generated Value** | None | | |
| **Description** | Get all references' values from name  Algorithm: LET paramList = list of return value of scanning ReferenceValues of this container where its definition reference equals to paramName   RETURN paramList | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::CDFData::Items::ItemValue

|  |  |
| --- | --- |
| **Location File Name** | ItemValue.cs |
| **Description** | This class stores information of Parameter value specified by CDF tag ECUC-TEXTUAL-PARAM-VALUE or ECUC-NUMERICAL-PARAM-VALUE Reference value specified by CDF tag ECUC-REFERENCE-VALUE |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | definitionRef | Private | This field store Definition reference of a ECUC-REFERENCE-VALUE |
| (2) | value | Private | This field store Value of reference of a ECUC-REFERENCE-VALUE |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ItemValue | Public | Item value constructor |
| (2) | DefinitionRef | Public | This function return Definition reference |
| (3) | Value< T > | Public | This function return Value of reference |
| (4) | InternalValue | Private | This function return Value of reference |

### Fields

#### definitionRef

GENERIC\_GDD\_CLS\_043\_001:

|  |  |
| --- | --- |
| **Variable Name** | definitionRef |
| **Type** | string |
| **Range** | None |
| **Description** | This field store Definition reference of a ECUC-REFERENCE-VALUE |

#### value

GENERIC\_GDD\_CLS\_043\_002:

|  |  |
| --- | --- |
| **Variable Name** | value |
| **Type** | object |
| **Range** | None |
| **Description** | This field store Value of reference of a ECUC-REFERENCE-VALUE |

### Methods

#### DefinitionRef

GENERIC\_GDD\_CLS\_043\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | DefinitionRef | | |
| **Arguments** | None | | |
| **Return** | string | Definition reference | |
| Range: | empty/valid vaule |
| **Generated Value** | None | | |
| **Description** | This function return Definition reference  Algorithm: RETURN definitionRef | | |

{Ref: N/A}

#### ItemValue

GENERIC\_GDD\_CLS\_043\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ItemValue | | |
| **Arguments** | definitionRef | Definition reference | |
| Range: | null/valid string |
| value | Value | |
| Range: | null/valid object |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Item value constructor | | |

{Ref: N/A}

#### Value< T >

GENERIC\_GDD\_CLS\_043\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Value< T > | | |
| **Arguments** | None | | |
| **Return** | Type of value | value of reference | |
| Range: | null/valid vaule |
| **Generated Value** | None | | |
| **Description** | This function return Value of reference  Algorithm: RETURN value | | |

{Ref: N/A}

#### InternalValue

GENERIC\_GDD\_CLS\_043\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | InternalValue | | |
| **Arguments** | None | | |
| **Return** | object | value of reference | |
| Range: | null/valid vaule |
| **Generated Value** | None | | |
| **Description** | This function return Value of reference  Algorithm: RETURN value | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::CDFData::RepositoryLinQ

|  |  |
| --- | --- |
| **Location File Name** | RepositoryLinQ.cs |
| **Description** | This |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | cdfData | Private | Cdf data |
| (3) | logger | Private | Logger |
| (4) | register | Private | Register |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetCdfFileName | Public | Get CDF file name |
| (2) | GetContainerByDefName | Public | Get a container by def name |
| (3) | GetContainersByDefName | Public | Get all containers by def name |
| (4) | GetAllInstancesContainersByDefName | Public | Get all containers based on definition name |
| (5) | GetContainerByShortNamePath | Public | Get a container by short name path |
| (6) | GetContainers | Public | Get all containers by module name |
| (7) | GetMacroAddressValue | Public | Get macro address value |
| (8) | GetAddressByMacroDefinition | Public | Get address by macro definition |
| (9) | GetMacroLabelValue | Public | Get macro label value |
| (10) | GetParameterValue | Public | Get parameter value by def name |
| (11) | GetParameterValues | Public | Get parameter values by def name |
| (12) | GetReferenceValue | Public | Get reference value |
| (13) | GetReferenceValues | Public | To get reference values |
| (14) | GetChilds | Public | Get all children container of a container which is identified by module name and short name |
| (15) | GetContainer | Public | Get container by module name and short name |
| (16) | GetVersionInformation | Public | Get version detail information by name |
| (17) | IsExistedPath | Public | Check if path existed |
| (18) | GetStartOfDbToc | Public | Get start of db toc |
| (19) | IsExistModuleName | Public | Check if module name existed |
| (20) | GetConfiguration | Public | Get configuration by module name |
| (21) | GetConfigurations | Public | Get configurations |
| (22) | IsRegisterExist | Public | Check if register existed |
| (23) | PrepareInstance | Public | Update current instance |
| (24) | ComputeOnOffParams | Public | Compute values of defined parameters that is bool data type. |
| (25) | ComputeDemEventParams | Public | Process values of dem event paramaters. |
| (26) | GetVersionInformation | Public | To compute value of all param Version Information |
| (27) | RepositoryLinQ | Protected | Repository LinQ constructor |
| (28) | RepositoryLinQ | Protected | Repository LinQ constructor |
| (29) | getContainersByDefName | Private | Get containers based DefName |
| (30) | initVersionInformation | Private | Init values autosar version requirement and software version |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_044\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly RepositoryLinQ |
| **Range** | None |
| **Description** | None |

#### cdfData

GENERIC\_GDD\_CLS\_044\_002:

|  |  |
| --- | --- |
| **Variable Name** | cdfData |
| **Type** | ICdfData |
| **Range** | None |
| **Description** | Cdf data |

#### logger

GENERIC\_GDD\_CLS\_044\_003:

|  |  |
| --- | --- |
| **Variable Name** | logger |
| **Type** | ILogger |
| **Range** | None |
| **Description** | Logger |

#### register

GENERIC\_GDD\_CLS\_044\_004:

|  |  |
| --- | --- |
| **Variable Name** | register |
| **Type** | IRegisterProcessing |
| **Range** | None |
| **Description** | Register |

