

Ver.1.0.0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | D-RHA-235086 | | |
| Deliver to | Generic Generation Tool AUTOSAR R4.0.3 and R4.2.2  Software Detail Design Document | 03/01/20 | | |
| RVC/QA  RVC/PQA  RVC/PM, PL | Renesas Design Vietnam Co., Ltd.  Software Engineering Division  Software Engineering Department | | |
| Approved | Checked | Issued |
| Phuong Nguyen | Khanh Tran | TuPham |

Table of contents

[Chapter 0 - Preface 6](#_Toc29218925)

[0.1 Manual conventions 6](#_Toc29218926)

[0.2 Related documents 6](#_Toc29218927)

[0.3 How to read this document 6](#_Toc29218928)

[0.4 Abbreviations & Acronyms 7](#_Toc29218929)

[Chapter 1 - Introduction 8](#_Toc29218930)

[1.1 Scope 8](#_Toc29218931)

[1.2 Generation Tool Overview 8](#_Toc29218932)

[Chapter 2 - Architecture 9](#_Toc29218933)

[2.1 Overview of Generation Tool Architecture 9](#_Toc29218934)

[2.2 Dependencies to Other Modules 11](#_Toc29218935)

[Chapter 3 - Environment 12](#_Toc29218936)

[3.1 Hardware Requirements 12](#_Toc29218937)

[3.2 Software Requirements 12](#_Toc29218938)

[Chapter 4 - Design 13](#_Toc29218939)

[4.1 Concept 13](#_Toc29218940)

[4.1.1 Presentation 15](#_Toc29218941)

[4.1.2 Business 15](#_Toc29218942)

[4.1.3 Data 16](#_Toc29218943)

[4.1.4 Cross cutting 17](#_Toc29218944)

[4.2 File structure 18](#_Toc29218945)

[Chapter 5 - Class 21](#_Toc29218946)

[5.1 CommandLine 22](#_Toc29218947)

[5.1.1 UML 22](#_Toc29218948)

[5.1.2 Renesas::Generator::MCALConfGen::Business::CommandLine::CommandLine 22](#_Toc29218949)

[5.1.3 Renesas::Generator::MCALConfGen::Business::CommandLine::CommandLineInput 35](#_Toc29218950)

[5.1.4 Renesas::Generator::MCALConfGen::Business::CommandLine::OptionAttribute 37](#_Toc29218951)

[5.2 Generation 38](#_Toc29218952)

[5.2.1 UML 39](#_Toc29218953)

[5.2.2 Renesas::Generator::MCALConfGen::Business::Generation::FileGeneration 41](#_Toc29218955)

[5.2.3 Renesas::Generator::MCALConfGen::Business::Generation::Generation 49](#_Toc29218956)

[5.2.4 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::BreakLineGenerationItem 51](#_Toc29218957)

[5.2.5 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructMemberGenerationItem 53](#_Toc29218958)

[5.2.6 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::BaseGenerationItem 54](#_Toc29218959)

[5.2.7 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::CommentGenerationItem 61](#_Toc29218960)

[5.2.8 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::DefineGenerationItem 62](#_Toc29218961)

[5.2.9 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StringGenerationItem 65](#_Toc29218962)

[5.2.10 Renesas::Generator::MCALConfGen::Business::Generation::GenerationItems::StructGenerationItem 66](#_Toc29218963)

[5.2.11 Renesas::Generator::MCALConfGen::Business::Generation::HeaderFileGeneration 68](#_Toc29218964)

[5.2.12 Renesas::Generator::MCALConfGen::Business::Generation::ItemGenerationAttribute 72](#_Toc29218965)

[5.2.13 Renesas::Generator::MCALConfGen::Business::Generation::NewLineAttribute 74](#_Toc29218966)

[5.2.14 Renesas::Generator::MCALConfGen::Business::Generation::Section 75](#_Toc29218967)

[5.2.15 Renesas::Generator::MCALConfGen::Business::Generation::SourceFileGeneration 77](#_Toc29218968)

[5.3 Intermediate 79](#_Toc29218969)

[5.3.1 UML 80](#_Toc29218970)

[5.3.2 Renesas::Generator::MCALConfGen::Business::Intermediate::BaseIntermediate 80](#_Toc29218971)

[5.4 Parser 83](#_Toc29218972)

[5.4.1 UML 83](#_Toc29218973)

[5.4.2 Renesas::Generator::MCALConfGen::Business::Parser::ParserXML 84](#_Toc29218974)

[5.4.3 ParseConfigStatus 107](#_Toc29218975)

[5.5 Validation 107](#_Toc29218976)

[5.5.1 UML 108](#_Toc29218977)

[5.5.2 Renesas::Generator::MCALConfGen::Business::Validation::BaseValidation 108](#_Toc29218978)

[5.5.3 Renesas::Generator::MCALConfGen::Business::Validation::ValidationResult 121](#_Toc29218979)

[5.5.4 Renesas::Generator::MCALConfGen::Business::Validation::ValidationRuleAttribute 125](#_Toc29218980)

[5.6 DotNetApiWrapper 128](#_Toc29218981)

[5.6.1 UML 129](#_Toc29218982)

[5.6.2 Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::ConsoleWrapper 129](#_Toc29218983)

[5.6.3 Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::EnvironmentWrapper 130](#_Toc29218984)

[5.6.4 Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::IOWrapper 131](#_Toc29218985)

[5.6.5 Renesas::Generator::MCALConfGen::CrossCutting::DotNetApiWrapper::ReflectionWrapper 139](#_Toc29218986)

[5.7 ObjectFactory 142](#_Toc29218987)

[5.7.1 UML 142](#_Toc29218988)

[5.7.2 Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactory 143](#_Toc29218989)

[5.7.3 Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactory::CacheObject 148](#_Toc29218990)

[5.7.4 Renesas::Generator::MCALConfGen::CrossCutting::ObjectFactory::ObjectFactoryAttribute 150](#_Toc29218991)

[5.8 UserInterface 152](#_Toc29218992)

[5.8.1 UML 152](#_Toc29218993)

[5.8.2 Renesas::Generator::MCALConfGen::CrossCutting::UserInterface::UserInterface 152](#_Toc29218994)

[5.9 Util 164](#_Toc29218995)

[5.9.1 UML 164](#_Toc29218996)

[5.9.2 Renesas::Generator::MCALConfGen::CrossCutting::Util::Constant 164](#_Toc29218997)

[5.9.3 Renesas::Generator::MCALConfGen::CrossCutting::Util::StringUtils 203](#_Toc29218998)

[5.10 Data 211](#_Toc29219000)

[5.10.1 UML 211](#_Toc29219001)

[5.10.2 Renesas::Generator::MCALConfGen::Data::BasicConfiguration::BasicConfiguration 213](#_Toc29219002)

[5.10.3 Renesas::Generator::MCALConfGen::Data::CDFData::BaseRegisterProcessing 217](#_Toc29219003)

[5.10.4 Renesas::Generator::MCALConfGen::Data::CDFData::CdfDataDictionary 223](#_Toc29219004)

[5.10.5 Renesas::Generator::MCALConfGen::Data::CDFData::Items::BswConfig 233](#_Toc29219005)

[5.10.6 Renesas::Generator::MCALConfGen::Data::CDFData::Items::Configuration 236](#_Toc29219006)

[5.10.7 Renesas::Generator::MCALConfGen::Data::CDFData::Items::Container 238](#_Toc29219007)

[5.10.8 Renesas::Generator::MCALConfGen::Data::CDFData::Items::ItemValue 246](#_Toc29219008)

[5.10.9 Renesas::Generator::MCALConfGen::Data::CDFData::RepositoryLinQ 249](#_Toc29219009)

[5.10.10 Renesas::Generator::MCALConfGen::Data::RuntimeConfiguration::RuntimeConfiguration 264](#_Toc29219010)

[5.11 Intermediate 272](#_Toc29219011)

[5.11.1 UML 273](#_Toc29219012)

[5.11.2 Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateData 273](#_Toc29219013)

[5.11.3 Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateItemAttribute 276](#_Toc29219014)

[5.11.4 Renesas::Generator::MCALConfGen::Data::IntermediateData::Items::BaseIntermediateItem 278](#_Toc29219015)

[5.12 MainEntry 288](#_Toc29219018)

[5.12.1 UML 288](#_Toc29219019)

[5.12.2 Renesas::Generator::MCALConfGen::Presentation::MainEntry::MainEntry 288](#_Toc29219020)

[Chapter 6 - Error, Warning and Information Messages 293](#_Toc29219026)

[Chapter 7 - Error handling 294](#_Toc29219027)

[7.1 Responsibility 294](#_Toc29219028)

[7.2 Implementation 294](#_Toc29219029)

[7.3 Error handling flow 295](#_Toc29219030)

[Chapter 8 - Uncovered List 296](#_Toc29219031)

[Chapter 9 - Revision History 297](#_Toc29219032)

***List of Figures***

[Figure 1 Generic Generation Tool Hierarchical Architecture 9](D:\\Workspace\\Projects\\Commonize\\X2xCommonize\\X2x\\internal\\X2x\\gen_tool\\0x_DD\\01_WorkProduct\\generic\\RH850_X2x_GENERIC_GenTool_DD.docx" \l "_Toc29219202)

[Figure 2 Concept of Generic Generation Tool 13](D:\\Workspace\\Projects\\Commonize\\X2xCommonize\\X2x\\internal\\X2x\\gen_tool\\0x_DD\\01_WorkProduct\\generic\\RH850_X2x_GENERIC_GenTool_DD.docx" \l "_Toc29219203)

[Figure 3. Overview of class diagram 21](D:\\Workspace\\Projects\\Commonize\\X2xCommonize\\X2x\\internal\\X2x\\gen_tool\\0x_DD\\01_WorkProduct\\generic\\RH850_X2x_GENERIC_GenTool_DD.docx" \l "_Toc29219204)

# 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\_X2x\_Generic\_GenTool\_ErrorList.xls | 1.0 |

## How to read this document

| **Section** | **Contents** |
| --- | --- |
| Section 1 (Introduction) | This section contains the Scope of the document and Generation Tool overview. |
| Section 2 (Architecture) | This section contains overview of Generic Generation Tool architecture and dependency to other modules. |
| Section 3 (Environment) | This section contains Hardware/Software requirements. |
| Section 4 (Design) | This section contains Concept, File Structure for Generic Generation Tool. |
| Section 5 (Class) | This section contains Class and Function Flow for Generic Generation Tool. |
| Section 6 (Error, Warning and Information Messages) | This section provides Error, Warning and Information Messages. |
| Section 7 (Error handling) | This section describe how to handle errors in Generic module. |
| Section 8 (Uncovered List) | This section fill uncovered ID and reason in traceability |
| Section 9 (Revision History) | This section record revision history of document. |

## Abbreviations & Acronyms

| **Abbreviations/Acronym** | **Description** |
| --- | --- |
| API | Application Programming Interface |
| AUTOSAR | AUTomotive Open System ARchitecture |
| BSWMDT | Basic Software Module Description Template |
| CFD | Control Flow Diagram |
| CFGXML | Configuration XML File |
| DEM | Diagnostic Event Manager |
| DFD | Data Flow Diagram |
| ECU | Electronic Control Unit |
| Id | Identifier |
| IMR | Interrupt Mask Register |
| MHz | Mega Hertz |
| MCU | MicroController Unit |
| SW | SoftWare |
| TRXML | Translation XML File |
| XML | eXtensible Markup Language |

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

# Introduction

## Scope

This document provides generation Tool design for development of R4.0.3 and R4.2.2 AUTOSAR Common Executable for Renesas X2x. It also provides used, un-used parameters/container list, error, warning and information messages that are specific to AUTOSAR R4.0.3 and R4.2.2 and Renesas X2x 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

<MSN>Pre-validation

Pre-validation

Post-validation

Parser

Object Factor

Command Line

C Source & Header Files

Figure 1 Generic Generation Tool Hierarchical Architecture

{Ref: GENERIC\_TFD\_X2x\_GT\_OVV\_001, GENERIC\_TFD\_X2x\_GT\_OVV\_002}

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.

{Ref: GENERIC\_TFD\_X2x\_GT\_TR\_001, GENERIC\_TFD\_X2x\_GT\_TR\_002, GENERIC\_TFD\_X2x\_GT\_TR\_003, GENERIC\_TFD\_X2x\_GT\_TR\_004}

* 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.

{Ref: GENERIC\_TFD\_X2x\_GT\_CFG\_001, GENERIC\_TFD\_X2x\_GT\_CFG\_002}

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.