### Methods

#### ComputeDemEventParams

GENERIC\_GDD\_CLS\_044\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ComputeDemEventParams | | |
| **Arguments** | demEventParamsGroup | A collection contains parameter names | |
| Range: | null/valid Dictionary<string, string> |
| **Return** | BaseIntermediateItem | The intermediate data | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Process values of dem event paramaters.  Algorithm: If parameter is configured, concatenate "DemConf\_DemEventParameter\_" and last value of paramater | | |

{Ref: N/A}

#### ComputeOnOffParams

GENERIC\_GDD\_CLS\_044\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ComputeOnOffParams | | |
| **Arguments** | onOffParamsGroup | The collection of on/off parameters | |
| Range: | null/valid Dictionary<string, string> |
| **Return** | BaseIntermediateItem | The intermediate data | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Compute values of defined parameters that is bool data type.  Algorithm: Get value of parameter in module  Convert to STD\_ON if value is true, else is STD\_OFF | | |

{Ref: N/A}

#### GetAddressByMacroDefinition

GENERIC\_GDD\_CLS\_044\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAddressByMacroDefinition | | |
| **Arguments** | renesasMacroName | None | |
| Range: | None |
| **Return** | string | The macro address value | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get address by macro definition  Algorithm: Get macro value from macro definition by calling method GetMacroLabelValue  If macro value is NOT null or empty  Get macro address from macro value by calling method GetMacroAddressValue  Return macro address  Else  Return empty | | |

{Ref: N/A}

#### GetAllInstancesContainersByDefName

GENERIC\_GDD\_CLS\_044\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAllInstancesContainersByDefName | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| defName | Def name | |
| Range: | null/valid string |
| sortOption | Sort option | |
| Range: | 0/valid integer |
| **Return** | Dictionary<string, IList<Container>> | The list of containers | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get all containers based on definition name  Algorithm: Get all containers of instances  With each instance, get all containers based on definition name | | |

{Ref: N/A}

#### GetCdfFileName

GENERIC\_GDD\_CLS\_044\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetCdfFileName | | |
| **Arguments** | shortName | Short name | |
| Range: | null/valid string |
| **Return** | string | The cdf file name | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get CDF file name  Algorithm: Get CDF file name of shortName from cdfData | | |

{Ref: N/A}

#### GetChilds

GENERIC\_GDD\_CLS\_044\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetChilds | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| shortName | Short name | |
| Range: | null/valid string |
| **Return** | IList<Container> | The list of containers | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get all children container of a container which is identified by module name and short name  Algorithm: Get root container which is identified by module name and short name  If container != null  Recursively find children of the container  Return found children  Else  Return empty list | | |

{Ref: N/A}

#### GetConfiguration

GENERIC\_GDD\_CLS\_044\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetConfiguration | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | Configuration | The module configuration | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get configuration by module name  Algorithm: Get configuration in cdfData for moduleName | | |

{Ref: N/A}

#### GetConfigurations

GENERIC\_GDD\_CLS\_044\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetConfigurations | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | IList<Configuration> | The list of module configurations | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get configurations  Algorithm: Get all instances based on module name  Get configuration for each instance. | | |

{Ref: N/A}

#### GetContainer

GENERIC\_GDD\_CLS\_044\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainer | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| shortName | Short name | |
| Range: | null/valid string |
| **Return** | Container | The container | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get container by module name and short name  Algorithm: Get a list of containers in cdfData by moduleName  For each container  If container's ShortName = shortName  Return container  End  End   Return null if NOT found a container | | |

{Ref: N/A}

#### GetContainerByDefName

GENERIC\_GDD\_CLS\_044\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainerByDefName | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| defName | Def name | |
| Range: | null/valid string |
| sortOption | Sort option | |
| Range: | 0/valid integer |
| **Return** | Container | The container | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get a container by def name  Algorithm: CALL GetContainersByDefName(moduleName, defName, sortOption).FirstOrDefault() to get list of containers | | |

{Ref: N/A}

#### GetContainerByShortNamePath

GENERIC\_GDD\_CLS\_044\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainerByShortNamePath | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| path | Path | |
| Range: | null/valid string |
| **Return** | Container | The container | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get a container by short name path  Algorithm: Get all containers in cdfData by moduleName  If found a container where container.Path = path  RETURN first found  Else  RETURN null | | |

{Ref: N/A}

#### GetContainers

GENERIC\_GDD\_CLS\_044\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainers | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | IList<Container> | The list of containers | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get all containers by module name  Algorithm: Get all containers in cdfData by moduleName | | |

{Ref: N/A}

#### GetContainersByDefName

GENERIC\_GDD\_CLS\_044\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainersByDefName | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| defName | Def name | |
| Range: | null/valid string |
| sortOption | Sort option | |
| Range: | 0/valid integer |
| **Return** | IList<Container> | The list of containers | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get all containers by def name  Algorithm: LET containerList = (List<Container>)cdfData.GetCurrentInstanceContainers WITH moduleName  RETURN getContainersByDefName(containerList, defName, sortOption) | | |

{Ref: N/A}

#### GetMacroAddressValue

GENERIC\_GDD\_CLS\_044\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetMacroAddressValue | | |
| **Arguments** | macroValue | Macro value | |
| Range: | null/valid string |
| **Return** | string | The macro address values | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get macro address value  Algorithm: CALL GetMacroAddressValue from register to get address value | | |

{Ref: N/A}

#### GetMacroLabelValue

GENERIC\_GDD\_CLS\_044\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetMacroLabelValue | | |
| **Arguments** | renesasMacroName | None | |
| Range: | None |
| **Return** | string | The macro label value | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get macro label value  Algorithm: Call GetMacroLabelValue from register to get label value | | |

{Ref: N/A}

#### GetParameterValue

GENERIC\_GDD\_CLS\_044\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetParameterValue | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| defName | Def name | |
| Range: | null/valid string |
| **Return** | ItemValue | The parameter value | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get parameter value by def name  Algorithm: Get all containers in cdfData by moduleName  If found any parameter of container has defName  Return first found parameter  Else  Return null | | |

{Ref: N/A}

#### GetParameterValues

GENERIC\_GDD\_CLS\_044\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetParameterValues | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| defName | Def name | |
| Range: | null/valid string |
| **Return** | IList<ItemValue> | The list of parameter values | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get parameter values by def name  Algorithm: Get all containers in cdfData by moduleName  For each container  Get parameter value by defName and add to a collection "result"  End   Return collection "result" | | |

{Ref: N/A}

#### GetReferenceValue

GENERIC\_GDD\_CLS\_044\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetReferenceValue | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| defName | Def name | |
| Range: | null/valid string |
| **Return** | ItemValue | The reference value | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get reference value  Algorithm: Get all containers in cdfData by moduleName  If found any reference of container has defName  Return first found reference  Else  Return null | | |

{Ref: N/A}

#### GetReferenceValues

GENERIC\_GDD\_CLS\_044\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetReferenceValues | | |
| **Arguments** | paramName | param Name | |
| Range: | null/valid string |
| **Return** | List<ItemValue> | The list of reference values | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | To get reference values | | |

{Ref: N/A}

#### GetStartOfDbToc

GENERIC\_GDD\_CLS\_044\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetStartOfDbToc | | |
| **Arguments** | None | | |
| **Return** | string | The start of database | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get start of db toc  Algorithm: 1. Get swMajorVersion in basic software configuration "SW-VERSION"  Use 0 for swMajorVersion if it's NOT a integer  2. Get swMinorVersion in basic software configuration "SW-VERSION"  Use 0 for swMinorVersion if it's NOT a integer  3. Get moduleId in basic software configuration "MODULE-ID"  Use 0 for moduleId if it's NOT a integer  4. Get vendorId in basic software configuration "VENDOR-ID"  Use 0 for vendorId if it's NOT a integer   Calculate start of db toc =  shift\_left(vendorId, 22) or  shift\_left(moduleId, 14) or  shift\_left(swMajorVersion, 8) or  shift\_left(swMinorVersion, 3) | | |

{Ref: N/A}

#### GetVersionInformation

GENERIC\_GDD\_CLS\_044\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetVersionInformation | | |
| **Arguments** | name | Name | |
| Range: | null/valid string |
| moduleName | Module Name | |
| Range: | null/valid string |
| **Return** | string | The version information | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get version detail information by name  Algorithm: Get basic software config in cdfData by moduleName  If found a field in basic software config based that match with name  Return value of found field  Else  Return null | | |

{Ref: N/A}

#### GetVersionInformation

GENERIC\_GDD\_CLS\_044\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetVersionInformation | | |
| **Arguments** | None | | |
| **Return** | BaseIntermediateItem | The intermediate data | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | To compute value of all param Version Information  Algorithm: From Repository, call GetVersionInformation method for get  "AR-RELEASE-VERSION", "SW-VERSION", "VENDOR-ID", "MODULE-ID"  Check if Ar version & Sw version is not null or not empty  Slipt information to {x,y,z}.  Check if Ar version & Sw version have 3 element {x,y,z}  Add Cfg Version Information to child of return value  Add Cbk Version Information to child of return value  Add Pbcfg Version Information to child of return value  Add LinkTime Version Information to child of return value  else do nothing  else do nothing | | |

{Ref: N/A}

#### IsExistModuleName

GENERIC\_GDD\_CLS\_044\_027:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsExistModuleName | | |
| **Arguments** | moduleName | Module name | |
| Range: | null/valid string |
| **Return** | bool | The status of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check if module name existed  Algorithm: Call GetContainers method to get all containers by moduleName  If number of containers is larger than 0  Return true  Else  Return false | | |

{Ref: N/A}

#### IsExistedPath

GENERIC\_GDD\_CLS\_044\_028:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsExistedPath | | |
| **Arguments** | path | Path | |
| Range: | null/valid string |
| moduleName | Module name | |
| Range: | null/valid string |
| **Return** | bool | The status of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check if path existed  Algorithm: Get all containers in cdfData by moduleName  For each container  If found a container that container.Path = path  Return true  Else  Return false  End | | |

{Ref: N/A}

#### IsRegisterExist

GENERIC\_GDD\_CLS\_044\_029:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsRegisterExist | | |
| **Arguments** | renesasMacroDefinition | None | |
| Range: | None |
| **Return** | bool | The status of checking | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check if register existed  Algorithm: Call IsRegisterExist from register to check if register of macro definition existed | | |

{Ref: N/A}

#### PrepareInstance

GENERIC\_GDD\_CLS\_044\_030:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | PrepareInstance | | |
| **Arguments** | instanceIndex | Instance Index | |
| Range: | N/A |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Update current instance  Algorithm: Update current instance based index instance. | | |

{Ref: N/A}

#### RepositoryLinQ

GENERIC\_GDD\_CLS\_044\_031:

|  |  |
| --- | --- |
| **Method Name** | RepositoryLinQ |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Repository LinQ constructor |

{Ref: N/A}

#### RepositoryLinQ

GENERIC\_GDD\_CLS\_044\_032:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | RepositoryLinQ | | |
| **Arguments** | cdfData | Cdf data | |
| Range: | null/valid |
| logger | Logger | |
| Range: | null/valid ILogger |
| register | Register | |
| Range: | null/valid |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Repository LinQ constructor | | |

{Ref: N/A}

#### getContainersByDefName

GENERIC\_GDD\_CLS\_044\_033:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getContainersByDefName | | |
| **Arguments** | containers | List of container | |
| Range: | null/valid IList<Container> |
| defName | definition ref | |
| Range: | nul/valid string |
| sortOption | Sort type | |
| Range: | N/A |
| **Return** | IList<Container> | The list of containers | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get containers based DefName  Algorithm: Get all containers from input  Filter containers based on DefName  Sort containers that is filtered | | |