* 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: }

## 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 2GB or greater |
| **Hard Disk Drive** | 20GB or greater storeage capacity |

{Ref: GENERIC\_TFD\_X2x\_GT\_INSTREQ\_002}

## Software Requirements

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

{Ref: GENERIC\_TFD\_X2x\_GT\_INSTREQ\_001, GENERIC\_TFD\_X2x\_GT\_INST\_001, GENERIC\_TFD\_X2x\_GT\_UNINST\_001}

# Design

## Concept

Concept of execution for Generation Tool is as follows:

**Runtime Configuration**

**Command Line**

**CDF Data**

**Main Enty**

Presentation Layer

Data Layer

Business Layer

**Parser**

**Validation**

**Intermediate Data**

**Intermediate**

**Generation**

Figure 2 Concept of Generic Generation Tool

**DotNetApiWrapper**

**UserInterface**

**Util**

**ObjectFactory**

**Util**

**User Interface**

**Object Factory**

{Ref: **}**

**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:

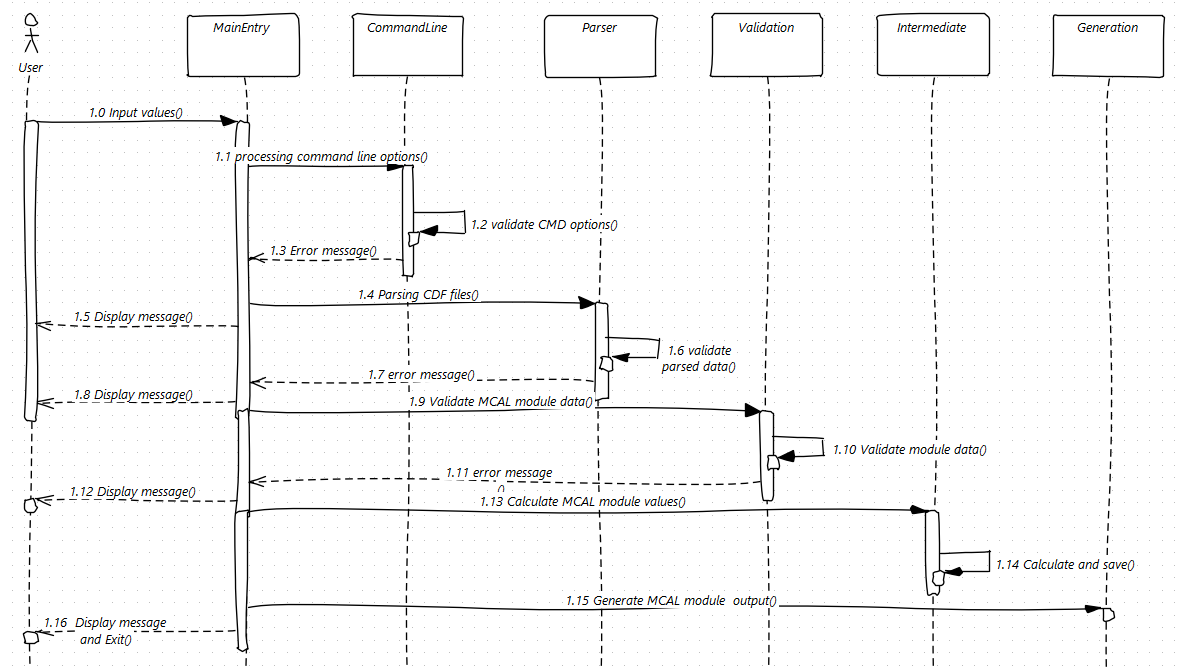
* Initiate loading DLLs of a specific module.
* Load basic configuration information.
* Read and process command line inputs.
* Parse input files.
* (Pre-)Validate unexpected values in CDF files.
* Calculate intermediate data.
* (Post-)Validate intermediate data.
* Generate header files (\*.h), source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

Below is main flow to describe collaboration of components with others.



### 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

#### 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. |

{Ref: GENERIC\_TFD\_X2x\_GT\_USG\_001}

#### 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. |

#### DotNetApiWrapper

|  |  |
| --- | --- |
| Location | The “DotNetApiWrapper” folder is as following:  1. MCALConfGen/ CrossCutting/ DotNetApiWrapper |
| Input | None |
| Output | Wrap all kind of .NET Error messages |

## File structure

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

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

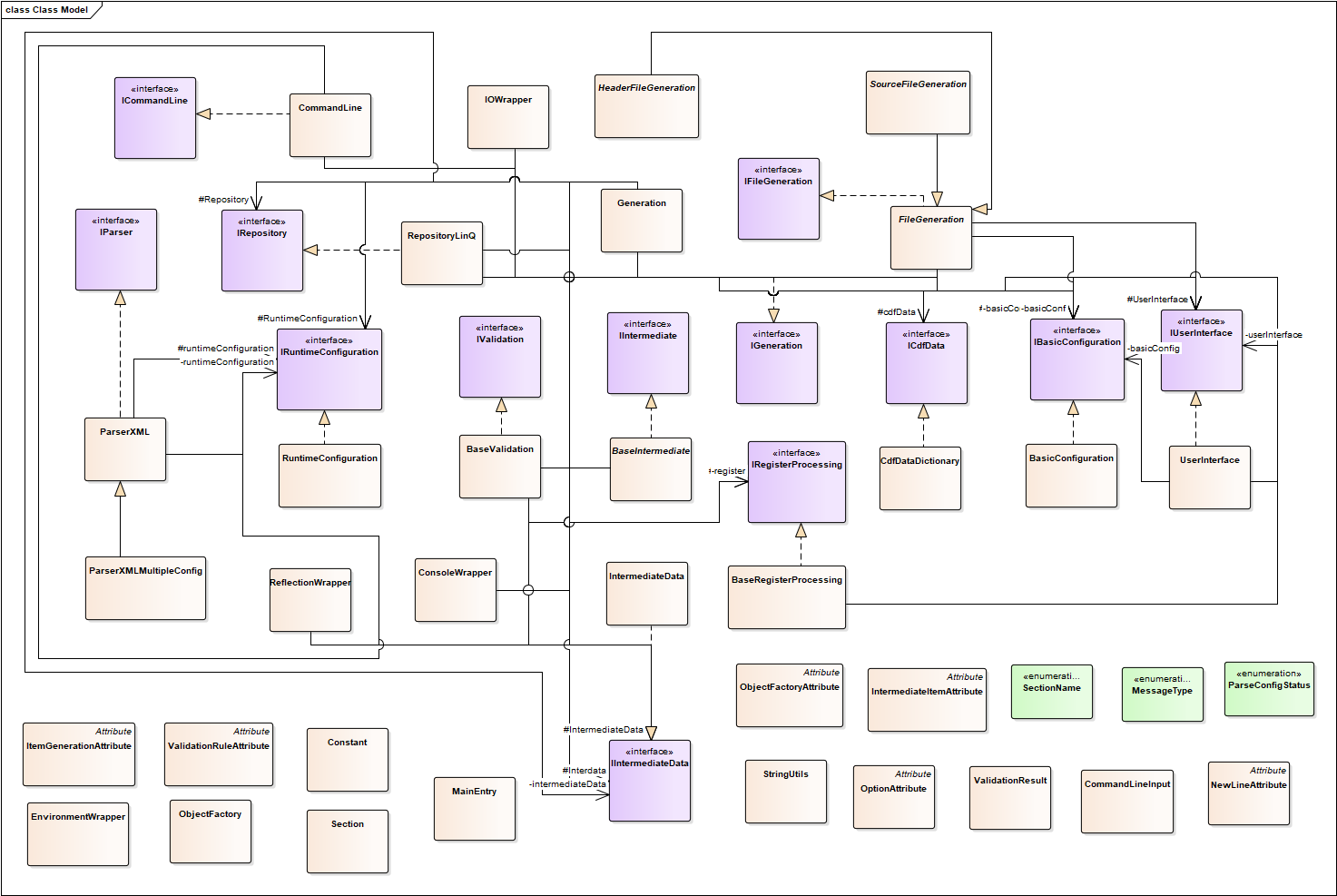
Table 1: Generic File List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MCALConfGen** | | | | |
| +---Business | | | | |
| | | +---CommandLine | | | |
| | | | |  | CommandLine.cs | |
| | | | |  | CommandLineInput.cs | |
| | | | |  | ICommandLine.cs | |
| | | | |  | OptionAttribute.cs | |
| | | +---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 |
| | | +---Intermediate | | | |
| | | | |  | BaseIntermediate.cs | |
| | | | |  | IIntermediate.cs | |
| | | +---Parser | | | |
| | | | |  | IParser.cs | |
| | | | |  | ParserXML.cs | |
| | | \---Validation | | | |
| | |  |  | BaseValidation.cs | |
| | |  |  | IValidation.cs | |
| | |  |  | ValidationResult.cs | |
| | |  |  | ValidationRuleAttribute.cs | |
| +---CrossCutting | | | | |
| | | +---DotNetApiWrapper | | | |
| | | | |  | ConsoleWrapper.cs | |
| | | | |  | EnvironmentWrapper.cs | |
| | | | |  | IOWrapper.cs | |
| | | | |  | ReflectionWrapper.cs | |
| | | +---ObjectFactory | | | |
| | | | |  | ObjectFactory.cs | |
| | | | |  | ObjectFactoryAttribute.cs | |
| | | +---UserInterface | | | |
| | | | |  | IUserInterface.cs | |
| | | | |  | UserInterface.cs | |
| | | \---Util | | | |
| | |  |  | Constant.cs | |
| | |  |  | StringUtils.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 Generic Generation Tool.



{Ref: }

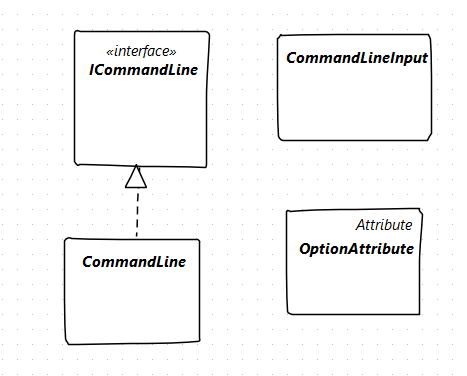
Note: All methods with type “Protected” are represented by “protected virtual” in source code and can be overrided by subclasses.

Figure 3. Overview of class diagram

## CommandLine

CommandLine class bear the responsibility for handling user input information such as command line options, cdf file path, bswmdt file path,…

### UML



Use Interface ICommandLine to declare methods that handle user input information. In addition to CommandLine class implement Interface, CommandLineInput class contain values that are provided by user such as cdf file (can.arxml, mcu.arxml,…), options: –h, -l, -o, -c,…..

Some command line errors are also processed by class CommandLine such as does not exist input file, invalid option,…

OptionAttribute class manage all valid options in Generation Tool such as -h, -o, -c, -l, ….

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

|  |  |
| --- | --- |
| **Location File Name** | CommandLine.cs |
| **Description** | CommandLine class to handle input from command lines |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | CommandLine instance used by ObjectFactory to get a new CommandLine object |
| (2) | runtimeConfiguration | Private | Runtime configuration contains values of option command line that is input from user |
| (3) | basicConf | Private | Basic configuration contains basic information of project such as micro family, DeviceNames, module id,... |
| (4) | userInterface | Private | User interface is used to report error/info/warning to user |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | PrintToolHelp | Public | This method provide help information of |
| (2) | GetAndValidateModuleName | Public | Get and validate module name from command line options -m/-module |
| (3) | ProcessCommandLine | Public | Process command line. A composite method to coordinate other methods to process command line options |
| (4) | CommandLine | Protected | Command line constructor without arguments |
| (5) | CommandLine | Protected | Command line constructor with arguments |
| (6) | displayCmdInformation | Private | Display command line information of user input |
| (7) | parseAndValidateOptions | Private | Parse and validate options from command line and config xml file |
| (8) | parseUserInputs | Private | Parse all information from command line options entried by users |
| (9) | filterValueVendorName | Private | Check value of option -fr/-filter from command line or config xml file. |
| (10) | checkConfigOptionFileName | Private | Check value of option -c/-config from command line or config xml file. |
| (11) | checkInvalidOption | Private | Check invalid options that is not in supported option by command line tool |
| (12) | checkOutputDirectory | Private | Check value of option -o/-output from command line and config xml file |
| (13) | validateUserInputs | Private | Validate command line options by calling sub methods |
| (14) | parseXmlConfig | Private | Read all value of tags in config xml file and store value. |
| (15) | validateXmlConfig | Private | Check values of options which is provided by config xml file. |
| (16) | checkExplicitOptions | Private | Store options that value of options is explicit provided by command line or config xml file. |
| (17) | validateBothUserAndXmlInputs | Private | Check exists options from command line and config xml file. |
| (18) | validateAcceptedFiles | Private | Validate input files with extension .trxml, .arxml |
| (19) | showInvalidInputPath | Private | Check exists of file CDFs input from command line and config xml file |
| (20) | showInvalidOutputPath | Private | Check value of option -o/-output from command line and config xml file. |
| (21) | getOptionValueFromXmlConfig | Private | Get an option value from .cfgxml file |
| (22) | checkMissingValueForOption | Private | Checking missing value for options command line. |

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_001\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly CommandLine |
| **Range** | None |
| **Description** | CommandLine instance used by ObjectFactory to get a new CommandLine object |

##### runtimeConfiguration

GENERIC\_GDD\_CLS\_001\_002:

|  |  |
| --- | --- |
| **Variable Name** | runtimeConfiguration |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | Runtime configuration contains values of option command line that is input from user |

##### basicConf

GENERIC\_GDD\_CLS\_001\_003:

|  |  |
| --- | --- |
| **Variable Name** | basicConf |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | Basic configuration contains basic information of project such as micro family, DeviceNames, module id,... |

##### userInterface

GENERIC\_GDD\_CLS\_001\_004:

|  |  |
| --- | --- |
| **Variable Name** | userInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | User interface is used to report error/info/warning to user |

#### Methods

##### GetAndValidateModuleName

GENERIC\_GDD\_CLS\_001\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAndValidateModuleName | | |
| **Arguments** | None | | |
| **Return** | RETURN a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: [1] CMN\_GFD\_ERR\_031, CMN\_GFD\_INF\_005, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_001, GENERIC\_TFD\_X2x\_GT\_INF\_005}