{Ref: N/A}

#### initVersionInformation

GENERIC\_GDD\_CLS\_044\_034:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | initVersionInformation | | |
| **Arguments** | autosarReqInfo | Autosar version info | |
| Range: | null/valid string[] |
| softWareInfo | software version info | |
| Range: | null/valid string[] |
| vendorId | The vendor Id | |
| Range: | null/valid string |
| moduleId | The module Id | |
| Range: | null/valid string |
| **Return** | Dictionary<string, string> | The collection of version info | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Init values autosar version requirement and software version  Algorithm: Compute all autosar version and software version that is accessed from bsw module. | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateData

|  |  |
| --- | --- |
| **Location File Name** | IntermediateData.cs |
| **Description** | This |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | None |
| (2) | root | Private | Root |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | IntermediateData | Public | Intermediate data constructor |
| (2) | GetRoot | Public | Get root |
| (3) | GetStringDataByPath | Public | Get string data by path |
| (4) | GetItemByPath | Public | Get item by path |
| (5) | Store | Public | Store BaseIntermediateItem into root |

### Fields

#### Instance

GENERIC\_GDD\_CLS\_045\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly IntermediateData |
| **Range** | None |
| **Description** | None |

#### root

GENERIC\_GDD\_CLS\_045\_002:

|  |  |
| --- | --- |
| **Variable Name** | root |
| **Type** | BaseIntermediateItem |
| **Range** | None |
| **Description** | Root |

### Methods

#### GetItemByPath

GENERIC\_GDD\_CLS\_045\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetItemByPath | | |
| **Arguments** | path | Path | |
| Range: | null/valid string |
| **Return** | BaseIntermediateItem | Intermediate data | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get item by path  Algorithm: Get BaseIntermediateItem by path from root BaseIntermediateItem | | |

{Ref: N/A}

#### GetRoot

GENERIC\_GDD\_CLS\_045\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetRoot | | |
| **Arguments** | None | | |
| **Return** | BaseIntermediateItem | Intermediate data | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get root  Algorithm: Return root BaseIntermediateItem | | |

{Ref: N/A}

#### GetStringDataByPath

GENERIC\_GDD\_CLS\_045\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetStringDataByPath | | |
| **Arguments** | path | Path | |
| Range: | null/valid string |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get string data by path  Algorithm: Get string data by path from root BaseIntermediateItem | | |

{Ref: N/A}

#### IntermediateData

GENERIC\_GDD\_CLS\_045\_006:

|  |  |
| --- | --- |
| **Method Name** | IntermediateData |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Intermediate data constructor |

{Ref: N/A}

#### Store

GENERIC\_GDD\_CLS\_045\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Store | | |
| **Arguments** | path | Path | |
| Range: | null/valid string |
| target | Target | |
| Range: | null/valid BaseIntermediateItem |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Store BaseIntermediateItem into root  Algorithm: Store BaseIntermediateItem with path into root | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateItemAttribute

|  |  |
| --- | --- |
| **Location File Name** | IntermediateItemAttribute.cs |
| **Description** | This |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | DBPath | Public | DB path |
| (2) | Line | Public | Line number |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | IntermediateItemAttribute | Public | Intermediate item attribute constructor |

### Fields

#### DBPath

GENERIC\_GDD\_CLS\_046\_001:

|  |  |
| --- | --- |
| **Variable Name** | DBPath |
| **Type** | string |
| **Range** | None |
| **Description** | DB path |

#### Line

GENERIC\_GDD\_CLS\_046\_002:

|  |  |
| --- | --- |
| **Variable Name** | Line |
| **Type** | int |
| **Range** | None |
| **Description** | Line number |

### Methods

#### IntermediateItemAttribute

GENERIC\_GDD\_CLS\_046\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IntermediateItemAttribute | | |
| **Arguments** | line | line | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Intermediate item attribute constructor | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::IntermediateData::Items::BaseIntermediateItem

|  |  |
| --- | --- |
| **Location File Name** | BaseIntermediateItem.cs |
| **Description** | This |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Name | Public | Name |
| (2) | Value | Public | Value |
| (3) | Childs | Public | Childs |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | BaseIntermediateItem | Public | Base intermediate item constructor |
| (2) | BaseIntermediateItem | Public | Base intermediate item constructor |
| (3) | AddChild | Public | Add child |
| (4) | ToString | Public | To string method |
| (5) | GetStringDataByPath | Public | Get string data by path |
| (6) | GetItemByPath | Public | Get item by path |
| (7) | Store | Public | Store |

### Fields

#### Name

GENERIC\_GDD\_CLS\_047\_001:

|  |  |
| --- | --- |
| **Variable Name** | Name |
| **Type** | string |
| **Range** | None |
| **Description** | Name |

#### Value

GENERIC\_GDD\_CLS\_047\_002:

|  |  |
| --- | --- |
| **Variable Name** | Value |
| **Type** | string |
| **Range** | None |
| **Description** | Value |

#### Childs

GENERIC\_GDD\_CLS\_047\_003:

|  |  |
| --- | --- |
| **Variable Name** | Childs |
| **Type** | List< BaseIntermediateItem |
| **Range** | None |
| **Description** | Childs |

### Methods

#### AddChild

GENERIC\_GDD\_CLS\_047\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AddChild | | |
| **Arguments** | child | Childs | |
| Range: | null/valid List<BaseIntermediateItem> |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Add child  Algorithm: Add child to a children list | | |

{Ref: N/A}

#### BaseIntermediateItem

GENERIC\_GDD\_CLS\_047\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseIntermediateItem | | |
| **Arguments** | name | Name | |
| Range: | None |
| value | Value | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base intermediate item constructor | | |

{Ref: N/A}

#### BaseIntermediateItem

GENERIC\_GDD\_CLS\_047\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseIntermediateItem | | |
| **Arguments** | name | Name | |
| Range: | null/valid string |
| value | Value | |
| Range: | null/valid string |
| childs | Childs | |
| Range: | null/valid List<BaseIntermediateItem> |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base intermediate item constructor | | |

{Ref: N/A}

#### GetItemByPath

GENERIC\_GDD\_CLS\_047\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetItemByPath | | |
| **Arguments** | path | Path | |
| Range: | null/valid string |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Get item by path  Algorithm: Path has format of multiple segments "/Segment/Segment/Segment"  Tracing path by comparing BaseIntermediateItem.Name with each path segment  If found a BaseIntermediateItem  Return the item  Else  Return null | | |

{Ref: N/A}

#### GetStringDataByPath

GENERIC\_GDD\_CLS\_047\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetStringDataByPath | | |
| **Arguments** | path | Path | |
| Range: | null/valid string |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get string data by path  Algorithm: Get BaseIntermediateItem by path by calling method GetItemByPath  If BaseIntermediateItem is NOT null  Return BaseIntermediateItem.Value  Else  Return empty | | |

{Ref: N/A}

#### Store

GENERIC\_GDD\_CLS\_047\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Store | | |
| **Arguments** | path | Path | |
| Range: | null/valid string |
| target | Target | |
| Range: | null/valid |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Store  Algorithm: Path has format of multiple segments "/Segment/Segment/Segment"  From current BaseIntermediateItem (this), tracing the path to find a farthest BaseIntermediateItem.  Farthest BaseIntermediateItem can be  - A last segment or  - An intermediate segment where there are NO corresponding BaseIntermediateItems for remaining segments   Add target BaseIntermediateItem into the farthest BaseIntermediateItem. | | |

{Ref: N/A}

#### ToString

GENERIC\_GDD\_CLS\_047\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ToString | | |
| **Arguments** | None | | |
| **Return** | string | The output string | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | To string method  Algorithm: Return a string with following information  - Current node: Name, Value and Childs.Count  - All child nodes: Name, Value and inner Childs.Count | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Data::RuntimeConfiguration::RuntimeConfiguration

|  |  |
| --- | --- |
| **Location File Name** | RuntimeConfiguration.cs |
| **Description** | This class implement |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | DefaultVendor | Public | None |
| (2) | Instance | Public | None |
| (3) | TranslationFilePath | Public | Translation file path |
| (4) | CDFFilePaths | Public | CDF file paths |
| (5) | Module | Public | Module name |
| (6) | ConfigFilePath | Public | Config file path |
| (7) | PrintHelp | Public | Print help |
| (8) | SrcOutputDirectory | Public | Src output directory |
| (9) | IncOutputDirectory | Public | Inc output directory |
| (10) | WriteLog | Public | Write log |
| (11) | LogFilePath | Public | Log file path |
| (12) | DryRun | Public | Dry run |
| (13) | FolderOutput | Public | Folder output |
| (14) | VendorFilter | Public | Vendor filter |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetSingleCmdOptions | Public | Get single command options |
| (2) | GetAllCmdOptions | Public | Get all command options |
| (3) | OverrideCurrentConfigsBy | Public | This method override options from command line |
| (4) | SetDefaultOptionsValue | Public | Set default options value |
| (5) | RuntimeConfiguration | Private | Runtime configuration constructor |
| (6) | overrideOptions | Private | This method override options from command line |
| (7) | overrideFiles | Private | This method override CDF file option from command line |
| (8) | getCmdOptions | Private | Get commandline options which include both sort/completed format based on specific condition |
| (9) | concatPaths | Private | Concat paths |

### Fields

#### DefaultVendor

GENERIC\_GDD\_CLS\_048\_001:

|  |  |
| --- | --- |
| **Variable Name** | DefaultVendor |
| **Type** | readonly string |
| **Range** | None |
| **Description** | None |

#### Instance

GENERIC\_GDD\_CLS\_048\_002:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly RuntimeConfiguration |
| **Range** | None |
| **Description** | None |

#### TranslationFilePath

GENERIC\_GDD\_CLS\_048\_003:

|  |  |
| --- | --- |
| **Variable Name** | TranslationFilePath |
| **Type** | string |
| **Range** | None |
| **Description** | Translation file path |

#### CDFFilePaths

GENERIC\_GDD\_CLS\_048\_004:

|  |  |
| --- | --- |
| **Variable Name** | CDFFilePaths |
| **Type** | List< string > |
| **Range** | None |
| **Description** | CDF file paths |

#### Module

GENERIC\_GDD\_CLS\_048\_005:

|  |  |
| --- | --- |
| **Variable Name** | Module |
| **Type** | string |
| **Range** | None |
| **Description** | Module name |

#### ConfigFilePath

GENERIC\_GDD\_CLS\_048\_006:

|  |  |
| --- | --- |
| **Variable Name** | ConfigFilePath |
| **Type** | string |
| **Range** | None |
| **Description** | Config file path |

#### PrintHelp

GENERIC\_GDD\_CLS\_048\_007:

|  |  |
| --- | --- |
| **Variable Name** | PrintHelp |
| **Type** | bool |
| **Range** | None |
| **Description** | Print help |

#### SrcOutputDirectory

GENERIC\_GDD\_CLS\_048\_008:

|  |  |
| --- | --- |
| **Variable Name** | SrcOutputDirectory |
| **Type** | string |
| **Range** | None |
| **Description** | Src output directory |

#### IncOutputDirectory

GENERIC\_GDD\_CLS\_048\_009:

|  |  |
| --- | --- |
| **Variable Name** | IncOutputDirectory |
| **Type** | string |
| **Range** | None |
| **Description** | Inc output directory |

#### WriteLog

GENERIC\_GDD\_CLS\_048\_010:

|  |  |
| --- | --- |
| **Variable Name** | WriteLog |
| **Type** | bool |
| **Range** | None |
| **Description** | Write log |

#### LogFilePath

GENERIC\_GDD\_CLS\_048\_011:

|  |  |
| --- | --- |
| **Variable Name** | LogFilePath |
| **Type** | string |
| **Range** | None |
| **Description** | Log file path |

#### DryRun

GENERIC\_GDD\_CLS\_048\_012:

|  |  |
| --- | --- |
| **Variable Name** | DryRun |
| **Type** | bool |
| **Range** | None |
| **Description** | Dry run |