##### PrintToolHelp

GENERIC\_GDD\_CLS\_001\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | PrintToolHelp | | |
| **Arguments** | None | | |
| **Return** | RETURN a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | This method provide help information of  Algorithm: Get options from command line and config xml file  Display help information when user input -h/-help from command line options  or option in config xml file. | | |

{Ref: GENERIC\_TFD\_X2x\_GT\_USG\_002, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_002}

##### ProcessCommandLine

GENERIC\_GDD\_CLS\_001\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ProcessCommandLine | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Process command line. A composite method to coordinate other methods to process command line options  Algorithm: 1. Call displayCmdInformation to display command line information  2. Call parseAndValidateOptions to parse and validate options  3. Call printToolHelp to print help information  4. Call showInvalidOutputPath to validate output path  5. Call showInvalidInputPath to validate input path | | |

{Ref: N/A}

##### CommandLine

GENERIC\_GDD\_CLS\_001\_008:

|  |  |
| --- | --- |
| **Method Name** | CommandLine |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Command line constructor without arguments |

{Ref: N/A}

##### CommandLine

GENERIC\_GDD\_CLS\_001\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CommandLine | | |
| **Arguments** | runtimeConfiguration | Runtime configuration | |
| Range: | None |
| basicConfiguration | Basic configuration | |
| Range: | None |
| userInterface | User Interface | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Command line constructor with arguments  Algorithm: Init IRuntimeConfiguration, IBasicConfiguration, IUserInterface | | |

{Ref: N/A}

##### checkConfigOptionFileName

GENERIC\_GDD\_CLS\_001\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkConfigOptionFileName | | |
| **Arguments** | userInputs | User inputs | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Algorithm** |  | | |

{Ref: [1] CMN\_GFD\_ERR\_002, GENERIC\_TFD\_X2x\_GT\_ERR\_002}

##### checkExplicitOptions

GENERIC\_GDD\_CLS\_001\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkExplicitOptions | | |
| **Arguments** | userInput | User input | |
| Range: | None |
| xmlConfig | Xml config | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Store options that value of options is explicit provided by command line or config xml file.  Algorithm: Read options -h, -d, -l from command line and config xml file  if value of options is not provided such as TRUE/FALSE  concatenate option into string  return string | | |

{Ref: }

##### checkInvalidOption

GENERIC\_GDD\_CLS\_001\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkInvalidOption | | |
| **Arguments** | userInputs | User inputs | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check invalid options that is not in supported option by command line tool  Algorithm: get all options from command line input from user  if have any option which is not supported by generation tool  report ERR000008  if have any option duplication  report WRN000002 | | |

{Ref: [1] CMN\_GFD\_ERR\_008, CMN\_GFD\_WRN\_002, GENERIC\_TFD\_X2x\_GT\_ERR\_008, GENERIC\_TFD\_X2x\_GT\_WRN\_002}

##### checkOutputDirectory

GENERIC\_GDD\_CLS\_001\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkOutputDirectory | | |
| **Arguments** | userInputs | User inputs | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check value of option -o/-output from command line and config xml file  Algorithm: if exists option -o/-output and value of option is not provided by command line or exists tags OUTPUT  and value of tag in not provided by config xml file.  report ERR000005  Or if value of option is invalid such as contains special characters  report ERR000010 | | |

{Ref: [1] CMN\_GFD\_ERR\_005, CMN\_GFD\_ERR\_010, GENERIC\_TFD\_X2x\_GT\_USG\_005, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_014, GENERIC\_TFD\_X2x\_GT\_ERR\_005, GENERIC\_TFD\_X2x\_GT\_ERR\_010}

##### displayCmdInformation

GENERIC\_GDD\_CLS\_001\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | displayCmdInformation | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Display command line information of user input  Algorithm: Display information files input from user as path file.  Report information INF000001, INF000002 | | |

{Ref: [1] CMN\_GFD\_INF\_001, CMN\_GFD\_INF\_002, GENERIC\_TFD\_X2x\_GT\_INF\_001, GENERIC\_TFD\_X2x\_GT\_INF\_002}

##### filterValueVendorName

GENERIC\_GDD\_CLS\_001\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | filterValueVendorName | | |
| **Arguments** | userInputs | User inputs | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check value of option -fr/-filter from command line or config xml file.  Algorithm: If exists option -fr/-filter and value of option is not provided by command line or config xml file  report ERR000032  or if value is invalid such as invalid name, contain special characters  report ERR000033 | | |

{Ref: [1] CMN\_GFD\_ERR\_032, CMN\_GFD\_ERR\_033, GENERIC\_TFD\_X2x\_GT\_ERR\_032}

##### getOptionValueFromXmlConfig

GENERIC\_GDD\_CLS\_001\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getOptionValueFromXmlConfig | | |
| **Arguments** | XDocument document | Parsed .cfgxml file | |
| Range: | None |
| string option | Option | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Scan .cfgxml file to get value of the option.  Algorithm: Scan .cfgxml file to get value of the option.  IF found AND value of tag FILTER is invalid  report to ERR000033  ELSE  add value to return value | | |

{Ref: [1] CMN\_GFD\_ERR\_033, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_012, GENERIC\_TFD\_X2x\_GT\_ERR\_033}

##### parseAndValidateOptions

GENERIC\_GDD\_CLS\_001\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseAndValidateOptions | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parse and validate options from command line and config xml file  Algorithm: 1. Parse all information from command line options  2. Validate the command line options to handle errors and warning  ERR000008  ERR000002  ERR000005  ERR000010  ERR000032  WRN000001  3. If configuration file is NOT specified by -c/-configfile  Use default configuration file {MSN}\_{MICRO\_FAMILY\_NAME}.cfgxml at the same MCALConfGen.exe location  End  4. Parse all information from .cfgxml file  5. Validate options from .cfgxml files to handle errors and warning  ERR000006  ERR000033  WRN000001  6. If output folder is NOT specified by -o/-output  Use default output folder {MODULE\_NAME}\_Output  End  7. Merge all options from command line and .cfgxml into runtimeConfiguration.  Command line options override .cfgxml options  8. Validate both command line options and .cfgxml file options to handle errors  ERR000007 | | |

{Ref: [1] CMN\_GFD\_ERR\_002, CMN\_GFD\_ERR\_005, CMN\_GFD\_ERR\_006, CMN\_GFD\_ERR\_007, CMN\_GFD\_ERR\_008, CMN\_GFD\_ERR\_010, CMN\_GFD\_ERR\_032, CMN\_GFD\_ERR\_033, CMN\_GFD\_WRN\_001, GENERIC\_TFD\_X2x\_GT\_USG\_004, GENERIC\_TFD\_X2x\_GT\_USG\_012, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_009, GENERIC\_TFD\_X2x\_GT\_ERR\_006, GENERIC\_TFD\_X2x\_GT\_WRN\_001}

##### parseUserInputs

GENERIC\_GDD\_CLS\_001\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseUserInputs | | |
| **Arguments** | None | | |
| **Return** | return a | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Parse all information from command line options entried by users  Algorithm: Get options from command line and config xml file  Store value of options such as -h/-help, -m/-module, -c/-config,.....  Store option duplication to report info if have any. | | |

{Ref: GENERIC\_TFD\_X2x\_GT\_USG\_003}

##### parseXmlConfig

GENERIC\_GDD\_CLS\_001\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseXmlConfig | | |
| **Arguments** | string path | Path to .cfgxml file | |
| Range: | None |
| **Return** | return a CommandLineInput | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Read all value of tags in config xml file and store value.  Algorithm: Read config xml file and store values when match to tags in Generation Tools such as  OUTPUT-PATH, HELP, INPUT-FILE, FILTER,...  if not exists file  report ERR000001 | | |

{Ref: [1] CMN\_GFD\_ERR\_001, GENERIC\_TFD\_X2x\_GT\_ERR\_001}

##### showInvalidInputPath

GENERIC\_GDD\_CLS\_001\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | showInvalidInputPath | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check exists of file CDFs input from command line and config xml file  Algorithm: Get file information which input from command line and config xml file.  If file information is not provided by user  report ERR000011  if not exists file path  report ERR000001 | | |

{Ref: [1] CMN\_GFD\_ERR\_001, CMN\_GFD\_ERR\_011, GENERIC\_TFD\_X2x\_GT\_ERR\_011}

##### showInvalidOutputPath

GENERIC\_GDD\_CLS\_001\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | showInvalidOutputPath | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check value of option -o/-output from command line and config xml file.  Algorithm: If folder name is the same file name  report ERR000009  if cannot create folder with any reason  report ERR000018 | | |

{Ref: [1] CMN\_GFD\_ERR\_009, CMN\_GFD\_ERR\_018, GENERIC\_TFD\_X2x\_GT\_ERR\_009}

##### validateAcceptedFiles

GENERIC\_GDD\_CLS\_001\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | validateAcceptedFiles | | |
| **Arguments** | List< string > filePaths | File paths | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Validate input files with extension .trxml, .arxml  Algorithm: Get extension of file input from command line and config xml file  If have any extension which is not .trxml or .arxml  report warning WRN000001 for violated file | | |

{Ref: [1] CMN\_GFD\_WRN\_001}

##### validateBothUserAndXmlInputs

GENERIC\_GDD\_CLS\_001\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | validateBothUserAndXmlInputs | | |
| **Arguments** | userInput | User input | |
| Range: | None |
| xmlConfig | Xml config | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check exists options from command line and config xml file.  Algorithm: If options is not provided by command line and config xml file  report ERR000007, INF000003  if have any value of option is not provided such as TRUE/FALSE  report ERR000034 | | |

{Ref: [1] CMN\_GFD\_ERR\_007, CMN\_GFD\_ERR\_034, CMN\_GFD\_INF\_003, GENERIC\_TFD\_X2x\_GT\_ERR\_007, GENERIC\_TFD\_X2x\_GT\_ERR\_034, GENERIC\_TFD\_X2x\_GT\_INF\_003}

##### validateUserInputs

GENERIC\_GDD\_CLS\_001\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | validateUserInputs | | |
| **Arguments** | userInputs | User inputs | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Validate command line options by calling sub methods  Algorithm: Validate the command line options to handle errors and warning  Invoke method checkInvalidOption()  Invoke method checkConfigOptionFileName()  Invoke method checkOutputDirectory()  Invoke method filterValueVendorName()  Invoke method validateAcceptedFiles() | | |

{Ref: N/A}

##### validateXmlConfig

GENERIC\_GDD\_CLS\_001\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | validateXmlConfig | | |
| **Arguments** | string xmlConfigFile | .cfgxml files file path | |
| Range: | None |
| xmlConfig | Xml config | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check values of options which is provided by config xml file.  Algorithm: Get option in tag OUTPUT-FILE  if not exists value of option  report ERR000006  If extension of files is not .trxml and .arxml  report warning WRN000001 | | |

{Ref: [1] CMN\_GFD\_ERR\_006, CMN\_GFD\_WRN\_001}

##### checkMissingValueForOption

GENERIC\_GDD\_CLS\_001\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkMissingValueForOption | | |
| **Arguments** | string options | Options like -c, -fr,.. | |
| Range: | None |
| userInputs | Contains options’ values | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Algorithm** |  | | |

{Ref: [1] CMN\_GFD\_ERR\_006, CMN\_GFD\_WRN\_001}

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

|  |  |
| --- | --- |
| **Location File Name** | CommandLineInput.cs |
| **Description** | CommandLineInput class that store options, their values and files entried by user. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | OptionsWithValues | Public | Contains options with their values such as {"-o":"D:\\test"}, {"-fr":"renesas"},... |
| (2) | OptionsAppearances | Public | Contain options duplication |
| (3) | Files | Public | Files |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | CommandLineInput | Public | Command line input constructor without arguments |

#### Fields

##### OptionsWithValues

GENERIC\_GDD\_CLS\_002\_001:

|  |  |
| --- | --- |
| **Variable Name** | OptionsWithValues |
| **Type** | Dictionary< string, string > |
| **Range** | None |
| **Description** | Contains options with their values such as {"-o":"D:\\test"}, {"-fr":"renesas"},... |

{Ref: GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_010}

##### OptionsAppearances

GENERIC\_GDD\_CLS\_002\_002:

|  |  |
| --- | --- |
| **Variable Name** | OptionsAppearances |
| **Type** | Dictionary< string, int > |
| **Range** | None |
| **Description** | Contain options duplication |

##### 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 without arguments  Algorithm: Init variables such as OptionsWithValues, OptionsAppearances, Files |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | OptionAttribute.cs |
| **Description** | OptionAttribute class that decorates for a fields which store an cmd option. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | option | Private | Contains option name information such as -h, -o, -c, -fr,... |

1. Method List

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

#### Fields

##### option

GENERIC\_GDD\_CLS\_003\_001:

|  |  |
| --- | --- |
| **Variable Name** | option |
| **Type** | string |
| **Range** | None |
| **Description** | Contains option name information such as -h, -o, -c, -fr,... |

#### Methods

##### GetOption

GENERIC\_GDD\_CLS\_003\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetOption | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get option name  Algorithm: Access option name | | |

{Ref: N/A}

##### OptionAttribute

GENERIC\_GDD\_CLS\_003\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | OptionAttribute | | |
| **Arguments** | string option | Option | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Option attribute constructor with argument option  Algorithm: Init option name | | |

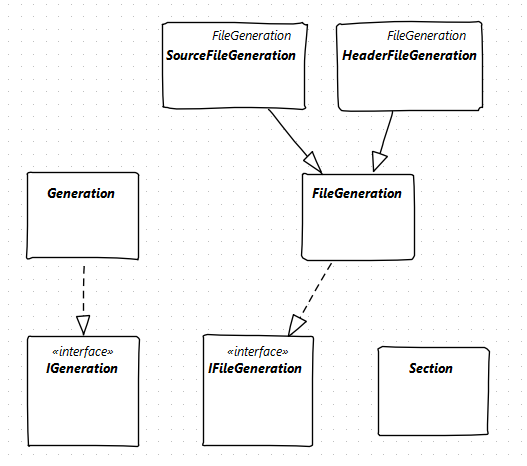
{Ref: N/A}

## Generation

Generation component process output of Generation Tool such as Header and Source files.