#### FolderOutput

GENERIC\_GDD\_CLS\_048\_013:

|  |  |
| --- | --- |
| **Variable Name** | FolderOutput |
| **Type** | string |
| **Range** | None |
| **Description** | Folder output |

#### VendorFilter

GENERIC\_GDD\_CLS\_048\_014:

|  |  |
| --- | --- |
| **Variable Name** | VendorFilter |
| **Type** | string |
| **Range** | None |
| **Description** | Vendor filter |

### Methods

#### GetAllCmdOptions

GENERIC\_GDD\_CLS\_048\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAllCmdOptions | | |
| **Arguments** | None | | |
| **Return** | string[] | Array of command option | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get all command options  Algorithm: Get all valid options of command line | | |

{Ref: N/A}

#### GetSingleCmdOptions

GENERIC\_GDD\_CLS\_048\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetSingleCmdOptions | | |
| **Arguments** | None | | |
| **Return** | string[] | Array of command option | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get single command options  Algorithm: Get all options that has type bool | | |

{Ref: N/A}

#### OverrideCurrentConfigsBy

GENERIC\_GDD\_CLS\_048\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | OverrideCurrentConfigsBy | | |
| **Arguments** | options | Collection of command line options | |
| Range: | null/valid Dictionary<string, string> |
| filePaths | File paths | |
| Range: | null/valid List<string> |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | This method override options from command line  Algorithm: CALL overrideOptions to override all options  CALL overrideFiles to override all files | | |

{Ref: N/A}

#### SetDefaultOptionsValue

GENERIC\_GDD\_CLS\_048\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SetDefaultOptionsValue | | |
| **Arguments** | basicConfig | Basic config | |
| Range: | null/valid IBasicConfiguration |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Set default options value  Algorithm: IF options is not configed, it is assigned by default value.  IF IncOutputDirectory is NOT specified  IncOutputDirectory = FolderOutput + "\include\" // Implementation: GENERIC\_GDD\_CLS\_048\_009  ELSE  IncOutputDirectory = IncOutputDirectory + "\include\" // Implementation: GENERIC\_GDD\_CLS\_048\_009   IF SrcOutputDirectory is NOT specified  SrcOutputDirectory = FolderOutput + "\include\" // Implementation: GENERIC\_GDD\_CLS\_048\_008  ELSE  SrcOutputDirectory = SrcOutputDirectory + "\src\" // Implementation: GENERIC\_GDD\_CLS\_048\_008   IF TranslationFilePath is NOT specified  TranslationFilePath = ExeDirectory + ModuleName + "\_" + MicroFamilyName + ".trxml" // Implementation: GENERIC\_GDD\_CLS\_048\_003  End | | |

{Ref: N/A}

#### RuntimeConfiguration

GENERIC\_GDD\_CLS\_048\_019:

|  |  |
| --- | --- |
| **Method Name** | RuntimeConfiguration |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Runtime configuration constructor |

{Ref: N/A}

#### concatPaths

GENERIC\_GDD\_CLS\_048\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | concatPaths | | |
| **Arguments** | paths | Paths | |
| Range: | null/valid string[] |
| **Return** | string | The file path | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Concat paths  Algorithm: concatenate paths to create a valid path with correct separator | | |

{Ref: N/A}

#### getCmdOptions

GENERIC\_GDD\_CLS\_048\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getCmdOptions | | |
| **Arguments** | condition | Specific condition to filter result | |
| Range: | null/valid Predicate<PropertyInfo> |
| **Return** | List<string> | List of commandline options | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | Get commandline options which include both sort/completed format based on specific condition  Algorithm: Scan all fields decorated by OptionAttribute  Add both sort/completed format of option to collection "result"  End   Return collection "result" | | |

{Ref: N/A}

#### overrideFiles

GENERIC\_GDD\_CLS\_048\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | overrideFiles | | |
| **Arguments** | filePaths | File paths | |
| Range: | null/valid List<string> |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | This method override CDF file option from command line  Algorithm: IF filePaths is not null AND count of filePaths is not 0  LET TranslationFilePath = String.Empty // Implementation: GENERIC\_GDD\_CLS\_048\_003  LET CDFFilePaths = init new List<string>() // Implementation: GENERIC\_GDD\_CLS\_048\_004  FOREACH filePath IN filePaths  IF filePath ends with '.trxml'  LET TranslationFilePath = filePath // Implementation: GENERIC\_GDD\_CLS\_048\_003  ELSE  ADD filePath to CDFFilePaths  ELSE  Do nothing | | |

{Ref: [1] <GENERIC\_TUM\_U2x\_GT\_TR\_005>}

#### overrideOptions

GENERIC\_GDD\_CLS\_048\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | overrideOptions | | |
| **Arguments** | options | Collection of command line options | |
| Range: | null/valid Dictionary<string, string> |
| **Return** | void | N/A | |
| Range: | N/A |
| **Generated Value** | None | | |
| **Description** | This method override options from command line  Algorithm: Loop all options and override it by options from command line | | |

{Ref: N/A}

## Renesas::Generator::MCALConfGen::Presentation::MainEntry::MainEntry

|  |  |
| --- | --- |
| **Location File Name** | MainEntry.cs |
| **Description** | This |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| - | - | - | - |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Main | Public | Main |
| (2) | unhandledExceptionHandler | Private | Handle unexpected exception |
| (3) | collectExceptionHierarchyInfo | Private | Collect information in exception object and print output |
| (4) | printHelpIfNeed | Private | print help if need |
| (5) | registerAssemblyTypes | Private | Register assembly types |
| (6) | loadConfiguration | Private | Load basic information from specific module such as module name, module id, project title, microfamily. |
| (7) | getModuleName | Private | Get module name |

### Fields

### Methods

#### Main

GENERIC\_GDD\_CLS\_049\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Main | | |
| **Arguments** | args | Args | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Main  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET initialize ICommandLine, IParser, IValidation, IIntermediate, IGeneration, IUserInterface,  IBasicConfiguration, IRepository  LET handle unexpected exception  LET handle command line  LET handle outline algorithm parse and validate CDFs in Autosar  LET configs = CALL GetConfigurations WITH basicConfig.ModuleName in repo  LET numberOfInstances = 0  FOREACH config in configs  LET defName = CALL GetElementLastInArray WITH DefinitionRef of config  IF defName is basicConfig.ModuleName THEN  numberOfInstances += 1  IF module is multi-instance THEN  LET apply output format of multi-instance module  Loop via instance and process each instance as normal instance. | | |