Ex. Lin\_Cbk.h, Lin\_Cfg.h, Lin\_PBcfg.c

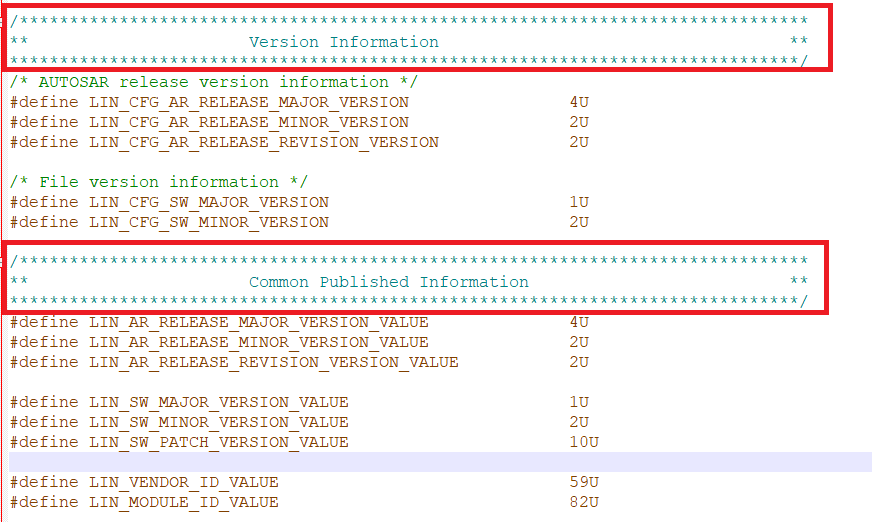
### UML



Generation class implement from IGeneration to generate all files (.c, .h) in Generation Tool. Each MCAL module need extending base class such as SourceFileGeneration to define output file (.c) and HeaderFileGeneration to define output file (.h)

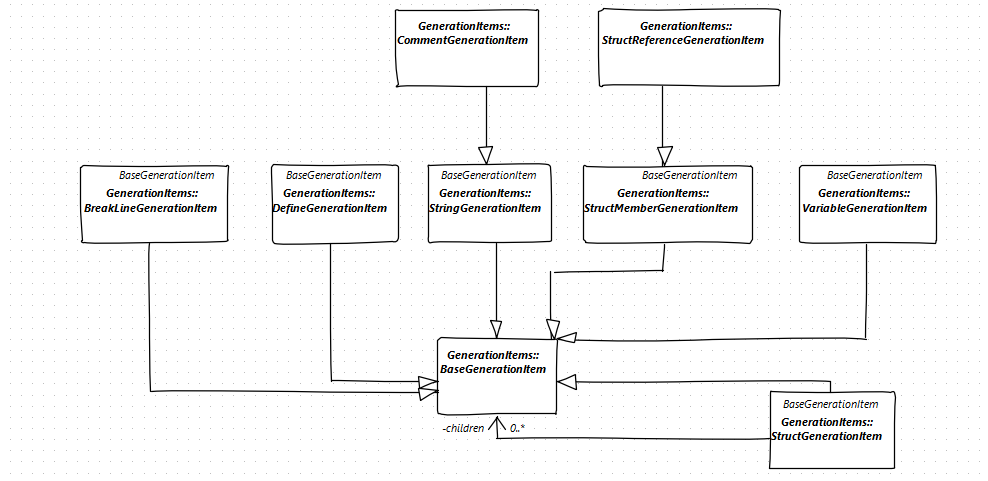
Common information will be stored on parent class as FileGeneration, objective is re-use information both Header and Source file. In addition, this class define steps to process output file.

Section class handle section template in Header or Source files, section group generated parameters such as Global Data Types, Global Data, Function Definitions, Version Information, Common Published Information,…..



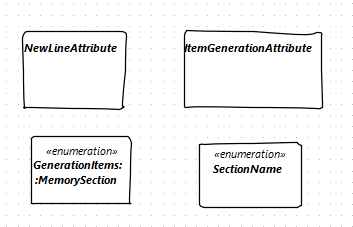
Data structures in Header and Source file are processed in class BaseGenerationItem, some common information such as name of generated definition macro, generated value of macro, pre-comment, post-comment, start qac and end qac.

Each generated items is expressed with types such as: struct, definition macro, C comment, raw text. All types are managed by classes such as DefineGenerationItem, StructMemberGenerationItem,… Some common informations are presented on base class as BaseGenerationItem.



The MCAL module need creating instances to generate coresponding output. Table below describe each output types that supported by Tool.

|  |  |
| --- | --- |
| Class name | Output |
| Instance of class CommentGenerationItem | Generate C comment. Ex. /\* … \*/ |
| Instance of class BreakLineGenerationItem | Define number of new lines each output row |
| Instance of class DefineGenerationItem | Generate definition macro values |
| Instance of class StringGenerationItem | Generate any text |
| Instance of class StructMemberGenerationItem | Generate member of struct in C. Ex:  CONST(Lin\_ChannelInfo, LIN\_CONFIG\_DATA) Lin\_GaaChannelConfig[] =  {  /\* member 1 \*/  0x00U,  /\* member 2 \*/  0x00U,  /\* member 3\*/  LIN\_TRUE,  } |
| Instance of class StructGenerationItem | Generate a struct in C. Ex:  CONST(Lin\_ChannelInfo, LIN\_CONFIG\_DATA) Lin\_GaaChannelConfig[] =  {  } |



Some classes play the Attribute role to mark method such as a new line will be inserted in file output. In addition, Enum data types represent a combination of significative choices as section name in file output.

Ex.

NewLineAttribute: mark for adding new line that follow generated definition marco

ItemGenerateAttribute: mark for whether definition macro is generated or not.

SectionName data type: define enum members for Section choices such as Global Data Types, Global Data, Function Definitions, Version Information, Common Published Information.

MemorySection data type: define members for memory section choices such as start section, stop section,….

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

|  |  |
| --- | --- |
| **Location File Name** | FileGeneration.cs |
| **Description** | FileGeneration an abstract class for other classes used to generate .h and .c files.  It also implements GenerateFile method of interface IFileGeneration. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | FileName | Public | Contain output file name |
| (2) | OutputDirectory | Public | Contain output directory to generate files |
| (3) | Description | Public | Contain description information |
| (4) | Sections | Public | Contain sections information in a generated file. |
| (5) | BasicConfiguration | Protected | Contain basic configuration information such as module id, ModuleName, ... |
| (6) | RuntimeConfiguration | Protected | Contain runtime configuration information such as cdf files, translation file path,... |
| (7) | IntermediateData | Protected | Contain intermediate data that store calculated value in intermediate class. |
| (8) | UserInterface | Private | Contain user interface that is used to report error/info/warning to user |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetAttrSection | Public | Get attribute section |
| (2) | GetAttrSectionOrder | Public | Get order attribute of method that decide order running of methods |
| (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 with arguments |
| (6) | GenCOPYRIGHTSection | Protected | Generate copyright section |
| (7) | GenIncludeSection | Protected | Generate include section |
| (8) | GenVersionInformationSection | Protected | Generate version information section |
| (9) | GenEND\_OF\_FILE | Protected | Generate end of file section |
| (10) | Gen\_GentoolVersion | Protected | Generate tool version section |
| (11) | Gen\_InputFiles | Protected | Generate input files' paths |

#### Fields

##### FileName

GENERIC\_GDD\_CLS\_004\_001:

|  |  |
| --- | --- |
| **Variable Name** | FileName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain output file name |

##### OutputDirectory

GENERIC\_GDD\_CLS\_004\_002:

|  |  |
| --- | --- |
| **Variable Name** | OutputDirectory |
| **Type** | string |
| **Range** | None |
| **Description** | Contain output directory to generate files |

##### Description

GENERIC\_GDD\_CLS\_004\_003:

|  |  |
| --- | --- |
| **Variable Name** | Description |
| **Type** | string |
| **Range** | None |
| **Description** | Contain description information |

##### Sections

GENERIC\_GDD\_CLS\_004\_004:

|  |  |
| --- | --- |
| **Variable Name** | Sections |
| **Type** | Section |
| **Range** | None |
| **Description** | Contain sections information in a generated file. |

##### BasicConfiguration

GENERIC\_GDD\_CLS\_004\_005:

|  |  |
| --- | --- |
| **Variable Name** | BasicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | Contain basic configuration information such as module id, ModuleName, ... |

##### RuntimeConfiguration

GENERIC\_GDD\_CLS\_004\_006:

|  |  |
| --- | --- |
| **Variable Name** | RuntimeConfiguration |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | Contain runtime configuration information such as cdf files, translation file path,... |

##### IntermediateData

GENERIC\_GDD\_CLS\_004\_007:

|  |  |
| --- | --- |
| **Variable Name** | IntermediateData |
| **Type** | IIntermediateData |
| **Range** | None |
| **Description** | Contain intermediate data that store calculated value in intermediate class. |

##### UserInterface

GENERIC\_GDD\_CLS\_004\_008:

|  |  |
| --- | --- |
| **Variable Name** | UserInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | Contain user interface that is used to report error/info/warning to user |

#### Methods

##### GenerateFile

GENERIC\_GDD\_CLS\_004\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenerateFile | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate an output file. It can be .h or .c file.  Algorithm: Scan all non-public methods which have ItemGeneration attribute and sort them by Section  FOREACH each method  IF method has NewLine attribute THEN  Generate new lines based on value of NumberOfNewLines  END  Calcuate each method to write result to output files  END   Handle error ERR000019 if can NOT write output files | | |

{Ref: [1] CMN\_GFD\_ERR\_019, GENERIC\_TFD\_X2x\_GT\_ERR\_019}

##### GetAttrSection

GENERIC\_GDD\_CLS\_004\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAttrSection | | |
| **Arguments** | MethodInfo method | None | |
| Range: | None |
| **Return** | return SectionName | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get attribute section  Algorithm: Get attribute depend on Method of classes  return generated sections | | |

{Ref: N/A}

##### GetAttrSectionOrder

GENERIC\_GDD\_CLS\_004\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAttrSectionOrder | | |
| **Arguments** | MethodInfo method | None | |
| Range: | None |
| **Return** | return float | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get order attribute of method that decide order running of methods  Algorithm: If order attribute of method is not null, return value of order attribute. | | |

{Ref: N/A}

##### WriteToFile

GENERIC\_GDD\_CLS\_004\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | WriteToFile | | |
| **Arguments** | string buffer | None | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Write to file  Algorithm: IF runtimeConfiguration.DryRun = true THEN  Do nothing  End   Scan all non-public methods which have ItemGeneration attribute and sort them by Section  FOR each method  IF method has NewLine attribute THEN  Generate new lines based on value of NumberOfNewLines  END  Calcuate each method to write result to output files  END   Handle error ERR000019 if can NOT write output files | | |

{Ref: [1] CMN\_GFD\_ERR\_019}

##### FileGeneration

GENERIC\_GDD\_CLS\_004\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | FileGeneration | | |
| **Arguments** | string fileName | File name | |
| Range: | None |
| string outputDirectory | Output directory | |
| Range: | None |
| basicConfiguration | Basic configuration | |
| Range: | None |
| runtimeConfiguration | Runtime configuration | |
| Range: | None |
| intermediateData | Intermediate Data | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | File generation constructor with arguments  Algorithm: Init values such as file name, output directory, basic configuration,  runtime configuration, intermediate data | | |

{Ref: GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_007, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_008}

##### GenCOPYRIGHTSection

GENERIC\_GDD\_CLS\_004\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenCOPYRIGHTSection | | |
| **Arguments** | None | | |
| **Return** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate copyright section | | |

{Ref: N/A}

##### GenEND\_OF\_FILE

GENERIC\_GDD\_CLS\_004\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenEND\_OF\_FILE | | |
| **Arguments** | None | | |
| **Return** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate end of file section  Algorithm: Generate end of file section. An abstract method overrided by specific module  to provide end of file information | | |

{Ref: N/A}

##### GenIncludeSection

GENERIC\_GDD\_CLS\_004\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenIncludeSection | | |
| **Arguments** | None | | |
| **Return** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate include section  Algorithm: Generate include section. An abstract method overrided by specific module  to provide include information | | |

{Ref: N/A}

##### GenVersionInformationSection

GENERIC\_GDD\_CLS\_004\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenVersionInformationSection | | |
| **Arguments** | None | | |
| **Return** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate version information section  Algorithm: Generate version information. An abstract method overrided by specific module  to provide version information | | |

{Ref: N/A}

##### Gen\_GentoolVersion

GENERIC\_GDD\_CLS\_004\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Gen\_GentoolVersion | | |
| **Arguments** | None | | |
| **Return** | RETURN a BaseGenerationItem | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate tool version section  Algorithm: Generate tool version from template | | |

{Ref: N/A}

##### Gen\_InputFiles

GENERIC\_GDD\_CLS\_004\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Gen\_InputFiles | | |
| **Arguments** | None | | |
| **Return** | RETURN a BaseGenerationItem | None | |
| Range: | None |
| **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** | Generation class that look up all classes used to generate .h and .c files and call their GenerateFile method. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | Generation instance used by ObjectFactory to get a new Generation object |
| (2) | basicConfiguration | Private | Contain basic configuration information such as module name, module id,... |
| (3) | runtimeConfiguration | Private | Contain runtime configuration information such as cdf files, SrcOutputDirectory,... |
| (4) | intermediateData | Private | Contain intermediate data is used to get calculated values by |

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** | Generation instance used by ObjectFactory to get a new Generation object |

##### basicConfiguration

GENERIC\_GDD\_CLS\_005\_002:

|  |  |
| --- | --- |
| **Variable Name** | basicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | Contain basic configuration information such as module name, module id,... |

##### runtimeConfiguration

GENERIC\_GDD\_CLS\_005\_003:

|  |  |
| --- | --- |
| **Variable Name** | runtimeConfiguration |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | Contain runtime configuration information such as cdf files, SrcOutputDirectory,... |

##### intermediateData

GENERIC\_GDD\_CLS\_005\_004:

|  |  |
| --- | --- |
| **Variable Name** | intermediateData |
| **Type** | IIntermediateData |
| **Range** | None |
| **Description** | Contain intermediate data is used to get calculated values by |

#### Methods

##### GenerateOuput

GENERIC\_GDD\_CLS\_005\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenerateOuput | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate all output files  Algorithm: Get all instances having interface IFileGeneration  For each instance  Generate a file  End | | |

{Ref: N/A}

##### Generation

GENERIC\_GDD\_CLS\_005\_006:

|  |  |
| --- | --- |
| **Method Name** | Generation |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Generation constructor without arguments |

{Ref: N/A}

##### Generation

GENERIC\_GDD\_CLS\_005\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generation | | |
| **Arguments** | basicConfiguration | Basic configuration | |
| Range: | None |
| runtimeConfiguration | Runtime configuration | |
| Range: | None |
| intermediateData | Intermediate Data | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Generation constructor method with arguments  Algorithm: Init values such as basic configuration, runtime configuration, intermediate data | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | BreakLineGenerationItem.cs |
| **Description** | BreakLineGenerationItem class that defines one or more new lines will be output |

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 with argument numberOfLines |
| (2) | Generate | Public | Generate output string of break lines |

#### Fields

##### numberOfLines

GENERIC\_GDD\_CLS\_006\_001:

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

#### Methods

##### BreakLineGenerationItem

GENERIC\_GDD\_CLS\_006\_002:

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

{Ref: N/A}

##### Generate

GENERIC\_GDD\_CLS\_006\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate output string of break lines  Algorithm: Generate output string of a number of break lines which is specified by numberOfLines | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | StructMemberGenerationItem.cs |
| **Description** | StructMemberGenerationItem class that defines a field of struct will be output. |

1. Field List

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

1. Method List

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

#### Fields

#### Methods

##### Generate

GENERIC\_GDD\_CLS\_007\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate output string of a struct item  Algorithm: Generation string include  1. Precomment  2. Struct fields (all fields have comma (,) at the end, excepted last field)  3. Post comment  Break into multiple lines if length of (1) to (3) is more than MAX\_LINE\_LENGH | | |

{Ref: N/A}

##### StructMemberGenerationItem

GENERIC\_GDD\_CLS\_007\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StructMemberGenerationItem | | |
| **Arguments** | string preComment | Pre comment | |
| Range: | None |
| string postComment | Post comment | |
| Range: | None |
| string value | Value of struct element | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Struct member generation item constructor with arguments | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | BaseGenerationItem.cs |
| **Description** | BaseGenerationItem a base class for other generation item classes. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | PreComment | Public | Comment in front of generated define directive or struct |
| (2) | PostComment | Public | Comment behind generated define directive or struct |
| (3) | Name | Public | Name of definition macro or struct name |
| (4) | Type | Public | Type of a struct |
| (5) | Value | Public | Value of definition macro |
| (6) | Level | Public | Level of inner struct |
| (7) | MAX\_LINE\_LENGH | Protected | Maxinum line lengh when writting output files .h and .c |
| (8) | DEFINE\_INDENT | Protected | Define indent of each define directive when output |
| (9) | DEFINE\_SPACE | Protected | Define space of define directive and value |
| (10) | PARENT | Protected | None |
| (11) | isLast | Protected | Marked whether is last element in a struct |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | BaseGenerationItem | Public | Base generation item constructor with arguments |
| (2) | SetLevel | Public | Set level of inner struct |
| (3) | SetLastItem | Public | Set if last element in a struct |
| (4) | Generate | Public | Generate output string of a generation item An abstract method to be override by subclasses, e.g. |
| (5) | Generate | Public | Generate output strings of an array of generation items |
| (6) | WrapText | Protected | Wrap long text to meets MAX\_LINE\_LENGH |
| (7) | AddComment | Protected | Add pre-comment and post-comment |

#### Fields

##### PreComment

GENERIC\_GDD\_CLS\_008\_001:

|  |  |
| --- | --- |
| **Variable Name** | PreComment |
| **Type** | string |
| **Range** | None |
| **Description** | Comment in front of generated define directive or struct |

##### PostComment

GENERIC\_GDD\_CLS\_008\_002:

|  |  |
| --- | --- |
| **Variable Name** | PostComment |
| **Type** | string |
| **Range** | None |
| **Description** | Comment behind generated define directive or struct |

##### Name

GENERIC\_GDD\_CLS\_008\_003:

|  |  |
| --- | --- |
| **Variable Name** | Name |
| **Type** | string |
| **Range** | None |
| **Description** | Name of definition macro or struct name |

##### Type

GENERIC\_GDD\_CLS\_008\_004:

|  |  |
| --- | --- |
| **Variable Name** | Type |
| **Type** | string |
| **Range** | None |
| **Description** | Type of a struct |

##### Value

GENERIC\_GDD\_CLS\_008\_005:

|  |  |
| --- | --- |
| **Variable Name** | Value |
| **Type** | string |
| **Range** | None |
| **Description** | Value of definition macro |

##### Level

GENERIC\_GDD\_CLS\_008\_006:

|  |  |
| --- | --- |
| **Variable Name** | Level |
| **Type** | int |
| **Range** | None |
| **Description** | Level of inner struct |

##### MAX\_LINE\_LENGH

GENERIC\_GDD\_CLS\_008\_007:

|  |  |
| --- | --- |
| **Variable Name** | MAX\_LINE\_LENGH |
| **Type** | readonly int |
| **Range** | None |
| **Description** | Maxinum line lengh when writting output files .h and .c |

##### DEFINE\_INDENT

GENERIC\_GDD\_CLS\_008\_008:

|  |  |
| --- | --- |
| **Variable Name** | DEFINE\_INDENT |
| **Type** | readonly int |
| **Range** | None |
| **Description** | Define indent of each define directive when output |

##### DEFINE\_SPACE

GENERIC\_GDD\_CLS\_008\_009:

|  |  |
| --- | --- |
| **Variable Name** | DEFINE\_SPACE |
| **Type** | readonly int |
| **Range** | None |
| **Description** | Define space of define directive and value |

##### PARENT

GENERIC\_GDD\_CLS\_008\_010:

|  |  |
| --- | --- |
| **Variable Name** | PARENT |
| **Type** | const int |
| **Range** | None |
| **Description** | None |

##### isLast

GENERIC\_GDD\_CLS\_008\_011:

|  |  |
| --- | --- |
| **Variable Name** | isLast |
| **Type** | bool |
| **Range** | None |
| **Description** | Marked whether is last element in a struct |

#### Methods

##### BaseGenerationItem

GENERIC\_GDD\_CLS\_008\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseGenerationItem | | |
| **Arguments** | string preComment | Pre comment | |
| Range: | None |
| string postComment | Post comment | |
| Range: | None |
| string name | Name of define statement or struct | |
| Range: | None |
| string type | Type of a struct | |
| Range: | None |
| string value | Value of define statement | |
| Range: | None |
| int level | Level of inner struct | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base generation item constructor with arguments  Algorithm: Init values of PreComment, PostComment, Name, Type, Level, Value | | |

{Ref: N/A}

##### Generate

GENERIC\_GDD\_CLS\_008\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate output string of a generation item An abstract method to be override by subclasses, e.g.  Algorithm: Override by specific module. | | |

{Ref: N/A}

##### Generate

GENERIC\_GDD\_CLS\_008\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | items | None | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate output strings of an array of generation items  Algorithm: Loop all generation items and concatenate generated strings by new line characters | | |

{Ref: N/A}

##### SetLastItem

GENERIC\_GDD\_CLS\_008\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SetLastItem | | |
| **Arguments** | bool value | Is last element in a struct | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Set if last element in a struct  Algorithm: Marked last item in structs | | |

{Ref: N/A}

##### SetLevel

GENERIC\_GDD\_CLS\_008\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SetLevel | | |
| **Arguments** | int level | Level of inner struct | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Set level of inner struct  Algorithm: Update level of object | | |

{Ref: N/A}

##### AddComment

GENERIC\_GDD\_CLS\_008\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AddComment | | |
| **Arguments** | StringBuilder outStr | outStr | |
| Range: | None |
| string indentStr | indentStr | |
| Range: | None |
| string value | value | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Add pre-comment and post-comment  Algorithm: Add pre-comment or post-comment to definition macro or struct members. | | |

{Ref: N/A}

##### WrapText

GENERIC\_GDD\_CLS\_008\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | WrapText | | |
| **Arguments** | string inputString | Input string | |
| Range: | None |
| string prefix | Prefix string | |
| Range: | None |
| string suffix | Suffix string | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Wrap long text to meets MAX\_LINE\_LENGH  Algorithm: 1. Split inputString into words by using separator (space bar, tab, new line)  2. Contate words into lines (including prefix and suffix) with length less than MAX\_LINE\_LENGH | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | CommentGenerationItem.cs |
| **Description** | Comment generation item constructor with argument text |

1. Field List

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

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | CommentGenerationItem | Public | Comment generation item constructor with argument text |
| (2) | Generate | Public | Generate output string of a comment |

#### Fields

#### Methods

##### CommentGenerationItem

GENERIC\_GDD\_CLS\_009\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CommentGenerationItem | | |
| **Arguments** | string text | Text | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Comment generation item constructor with argument text  Algorithm: Get value from comment by remove /\* and \*/ | | |

{Ref: N/A}

##### Generate

GENERIC\_GDD\_CLS\_009\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate output string of a comment  Algorithm: Generate output string of a comment by append "/\*" and "\*/" to text  Break into multiple lines if length of the comment is more than MAX\_LINE\_LENGH | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | DefineGenerationItem.cs |
| **Description** | DefineGenerationItem class that defines a macro definition will be output to .h and.c files. |

1. Field List

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

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | DefineGenerationItem | Public | Define generation item constructor with arguments |
| (2) | Generate | Public | Generate output string of a define macro |
| (3) | generateComment | Private | Create comment for definition macro or struct members |
| (4) | computeSpaces | Private | Macro definition has format: #define name {spaces} value This method calculate {spaces} length to make length of macro definition string is MAX\_LINE\_LENGH Mininum {spaces} length is 1 If length of macro definition string is more than MAX\_LINE\_LENGH Break into multiple lines End |

#### Fields

#### Methods

##### DefineGenerationItem

GENERIC\_GDD\_CLS\_010\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | DefineGenerationItem | | |
| **Arguments** | string preComment | Pre comment | |
| Range: | None |
| string postComment | Post comment | |
| Range: | None |
| string name | Name of define statement or struct | |
| Range: | None |
| string value | Value of define statement | |
| Range: | None |
| int level | Level of inner struct | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Define generation item constructor with arguments | | |

{Ref: N/A}

##### Generate

GENERIC\_GDD\_CLS\_010\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate output string of a define macro  Algorithm: Generate output string of a define macro with format:  1. Pre-comment  2. #define name {spaces} value  3. Post-comment  For (1) and (3)  - Break into multiple lines if length of (1) or (3) is more than MAX\_LINE\_LENGH  For (2)  - There are {spaces} between name and value to make length of (2) is MAX\_LINE\_LENGH  - Min of {spaces} is 1.  - Break into multiple lines if length of (2) is more than MAX\_LINE\_LENGH | | |

{Ref: N/A}

##### computeSpaces

GENERIC\_GDD\_CLS\_010\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | computeSpaces | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Macro definition has format: #define name {spaces} value This method calculate {spaces} length to make length of macro definition string is MAX\_LINE\_LENGH Mininum {spaces} length is 1 If length of macro definition string is more than MAX\_LINE\_LENGH Break into multiple lines End  Algorithm: Calculate number of spaces and get new line if name or value of macro are over max line length | | |

{Ref: N/A}

##### generateComment

GENERIC\_GDD\_CLS\_010\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | generateComment | | |
| **Arguments** | StringBuilder outStr | A string buffer | |
| Range: | None |
| string value | Comment | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Create comment for definition macro or struct members  Algorithm: Add comment to string buffer if value is not null or empty | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | StringGenerationItem.cs |
| **Description** | StringGenerationItem class that defines a string will be output to .h and.c files. |

1. Field List

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

1. Method List

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

#### Fields

#### Methods

##### Generate

GENERIC\_GDD\_CLS\_011\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate output string from Value attribute  Algorithm: Generate a value string | | |