{Ref: N/A}

#### collectExceptionHierarchyInfo

GENERIC\_GDD\_CLS\_049\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | collectExceptionHierarchyInfo | | |
| **Arguments** | e | Exception | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Collect information in exception object and print output  Algorithm: LET stringBuilder = initialize new StringBuilder  IF e is not null THEN  LET appendLine (--, e.Message, e.StackTrace, collectExceptionHierarchyInfo(e.InnerException))  into stringBuilder  RETURN stringBuilder | | |

{Ref: N/A}

#### getModuleName

GENERIC\_GDD\_CLS\_049\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getModuleName | | |
| **Arguments** | args | Args | |
| Range: | None |
| **Return** | string | The name of module | |
| Range: | null/!=null |
| **Generated Value** | None | | |
| **Description** | Get module name  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A   LET Read module name from commandline input of user  RETURN module name | | |

{Ref: N/A}

#### loadConfiguration

GENERIC\_GDD\_CLS\_049\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | loadConfiguration | | |
| **Arguments** | assembly | Specific module assembly | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Load basic information from specific module such as module name, module id, project title, microfamily.  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   ABNORMAL SCENARIO:  N/A  LET name = GET FullName in assembly  LET resourceManager = initialize new ResourceManager  LET configuration = ObjectFactory.GetInstance<IBasicConfiguration>  LET configuration.ModuleName = rm.GetString(Constant.MODULE\_NAME)  LET configuration.ModuleId = int.Parse(rm.GetString(Constant.MODULE\_TOOL\_ID))  LET configuration.MicroFamilyName = rm.GetString(Constant.MICROFAMILY\_NAME)  LET configuration.ExecutionName = $"{configuration.ModuleName}\_{configuration.MicroFamilyName}.exe"  LET configuration.OutputFolder = $"{configuration.ExeDirectory}{configuration.ModuleName}\_Output"  LET configuration.ModuleRequired = rm.GetString(Constant.MODULE\_REQUIRED)  LET configuration.ProjectTitle = rm.GetString(Constant.PROJECT\_TITLE) | | |

{Ref: N/A}

#### printHelpIfNeed

GENERIC\_GDD\_CLS\_049\_005:

|  |  |
| --- | --- |
| **Method Name** | printHelpIfNeed |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | print help if need  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   IF user only input option -h THEN  LET print help content to user |

{Ref: N/A}

#### registerAssemblyTypes

GENERIC\_GDD\_CLS\_049\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | registerAssemblyTypes | | |
| **Arguments** | args | Args | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Register assembly types  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET scannedLocation = read dlls path  LET allDlls = Find all file have extension .dll  IF allDlls is not null THEN  LET moduleName = CALL getModuleName  LET selectedDlls = get dll with moduleName  LET moduleDLL = get first element array selectedDlls  LET SET ObjectFactory.ModuleName = moduleName  LET assembly = CALL ReflectionWrapper.LoadFrom WITH moduleDLL  IF dll does not contain module name or module name not extracted from Assembly Product Attribute  LET Handle error ERR000045 | | |

{Ref: [1] < GENERIC\_TUM\_U2X\_GT\_ERR\_045>}

{Ref: [3] < CMN\_GDD\_ERR\_045>}

#### unhandledExceptionHandler

GENERIC\_GDD\_CLS\_049\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | unhandledExceptionHandler | | |
| **Arguments** | sender | Sender | |
| Range: | None |
| e | Exception | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Handle unexpected exception  Algorithm: GLOBAL VARIABLE IN:  N/A   GLOBAL VARIABLE OUT:  N/A   PRECONDITION:  N/A   LET userInterface = CALL ObjectFactory.GetInstance WITH IUserInterface  LET ex = e.ExceptionObject  LET type = CALL ex.GetType()  IF type is MissingFieldException  LET Handle error ERR000043  ELSE  LET RAISE error  Check type of exception, print message then exit | | |

{Ref: [1] < GENERIC\_TUM\_U2X\_GT\_ERR\_043>}

{Ref: [3] < CMN\_GDD\_ERR\_043>}

# Data Types

## SectionName

GENERIC\_GDD\_DTT\_001**:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | enum | | |
| **Elements:** | COPY\_RIGHT | Where copy right section is generated | |
| **Range:** | None |
| REVISION\_HISTORY | Where revision history section is generated | |
| **Range:** | None |
| GENTOOL\_VERSION | Where gentool version section is generated | |
| **Range:** | None |
| INPUT\_FILES | Where input file section is generated | |
| **Range:** | None |
| MISRAC\_RULE | Where misrac rule section is generated | |
| **Range:** | None |
| QAC\_WARNING | Where qac warning section is generated | |
| **Range:** | None |
| START\_MACRO | Where start macro section is generated | |
| **Range:** | None |
| INCLUDE | Where include section is generated | |
| **Range:** | None |
| VERSION\_INFO | Where version information section is generated | |
| **Range:** | None |
| COMMON\_PUBLISHED\_INFO | Where common published info section is generated | |
| **Range:** | None |
| GLOBAL\_SYMBOLS | Where global symbols section is generated | |
| **Range:** | None |
| VERSION\_CHECK | Where version check section is generated | |
| **Range:** | None |
| GLOBAL\_DATA\_TYPE | Where global data type section is generated | |
| **Range:** | None |
| GLOBAL\_DATA | Where global data section is generated | |
| **Range:** | None |
| FUNCTION\_PROTOTYPE | Where function prototype section is generated | |
| **Range:** | None |
| FUNTION\_DEFINE | Where function define section is generated | |
| **Range:** | None |
| END\_MACRO | Where end macro section is generated | |
| **Range:** | None |
| END\_OF\_FILE | Where end of file section is generated | |
| **Range:** | None |
| INSTANCE\_INDEX | Where instance index is generated | |
| **Range:** | None |
| **Description:** | This enum is used to declare where output is generated | | |

*{Ref: }*

## MessageType

GENERIC\_GDD\_DTT\_002**:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | enum | | |
| **Elements:** | SUCCESS | Success | |
| **Range:** | None |
| ERROR | Error. | |
| **Range:** | None |
| WARN | Warning | |
| **Range:** | None |
| INFO | Information | |
| **Range:** | None |
| **Description:** | This enum is used to declare type of validation result | | |

*{Ref: }*

## ParseConfigStatus

GENERIC\_GDD\_DTT\_003**:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | enum | | |
| **Elements:** | Success | Success | |
| **Range:** | None |
| Fail | When cannot parse configuration | |
| **Range:** | None |
| NoData | When cannot get configuration after filter data. | |
| **Range:** | None |
| **Description:** | This enum is used to evaluate status after parser parse configuration from CDF. | | |

*{Ref: }*

# Error, Warning and Information Messages

This section contains the list of errors, warning and information messages that will be generated by the Generic Generation Tool. The messages will help to identify the syntax or semantic errors in the ECU Configuration Description File(s). Hence it will ensure validity and correctness of the information available in the ECU Configuration Description File(s).

The detail of error, warning and information messages are described at [1]. Please refer this document.

# Error handling

## Responsibility

Errror handling takes responsibility for:

* Unifing handling error process between modules
* Recognizing missed, redundant, or mismatched among CD, FD/DD and Requirements to support Traceability Management

## Implementation

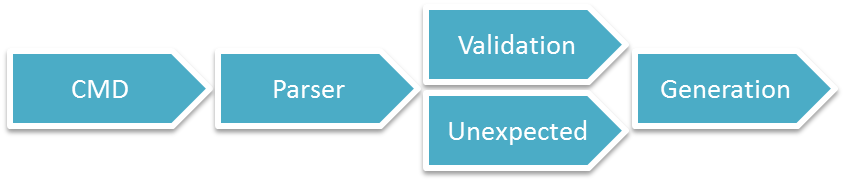
|  |  |
| --- | --- |
| **Phase** | Implementation |
| **Requirement** | AR-RS-0092.pdf.  Add one more functional requirement to describe how errors should be handled. |
| **FD/DD** | *AUTOSAR\_MSN\_Tool\_UserManual.docx*   * Add Error handling flow to Generic FD/DD * Group Error, Warning and Info messages into categories: Command Line, Parser, Validation, Generation, Unexpected. |
| **CD** | Handle errors in components below.   * Command Line (CMD) * Parser * Validation * Generation * Unexpected Errors |

Implemaintation in Coding phase (CD):

|  |  |
| --- | --- |
| **Module** | Implementation |
| **CMD** | Handle errors related to user inputs and default configuration XML (.cfgxml), e.g. invalid file name. |
| **Parser** | Handle error while parsing input files   1. CDF file: empty file content, XML format. 2. BSWMDT file: required information, e.g. module Id, module Name. 3. Translation configuration file (.trxml): empty file content, XML format, existence of translation/device header file. 4. Translation header file: duplicated macro name, empty value 5. Device header file: duplicated macro name, empty value. |
| **Validation** | 1. Pre-Validation: handle errors of CDF file content, e.g. mandatory parameters, or value of parameters in CDF file.   *NOTE:* Need a review of HW manual to confirm if TSDD is missing any validations.   1. Post-Validation: handle errors calculated by Intermediate. |
| **Generation** | Handle errors while writing output file \*.h and \*.c. |
| **Unexpected** | There are errors that are NOT covered by Gentool, called “external” errors, e.g. users edit CDF files manually or errors guaranteed by Davinci tool. However Gentool needs a common mechanic to prevent crash (if the “external” errors are “leaked” to the Gentool) and display a meaningful message to users. |

## Error handling flow

* Errors are handled in order of components.



* When Warning and Info messages are found, Gentool continues until it hits an error.
* When an error found, Gentool tries to collect all other errors that have the same type as the former, display them to end users and exit.

# Uncovered List

GENERIC\_TUM\_E2x\_GT\_UNCOVERED**:**

|  |  |  |
| --- | --- | --- |
| **Document name** | **Uncovered IDs** | **Reason** |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_TR\_002 | Description of Translation Header File should be presented in Driver DD |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_TR\_003 | Description of Device Specific Header Files should be presented in Driver DD |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_TR\_004 | It is only remark. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_CFG\_002 | It is only remark relate to Driver DD |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_USG\_001 | It does not present in DD because it only provides the information regarding usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_USG\_011 | It does not present in DD because it only provides the information regarding usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_USG\_012 | It does not present in DD because it only provides the information regarding usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_USG\_013 | It does not present in DD because it only provides the information regarding usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_USG\_014 | It does not present in DD because it only provides the information regarding usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_001 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_002 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_003 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_004 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_005 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_006 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_007 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_008 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_009 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_010 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_011 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_012 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_013 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_014 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_015 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_SAMPUSG\_016 | It does not present in DD because it only provides the information regarding sample usage of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_INST\_001 | It does not present in DD because it only provides the information regarding installation of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_UNINST\_001 | It does not present in DD because it only provides the information regarding uninstalling of the Generation Tool. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_ABBR\_001 | It does not present in DD because it only provides the information regarding name of version folder in repository. |
| 3\_GettingStarted\_MCAL\_Drivers\_U2x.docx | GENERIC\_TUM\_U2X\_GT\_UNCOVEREDGENERIC\_TUM\_E2x\_GT\_UNCOVERED GENERIC\_TUM\_E2x\_GT\_UNCOVERED | It does not present DD because it only provide the information reference table Uncovered List. |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Ver.** | **Change** | **Changed By** |
| 04-23-2019 | 1.0.0 | Initial version | Loc Nguyen |
| Update Class Diagram | Huy Tran |