{Ref: N/A}

##### StringGenerationItem

GENERIC\_GDD\_CLS\_011\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StringGenerationItem | | |
| **Arguments** | string value | Value of define statement | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | String generation item constructor with argument value  Algorithm: Init values | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | StructGenerationItem.cs |
| **Description** | StructGenerationItem class that defines a struct will be output to .c files. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | children | Private | Inner structs or fields |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | StructGenerationItem | Public | Struct generation item constructor with arguments |
| (2) | AddChild | Public | Add an inner struct or a field |
| (3) | Generate | Public | Generate output string of a struct |

#### Fields

##### children

GENERIC\_GDD\_CLS\_012\_001:

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

#### Methods

##### AddChild

GENERIC\_GDD\_CLS\_012\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AddChild | | |
| **Arguments** | child | Child item | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Add an inner struct or a field  Algorithm: Add a child struct or field to parent struct or struct | | |

{Ref: N/A}

##### Generate

GENERIC\_GDD\_CLS\_012\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate output string of a struct  Algorithm: Generation string include  1. Precomment  2. Struct type  3. Inner structs or fields (all items have comma (,) at the end, excepted last item)  4. Post comment  Break into multiple lines if length of (1) to (4) is more than MAX\_LINE\_LENGH | | |

{Ref: N/A}

##### StructGenerationItem

GENERIC\_GDD\_CLS\_012\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StructGenerationItem | | |
| **Arguments** | string preComment | Pre comment | |
| Range: | None |
| string postComment | Post comment | |
| Range: | None |
| string name | Name of define statement or struct | |
| Range: | None |
| string type | Type of a struct | |
| Range: | None |
| int level | Value of define statement | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Struct generation item constructor with arguments | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | HeaderFileGeneration.cs |
| **Description** | HeaderFileGeneration a abstract class for classes used to generate .h file. |

1. Field List

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

1. Method List

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

#### Fields

#### Methods

##### GenEND\_MACRO

GENERIC\_GDD\_CLS\_013\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenEND\_MACRO | | |
| **Arguments** | None | | |
| **Return** | RETURN a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate end macro section  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** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate function prototype section  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** | RETURN a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate global data section  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** | RETURN a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate global data types section  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** | RETURN a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate global symbols section  Algorithm: Generate global symbols. An abstract method overrided by specific module  to provide global symbols information | | |

{Ref: N/A}

##### GenSTART\_MACRO

GENERIC\_GDD\_CLS\_013\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenSTART\_MACRO | | |
| **Arguments** | None | | |
| **Return** | RETURN a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate start macro section  Algorithm: Define template start macro  Generate start macro from template | | |

{Ref: N/A}

##### HeaderFileGeneration

GENERIC\_GDD\_CLS\_013\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | HeaderFileGeneration | | |
| **Arguments** | string fileName | File name | |
| Range: | None |
| string outputDirectory | Output directory | |
| Range: | None |
| basicConfiguration | Basic configuration | |
| Range: | None |
| runtimeConfiguration | Runtime configuration | |
| Range: | None |
| intermediateData | Intermediate Data | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Header file generation constructor with argumnets  Algorithm: Init values of fileName, outputDirectory, basicConfiguration, runtimeConfiguration, intermediateData | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | SectionGenAttribute.cs |
| **Description** | ItemGenerationAttribute class that groups methods of generation classes into sections. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Section | Public | None |
| (2) | SectionOrder | Public | Order of section in file .c and .h |
| (3) | SectionString | Private | None |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetSectionName | Public | Get section name |
| (2) | CompareTo | Private | Compare 2 ItemGeneration attribute |

#### Fields

##### Section

GENERIC\_GDD\_CLS\_014\_001:

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

##### SectionOrder

GENERIC\_GDD\_CLS\_014\_002:

|  |  |
| --- | --- |
| **Variable Name** | SectionOrder |
| **Type** | float |
| **Range** | None |
| **Description** | Order of section in file .c and .h |

##### SectionString

GENERIC\_GDD\_CLS\_014\_003:

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

#### Methods

##### GetSectionName

GENERIC\_GDD\_CLS\_014\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetSectionName | | |
| **Arguments** | Section | None | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get section name  Algorithm: Get text of section based on index (enum) | | |

{Ref: N/A}

##### CompareTo

GENERIC\_GDD\_CLS\_014\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CompareTo | | |
| **Arguments** | target | Target attribute | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Compare 2 ItemGeneration attribute  Algorithm: Compare 2 ItemGeneration attributes by section name and section order  - First compare section names  If this.Section larger than target.Section Then Return positive number  If this.Section less than target.Section Then Return negative number  - Second compare section order, if 2 section names are equal  If this.SectionOrder larger than target.SectionOrder Then Return positive number  If this.SectionOrder less than target.SectionOrder Then Return negative number  If this.SectionOrder equal to target.SectionOrder Then Return 0 | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | NewLineAttribute.cs |
| **Description** | NewLineAttribute class used for generation methods which are decorated by ItemGenerationAttribute to add one more new lines before output content of the methods. |

1. Field List

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

1. Method List

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

#### Fields

##### NumberOfNewLines

GENERIC\_GDD\_CLS\_015\_001:

|  |  |
| --- | --- |
| **Variable Name** | NumberOfNewLines |
| **Type** | int |
| **Range** | None |
| **Description** | Contain information number of new lines |

#### Methods

##### NewLineAttribute

GENERIC\_GDD\_CLS\_015\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | NewLineAttribute | | |
| **Arguments** | int numberOfNewLines | 1 | |
| Range: | Number of new lines |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | New line attribute constructor with argument numberOfNewLines  Algorithm: Init value for NumberOfNewLines attribute | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | Section.cs |
| **Description** | Section class that is used to generate section template. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | title | Private | Contain title information of section |
| (2) | body | Private | Contain body information of section |

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** | Contain title information of section |

##### body

GENERIC\_GDD\_CLS\_016\_002:

|  |  |
| --- | --- |
| **Variable Name** | body |
| **Type** | string |
| **Range** | None |
| **Description** | Contain body information of section |

#### Methods

##### Generate

GENERIC\_GDD\_CLS\_016\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Generate | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate section information  Algorithm: Generate section information including title and body with length does not exceed 80 character | | |

{Ref: N/A}

##### Section

GENERIC\_GDD\_CLS\_016\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Section | | |
| **Arguments** | string title | Title | |
| Range: | None |
| string body | Body | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Algorithm: Init values of title, body | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | SourceFileGeneration.cs |
| **Description** | SourceFileGeneration an abstract class for classes used to generate .c file. |

1. Field List

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

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | SourceFileGeneration | Protected | Source file generation constructor with arguments |
| (2) | GenVERSION\_CHECK | Protected | Generate version checking setion If have any |
| (3) | GenGLOBAL\_DATA\_TYPE | Protected | Generate global data type section if have any |
| (4) | GenGLOBAL\_DATA | Protected | Generate global data section if have any |
| (5) | GenFUNTION\_DEFINE | Protected | Generate function definitions section if have any |

#### Fields

#### Methods

##### GenFUNTION\_DEFINE

GENERIC\_GDD\_CLS\_017\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenFUNTION\_DEFINE | | |
| **Arguments** | None | | |
| **Return** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate function definitions section if have any  Algorithm: Generate function definitions. An abstract method overrided by specific module  to provide function definitions information | | |

{Ref: N/A}

##### GenGLOBAL\_DATA

GENERIC\_GDD\_CLS\_017\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenGLOBAL\_DATA | | |
| **Arguments** | None | | |
| **Return** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate global data section if have any  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\_017\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenGLOBAL\_DATA\_TYPE | | |
| **Arguments** | None | | |
| **Return** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate global data type section if have any  Algorithm: Generate global data type. An abstract method overrided by specific module  to provide global data type information | | |

{Ref: N/A}

##### GenVERSION\_CHECK

GENERIC\_GDD\_CLS\_017\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GenVERSION\_CHECK | | |
| **Arguments** | None | | |
| **Return** | return a BaseGenerationItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Generate version checking setion If have any  Algorithm: Generate version check. An abstract method overrided by specific module  to provide version checking information | | |

{Ref: N/A}

##### SourceFileGeneration

GENERIC\_GDD\_CLS\_017\_006:

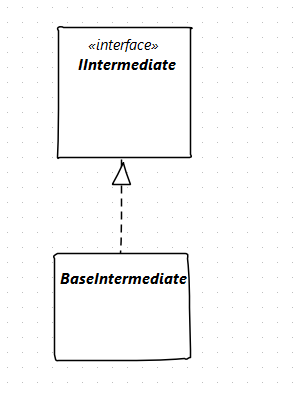
|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SourceFileGeneration | | |
| **Arguments** | string fileName | File name | |
| Range: | None |
| string outputDirectory | Output directory | |
| Range: | None |
| basicConfiguration | Basic configuration | |
| Range: | None |
| runtimeConfiguration | Runtime configuration | |
| Range: | None |
| intermediateData | Intermediate Data | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Source file generation constructor with arguments | | |

{Ref: N/A}

## Intermediate

This component creates intermediate data structures, which will contain information required to generate Generation Tool output files. The data will be calculated by the MCAL modules (MSN) based on their algorithm and will be managed by intermediate classes. The orthers component such as Validation or Generation on the MSN side use those data to validate/generate to output files.

### UML



BaseIntermediate class implement a mechanism to invoke all calculation algorithms on the MSN side and store computed values into database. See more detail at method ComputeAll()

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

|  |  |
| --- | --- |
| **Location File Name** | BaseIntermediate.cs |
| **Description** | BaseIntermediate an abstract class for classes used to calculate intermediate data. It implements ComputeAll method of IIntermediate interface to look up all methods decorated by IntermediateItemAttribute and invoke them to calculate intermediate data. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Repo | Protected | Contain object repository used to get data from CDF, BSW, Translation files |
| (2) | Interdata | Protected | Contain object intermediate data used to get calculated value from |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ComputeAll | Public | Compute all intermediate data by calling all intermediation methods |
| (2) | BaseIntermediate | Protected | Base intermedia constructor without arguments |
| (3) | BaseIntermediate | Protected | Base intermedia constructor with arguments |
| (4) | Init | Protected | Init some values based on override by Specific module. |

#### Fields

##### Repo

GENERIC\_GDD\_CLS\_018\_001:

|  |  |
| --- | --- |
| **Variable Name** | Repo |
| **Type** | IRepository |
| **Range** | None |
| **Description** | Contain object repository used to get data from CDF, BSW, Translation files |

##### Interdata

GENERIC\_GDD\_CLS\_018\_002:

|  |  |
| --- | --- |
| **Variable Name** | Interdata |
| **Type** | IIntermediateData |
| **Range** | None |
| **Description** | Contain object intermediate data used to get calculated value from |

#### Methods

##### ComputeAll

GENERIC\_GDD\_CLS\_018\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ComputeAll | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Compute all intermediate data by calling all intermediation methods  Algorithm: FOREACH all non-public methods that have IntermediateItem attribute and order by line number position  CALL intermediate methods to get result  Store result in intermediate data with DBPath retrieving from IntermediateItem attribute  End | | |

{Ref: N/A}

##### BaseIntermediate

GENERIC\_GDD\_CLS\_018\_004:

|  |  |
| --- | --- |
| **Method Name** | BaseIntermediate |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Base intermedia constructor without arguments |

{Ref: N/A}

##### BaseIntermediate

GENERIC\_GDD\_CLS\_018\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseIntermediate | | |
| **Arguments** | repo | Repository | |
| Range: | None |
| interdata | Intermediate Data | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base intermedia constructor with arguments  Algorithm: Init values of Repo, Interdata | | |

{Ref: N/A}

##### Init

GENERIC\_GDD\_CLS\_018\_006:

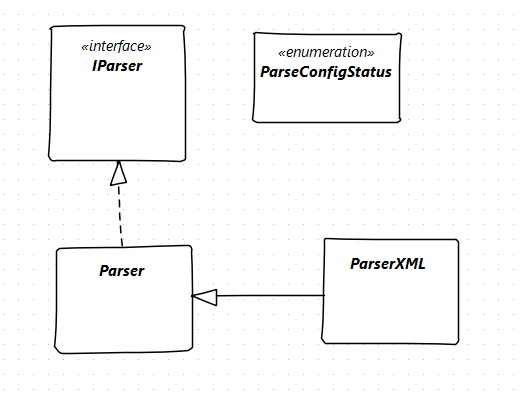
|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Init | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Init some values based on override by Specific module. | | |

{Ref: N/A}

## Parser

This component bear the responsibility for parsing input files such as cdf, bswmdt, configuration xml, translation files. It also handle errors relate to user’s input like does not find files, invalid file name, invalid format,…..

### UML



Parser component abide by template method design pattern, it process input files in an order steps as follow:

* Parsing CDF file
* Parsing translation file
* Checking dependency modules
* Storing data into database
* Updating necessary information

IParser interface declare a method to parsing input files

Parser class declare template methods that is overrided by concrete class.

ParserXML class is a concrete class that implement IParser, it play parsing file with XML format role.

ParseConfigStatus data type defines some status used in parsing process.

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

|  |  |
| --- | --- |
| **Location File Name** | ParserXML.cs |
| **Description** | ParserXML class that parses CDF, BSW, translation and device files |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | ParserXML instance used by ObjectFactory to get a new ParserXML object |
| (2) | cdfData | Protected | Contain CDF data as a database |
| (3) | runtimeConfiguration | Protected | Contain runtime configuration information such as cdf file path, src folder path, ... |
| (4) | basicConfiguration | Protected | Contain basic configuration information such as module name, module id,... |
| (5) | userInterface | Protected | Contain user interface component is used to report error/info/warning to user. |
| (6) | cdfFiles | Protected | Contain data of CDF files is parse by |
| (7) | register | Protected | Contain reference register is used to process register macro and address of register |
| (8) | parseConfigStatus | Protected | Contain status when parse content in CDF files such as Success, No |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | RunAll | Public | Process CDF files and store in database as Cdf |
| (2) | ParserXML | Protected | None |
| (3) | ParserXML | Protected | None |
| (4) | LoadFiles | Protected | Load all CDF files into buffer to process parsing later. |
| (5) | ParseFiles | Protected | Parsing CDF and BSW files and store into database such as CDF data and BSW data |
| (6) | CheckRequiredModules | Protected | Check required modules of specific modules such as CAN, DEM, CANIF, FrIf,... |
| (7) | GetDeviceName | Protected | Get device name that exists in CDF files, in normal case it is in msn\_DeviceName |
| (8) | ParseTranslationFile | Protected | Parsing translation file as .trxml |
| (9) | UpdateBasicConfigurationIfNeeded | Protected | Update basic configuration if needed |
| (10) | ParseAndSaveCDFFile | Protected | Parsing CDF file and store it into cdfData as database |
| (11) | ParseConfiguration | Protected | Parsing configuration in tag SHORT-NAME, DEFINITION-REF, MODULE-DESCRIPTION-REF in CDF file. |
| (12) | ShouldFilterOutConfiguration | Protected | Should filter out configuration that is match with provided vendor by user. |
| (13) | BuildAndSaveContainerTree | Protected | Build a containers tree of a module and save in cdfData as database |
| (14) | BuildSubContainers | Protected | Build sub containers that serving to create a containers tree. |
| (15) | ConvertToContainer | Protected | Convert XElement into a Container struct |
| (16) | CheckAutosarVersion | Protected | Check autosar version as 4.0.3 or 4.2.2 |
| (17) | CheckParamConfig | Protected | Check mandatory parameter in BSWMDT such as module id, vendor id, sw-version |
| (18) | ParseAndSaveBswConfiguration | Protected | Parsing BSWMDT configuration and verify each fields. |
| (19) | CheckERR000025 | Protected | This method check path consistency between CDF and ModuleDescriptionRef tag in BSWMDT If have any inconsistency, report error ERR000025 |
| (20) | CheckRelationBswmdtAndCdf | Protected | This method check relation between BSW and CDF files. It raise error ERR000048 if not any relation between BSW and CDF |
| (21) | CheckERR000049 | Protected | This method check VENDOR-SPECIFIC-MODULE-DEF-REF's value |
| (22) | CheckERR000027 | Protected | This method check inconsistence of AR-RELEASE-VERSION value |
| (23) | ParseSubElement | Protected | Parsing tags in BSWMDT such as SW-VERSION, VENDOR-ID, AR-RELEASE-VERSION,... |
| (24) | CheckDescription | Protected | Parsing tags in BSWMDT such as MODULE-ID,... |
| (25) | ValidateBswConfigurationDetail | Protected | Validate detail of basic software configuration |
| (26) | ValidateBswConfigurationsGeneral | Protected | Validate rules required for all BSW configuration |
| (27) | ParseThruParameters | Protected | Parsing tag PARAMETER-VALUES of container and store in Container as an attribute |
| (28) | ParseThruReferences | Protected | Parsing tag REFERENCES-VALUES of container and store in Container as an attribute |
| (29) | ParseItemValue | Protected | Parsing value from parameters on containers that in CDF files. |
| (30) | ParseTransFileSection | Protected | Parsing translation file section defined by tag TRANSLATION-FILE-PATH |
| (31) | ParseDeviceFileSection | Protected | Parsing device file section defined by tag DEVICE-FILE-PATH |
| (32) | GetRootPath | Protected | Get root path of a CDF/BSW file |
| (33) | IsExistModuleName | Protected | Check whether moduleName existed in cdfData or not |
| (34) | convertValueFromString | Private | Convert value to corresponding data type in CDF file. |
| (35) | convertToBool | Private | Convert value to corresponding bool data type. |
| (36) | convertToLong | Private | Convert value to corresponding long data type. |
| (37) | convertToDouble | Private | Convert value to corresponding double data type. |
| (38) | getRootPaths | Private | Get root paths of a CDF/BSW file |

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_019\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly ParserXML |
| **Range** | None |
| **Description** | ParserXML instance used by ObjectFactory to get a new ParserXML object |

##### cdfData

GENERIC\_GDD\_CLS\_019\_002:

|  |  |
| --- | --- |
| **Variable Name** | cdfData |
| **Type** | ICdfData |
| **Range** | None |
| **Description** | Contain CDF data as a database |

##### runtimeConfiguration

GENERIC\_GDD\_CLS\_019\_003:

|  |  |
| --- | --- |
| **Variable Name** | runtimeConfiguration |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | Contain runtime configuration information such as cdf file path, src folder path, ... |

##### basicConfiguration

GENERIC\_GDD\_CLS\_019\_004:

|  |  |
| --- | --- |
| **Variable Name** | basicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | Contain basic configuration information such as module name, module id,... |

##### userInterface

GENERIC\_GDD\_CLS\_019\_005:

|  |  |
| --- | --- |
| **Variable Name** | userInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | Contain user interface component is used to report error/info/warning to user. |

##### cdfFiles

GENERIC\_GDD\_CLS\_019\_006:

|  |  |
| --- | --- |
| **Variable Name** | cdfFiles |
| **Type** | Dictionary< string, XDocument > |
| **Range** | None |
| **Description** | Contain data of CDF files is parse by |

##### register

GENERIC\_GDD\_CLS\_019\_007:

|  |  |
| --- | --- |
| **Variable Name** | register |
| **Type** | IRegisterProcessing |
| **Range** | None |
| **Description** | Contain reference register is used to process register macro and address of register |

##### parseConfigStatus

GENERIC\_GDD\_CLS\_019\_008:

|  |  |
| --- | --- |
| **Variable Name** | parseConfigStatus |
| **Type** | List< Tuple< string, ParseConfigStatus |
| **Range** | None |
| **Description** | Contain status when parse content in CDF files such as Success, No |

#### Methods

##### RunAll

GENERIC\_GDD\_CLS\_019\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | RunAll | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Process CDF files and store in database as Cdf  Algorithm: 1. Call loadFiles to load XDocuments from CDF file paths  2. Call parseFiles to parse all CDF and BSW files  3. Call checkRequiredModules to check required modules  4. Call parseTranslationFile to parse translation files  5. Call updateBasicConfigurationIfNeeded to update information used for Object Factory | | |

{Ref: }

##### BuildAndSaveContainerTree

GENERIC\_GDD\_CLS\_019\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BuildAndSaveContainerTree | | |
| **Arguments** | XDocument file | CDF file in XDocument | |
| Range: | None |
| XElement module | Module in XElement | |
| Range: | None |
| config | Configuration of module | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Build a containers tree of a module and save in cdfData as database  Algorithm: Get all containers in module by getting all elements inside tag CONTAINERS  FOREACH element in container  Recursively build a container tree where this container is root and all of its children  Add container short name and container tree to cdfData  End | | |

{Ref: N/A}

##### BuildSubContainers

GENERIC\_GDD\_CLS\_019\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BuildSubContainers | | |
| **Arguments** | ref Stack< XElement > stack | ref Stack XElement | |
| Range: | None |
| container | Container | |
| Range: | None |
| XElement sub\_container | XElement | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Build sub containers that serving to create a containers tree.  Algorithm: Get all containers in module by getting all elements inside tag CONTAINERS  FOREACH subelement in container  Recursively build a container tree where this container is root and all of its children  Add container short name and container tree to cdfData  End | | |

{Ref: N/A}

##### CheckAutosarVersion

GENERIC\_GDD\_CLS\_019\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckAutosarVersion | | |
| **Arguments** | bswConfig | BswConfig | |
| Range: | None |
| **Return** | return a Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check autosar version as 4.0.3 or 4.2.2  Algorithm: IF AUTOSAR version of BSWMDT File are not "4.2.2" or "4.0.3"  - Report ERR000058 and exit  IF AUTOSAR version in BSWMDT File of either AUTOSAR R4.2.2 or R4.0.3 are not provided  Report ERR000027  IF if AUTOSAR version of BSWMDT File and Module Description File is different THEN  - Report ERR000029 and exit | | |

{Ref: [1] CMN\_GFD\_ERR\_027, CMN\_GFD\_ERR\_029, CMN\_GFD\_ERR\_058, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_015, GENERIC\_TFD\_X2x\_GT\_ERR\_027, GENERIC\_TFD\_X2x\_GT\_ERR\_029, GENERIC\_TFD\_X2x\_GT\_ERR\_058}

##### CheckDescription

GENERIC\_GDD\_CLS\_019\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckDescription | | |
| **Arguments** | XElement subElement | XElement | |
| Range: | None |
| string[] requiredFields | string[] | |
| Range: | None |
| Dictionary< string, string > tagNameToField | Dictionary | |
| Range: | None |
| ref bswConfig | ref BswConfig | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing tags in BSWMDT such as MODULE-ID,...  Algorithm: Scan tags in tag BSW-MODULE-DESCRIPTION  Parsing sub tags such as MODULE-ID | | |

{Ref: N/A}

##### CheckERR000025

GENERIC\_GDD\_CLS\_019\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckERR000025 | | |
| **Arguments** | XDocument bsw | The content of BSWMDT | |
| Range: | None |
| configs | THe configurations of CDF | |
| Range: | None |
| bswConfig | Information of BSWMDT | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This method check path consistency between CDF and ModuleDescriptionRef tag in BSWMDT If have any inconsistency, report error ERR000025  Algorithm: Read path from root tag to tags ImplementSortName  if path is not match with value of ModuleDescriptionRef in CDF file  report error ERR000025 | | |

{Ref: [1] CMN\_GFD\_ERR\_025, GENERIC\_TFD\_X2x\_GT\_ERR\_025}

##### CheckERR000027

GENERIC\_GDD\_CLS\_019\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckERR000027 | | |
| **Arguments** | string arVersion | Value of autosar version | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This method check inconsistence of AR-RELEASE-VERSION value  Algorithm: Handle error ERR000027 | | |

{Ref: [1] CMN\_GFD\_ERR\_027}

##### CheckERR000049

GENERIC\_GDD\_CLS\_019\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckERR000049 | | |
| **Arguments** | IList< bswConfigs | Information of bsw mdt | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | This method check VENDOR-SPECIFIC-MODULE-DEF-REF's value  Algorithm: Read value of tag VendorSpecificModuleDefRef in bswmdt  If not exists value of tag VendorSpecificModuleDefRef in bswmdt  report error ERR000049 | | |

{Ref: [1] CMN\_GFD\_ERR\_049, GENERIC\_TFD\_X2x\_GT\_ERR\_049}

##### CheckParamConfig

GENERIC\_GDD\_CLS\_019\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckParamConfig | | |
| **Arguments** | bswConfig | BswConfig | |
| Range: | None |
| string[] requiredFields | Required fields | |
| Range: | None |
| Dictionary< string, string > tagNameToField | Mapping from tag names to fields | |
| Range: | None |
| **Return** | return a void | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Check mandatory parameter in BSWMDT such as module id, vendor id, sw-version  Algorithm: IF required field is not configured in BSWMDT File THEN  -Record ERR000030 | | |

{Ref: [1] CMN\_GFD\_ERR\_030, GENERIC\_TFD\_X2x\_GT\_ERR\_030}

##### CheckRelationBswmdtAndCdf

GENERIC\_GDD\_CLS\_019\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckRelationBswmdtAndCdf | | |
| **Arguments** | IList< bswConfigs | 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: if DefinitionRef tag in CDF file is not match with VendorSpecificModuleDefRef tag in BSWMDT  report error ERR000048 | | |

{Ref: [1] CMN\_GFD\_ERR\_048, GENERIC\_TFD\_X2x\_GT\_ERR\_048}

##### CheckRequiredModules

GENERIC\_GDD\_CLS\_019\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckRequiredModules | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check required modules of specific modules such as CAN, DEM, CANIF, FrIf,...  Algorithm: FOREACH module name in basicConfiguration.ModuleRequired  CALL method isExistModuleName to check if the module name is existed in input CDF files  IF NOT existed THEN  Record absented module name  End  End   Record error ERR[MSN]003 for absented module names if found. | | |

{Ref: N/A}

##### ConvertToContainer

GENERIC\_GDD\_CLS\_019\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ConvertToContainer | | |
| **Arguments** | XElement elements | Elements | |
| Range: | None |
| **Return** | return a Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Convert XElement into a Container struct  Algorithm: 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\_019\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetDeviceName | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get device name that exists in CDF files, in normal case it is in msn\_DeviceName  Algorithm: Read all containers  Find parameter named as DeviceName  Read value of parameter and return | | |

{Ref: N/A}

##### GetRootPath

GENERIC\_GDD\_CLS\_019\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetRootPath | | |
| **Arguments** | XDocument xmlDoc | Xml doc | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get root path of a CDF/BSW file  Algorithm: 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}

##### IsExistModuleName

GENERIC\_GDD\_CLS\_019\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsExistModuleName | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether moduleName existed in cdfData or not  Algorithm: Get all modules from cdf file data by calling method cdfData.GetModules()  IF found a module name that matchs pattern {modulename}{number}\* THEN  RETURN true  ELSE  RETURN false | | |

{Ref: N/A}

##### LoadFiles

GENERIC\_GDD\_CLS\_019\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | LoadFiles | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Load all CDF files into buffer to process parsing later.  Algorithm: FOREACH each file path in runtimeConfiguration.CDFFilePaths  Load XDocument from the file path to store in cdfFiles  End   Handle error ERR000012 if can NOT read files or files have invalid XML format | | |

{Ref: [1] CMN\_GFD\_ERR\_012, GENERIC\_TFD\_X2x\_GT\_ERR\_012}

##### ParseAndSaveBswConfiguration

GENERIC\_GDD\_CLS\_019\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseAndSaveBswConfiguration | | |
| **Arguments** | XDocument bsw | Bsw | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing BSWMDT configuration and verify each fields.  Algorithm: Invoke sub methods and report ERR000025,ERR000026, ERR000029 or ERR000030 if any  Save BSWMDT Configuration If have not any error. | | |

{Ref: [1] CMN\_GFD\_ERR\_025, CMN\_GFD\_ERR\_026, CMN\_GFD\_ERR\_029, CMN\_GFD\_ERR\_030}

##### ParseAndSaveCDFFile

GENERIC\_GDD\_CLS\_019\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseAndSaveCDFFile | | |
| **Arguments** | string path | CDF file path | |
| Range: | None |
| XDocument file | CDF file in XDocument | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing CDF file and store it into cdfData as database  Algorithm: FOREACH module found in file (module is wrapped in tag ELEMENTS)  Skip module have name ECUC-VALUE-COLLECTION  Call parseConfiguration then check it's parsing status  Record error ERR000028 when tag MODULE-DESCRIPTION-REF NOT found and exit  Save mapping module short name - CDF file path into cdfData  CALL method buildAndSaveContainerTree to build a container tree of this module and save into cdfData  Save parseConfiguration status.  End | | |

{Ref: [1] CMN\_GFD\_ERR\_028, GENERIC\_TFD\_X2x\_GT\_ERR\_028}

##### ParseConfiguration

GENERIC\_GDD\_CLS\_019\_027:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseConfiguration | | |
| **Arguments** | XElement element | Element | |
| Range: | None |
| ref status | ParseConfigStatus | |
| Range: | None |
| **Return** | return a Configuration | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing configuration in tag SHORT-NAME, DEFINITION-REF, MODULE-DESCRIPTION-REF in CDF file.  Algorithm: IF XElement is not null  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  Else  Set Configuration instance to null   CALL shouldFilterOutConfiguration to check Configuration instance should be filtered out  IF shouldFilterOutConfiguration return false  Set ParseConfigStatus to NoData  Else  Save Configuration instance into cdfData   RETURN Configuration instance | | |

{Ref: N/A}

##### ParseDeviceFileSection

GENERIC\_GDD\_CLS\_019\_028:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseDeviceFileSection | | |
| **Arguments** | string translationFilePath | Translation file path | |
| Range: | None |
| XElement deviceFileSection | Section DEVICE-FILE-PATH | |
| Range: | None |
| string deviceName | Device name | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing device file section defined by tag DEVICE-FILE-PATH  Algorithm: IF deviceFileSection is null THEN  RETURN false  End   Try to get device header file path from deviceName tag which is a child of deviceFileSection  - Report error ERR000016 if there are NO children of deviceFileSection and exit  - Report error ERR000017 if there are NO deviceName tag in deviceFileSection and exit   IF NO error found THEN  Save all define macro in device header file  RETURN true  End | | |

{Ref: [1] CMN\_GFD\_ERR\_016, CMN\_GFD\_ERR\_017, GENERIC\_TFD\_X2x\_GT\_ERR\_016, GENERIC\_TFD\_X2x\_GT\_ERR\_017}

##### ParseFiles

GENERIC\_GDD\_CLS\_019\_029:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseFiles | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing CDF and BSW files and store into database such as CDF data and BSW data  Algorithm: FOR each file in cdfFiles  IF file is CDF file  CALL method parseAndSaveCDFFile to parse all CDF file and save a tree of containers to cdfData  End  End   IF file is BSW and exists more once THEN print ERR000040  End  IF module is exists and no any config is parsed due to vendor filter THEN print error ERR000047  End  After process CDF files, call method parseAndSaveBswConfiguration to parse BSW configuration file  CALL method validateBswConfigurationsGeneral to validate parsed BSW configuration file | | |

{Ref: [1] CMN\_GFD\_ERR\_040, CMN\_GFD\_ERR\_047, GENERIC\_TFD\_X2x\_GT\_ERR\_040, GENERIC\_TFD\_X2x\_GT\_ERR\_047}

##### ParseItemValue

GENERIC\_GDD\_CLS\_019\_030:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseItemValue | | |
| **Arguments** | XElement element | Elements | |
| Range: | None |
| string definitionKey | Definition key contain definition tag | |
| Range: | None |
| string valueKey | Value key contain value tag of parameters | |
| Range: | None |
| **Return** | return a item value that contain definition ref and value of ItemValue | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing value from parameters on containers that in CDF files.  Algorithm: Find element in parameter and check if definition ref and value tag is not null or not empty  Check data type of parameter and parse value corresponding with data type. | | |

{Ref: N/A}

##### ParseSubElement

GENERIC\_GDD\_CLS\_019\_031:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseSubElement | | |
| **Arguments** | XElement element | XElement | |
| Range: | None |
| string[] requiredFields | string[] | |
| Range: | None |
| Dictionary< string, string > tagNameToField | Dictionary | |
| Range: | None |
| ref bswConfig | ref BswConfig | |
| Range: | None |
| ref string name | ref string | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing tags in BSWMDT such as SW-VERSION, VENDOR-ID, AR-RELEASE-VERSION,...  Algorithm: Scan tags in tag BSW-IMPLEMENTATION  Parsing sun tags such as SW-VERSION, VENDOR-ID, AR-RELEASE-VERSION, SHORT-NAME | | |

{Ref: GENERIC\_TFD\_X2x\_GT\_BSW\_001}

##### ParseThruParameters

GENERIC\_GDD\_CLS\_019\_032:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseThruParameters | | |
| **Arguments** | XElement element | Elements | |
| Range: | None |
| **Return** | return a IList ItemValue | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing tag PARAMETER-VALUES of container and store in Container as an attribute  Algorithm: Scan tags in tag PARAMETER-VALUES  Parsing each parameter value include definition ref and value  return list of parameter value | | |

{Ref: N/A}

##### ParseThruReferences

GENERIC\_GDD\_CLS\_019\_033:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseThruReferences | | |
| **Arguments** | XElement element | Elements | |
| Range: | None |
| **Return** | return a IList ItemValue | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing tag REFERENCES-VALUES of container and store in Container as an attribute  Algorithm: Scan tags in tag REFERENCES-VALUES  Parsing each reference value include definition ref and value  return list of reference value | | |

{Ref: N/A}

##### ParseTransFileSection

GENERIC\_GDD\_CLS\_019\_034:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseTransFileSection | | |
| **Arguments** | string translationFilePath | Translation file path | |
| Range: | None |
| XElement transFileSection | Section TRANSLATION-FILE-PATH | |
| Range: | None |
| string deviceName | Device name | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing translation file section defined by tag TRANSLATION-FILE-PATH  Algorithm: IF transFileSection is null THEN  Return false  End   Try to get translation header file path from deviceName tag which is a child of transFileSection  - Report error ERR000014 if there are NO children of transFileSection and exit  - Report error ERR000015 if there are NO deviceName tag in transFileSection and exit   IF NO error found THEN  Save all define macro in translation header file  RETURN true  End | | |

{Ref: [1] CMN\_GFD\_ERR\_014, CMN\_GFD\_ERR\_015, GENERIC\_TFD\_X2x\_GT\_ERR\_014, GENERIC\_TFD\_X2x\_GT\_ERR\_015}

##### ParseTranslationFile

GENERIC\_GDD\_CLS\_019\_035:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseTranslationFile | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing translation file as .trxml  Algorithm: Read translation file path from runtimeConfiguration.TranslationFilePath    Get device name configured in containers of current module (specfied by basicConfiguration.ModuleName)  IF device name is null or empty THEN  Report error ERR000039 and exit  End   IF device name has not support yet then report error ERR000046 THEN  Call method parseTransFileSection to parse TRANSLATION-FILE-PATH section  Call method parseDeviceFileSection to parse DEVICE-FILE-PATH section   IF result of parseTransFileSection or parseDeviceFileSection is FALSE  Report error ERR000013 and exit  End | | |

{Ref: [1] CMN\_GFD\_ERR\_013, CMN\_GFD\_ERR\_039, CMN\_GFD\_ERR\_046, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_011, GENERIC\_TFD\_X2x\_GT\_ERR\_013, GENERIC\_TFD\_X2x\_GT\_ERR\_039, GENERIC\_TFD\_X2x\_GT\_ERR\_046}

##### ParserXML

GENERIC\_GDD\_CLS\_019\_036:

|  |  |
| --- | --- |
| **Method Name** | ParserXML |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Parser XML constructor without arguments |

{Ref: N/A}

##### ParserXML

GENERIC\_GDD\_CLS\_019\_037:

|  |  |  |  |
| --- | --- | --- | --- |
| **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** | Parser XML constructor with arguments  Algorithm: Init values of ICdfData, IRuntimeConfiguration, IBasicConfiguration, IRegisterProcessing, IUserInterface | | |

{Ref: N/A}

##### ShouldFilterOutConfiguration

GENERIC\_GDD\_CLS\_019\_038:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ShouldFilterOutConfiguration | | |
| **Arguments** | config | Configuration | |
| Range: | None |
| List< string > vendors | List of string vendors | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Should filter out configuration that is match with provided vendor by user.  Algorithm: 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  END | | |

{Ref: N/A}

##### UpdateBasicConfigurationIfNeeded

GENERIC\_GDD\_CLS\_019\_039:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | UpdateBasicConfigurationIfNeeded | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Update basic configuration if needed  Algorithm: For each container in cdfData of current module (specified by basicConfiguration.ModuleName)  If found parameter DeviceName and its value is NOT null or empty  Add device name to collection "result"  End  End   If having device names in collection "result"  Set collection "result" to basicConfiguration.DeviceNames  End | | |

{Ref: N/A}

##### ValidateBswConfigurationDetail

GENERIC\_GDD\_CLS\_019\_040:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidateBswConfigurationDetail | | |
| **Arguments** | bswConfig | Bsw configuration | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Validate detail of basic software configuration  Algorithm: IF (1) bswConfig.DescriptionShortName is null or THEN  (2) bswConfig.DescriptionShortName NOT start with basicConfiguration.ModuleName  Report error ERR000026 and exit  End | | |

{Ref: [1] CMN\_GFD\_ERR\_026, GENERIC\_TFD\_X2x\_GT\_ERR\_026}

##### ValidateBswConfigurationsGeneral

GENERIC\_GDD\_CLS\_019\_041:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidateBswConfigurationsGeneral | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Validate rules required for all BSW configuration  Algorithm: Validate following rules  IF NO configuration found THEN  Report error ERR000036 and exit  End | | |

{Ref: [1] CMN\_GFD\_ERR\_036, GENERIC\_TFD\_X2x\_GT\_ERR\_036}

##### convertToBool

GENERIC\_GDD\_CLS\_019\_042:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | convertToBool | | |
| **Arguments** | string value | Value is converted to corresponding bool data type. | |
| Range: | None |
| **Return** | return a object | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Convert value to corresponding bool data type.  Algorithm: Convert string value such as true, false, 1, 0 to corresponding bool data type. | | |

{Ref: N/A}

##### convertToDouble

GENERIC\_GDD\_CLS\_019\_043:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | convertToDouble | | |
| **Arguments** | string value | Value is converted to corresponding double data type. | |
| Range: | None |
| **Return** | return a object | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Convert value to corresponding double data type.  Algorithm: Convert value such as NAN, -INF, INF or double value to corresponding double data type. | | |

{Ref: N/A}

##### convertToLong

GENERIC\_GDD\_CLS\_019\_044:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | convertToLong | | |
| **Arguments** | string value | Value is converted to corresponding long data type. | |
| Range: | None |
| **Return** | return a object | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Convert value to corresponding long data type.  Algorithm: Convert value such as hexa, binary, octal to corresponding long data type | | |

{Ref: N/A}

##### convertValueFromString

GENERIC\_GDD\_CLS\_019\_045:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | convertValueFromString | | |
| **Arguments** | string dataType | None | |
| Range: | None |
| string dataValue | None | |
| Range: | None |
| **Return** | return a object | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Convert value to corresponding data type in CDF file.  Algorithm: Check data type and convert value to corresponding data type. | | |

{Ref: N/A}

##### getRootPaths

GENERIC\_GDD\_CLS\_019\_046:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getRootPaths | | |
| **Arguments** | XDocument xmlDoc | Xml doc | |
| Range: | None |
| **Return** | return a string array | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get root paths of a CDF/BSW file  Algorithm: Get all inner elements of xmlDoc that have name AR-PACKAGE and have inner tag is AR-PACKAGES or ELEMENTS  RETURN root paths by concatenating their short name by "/" separator | | |

{Ref: N/A}

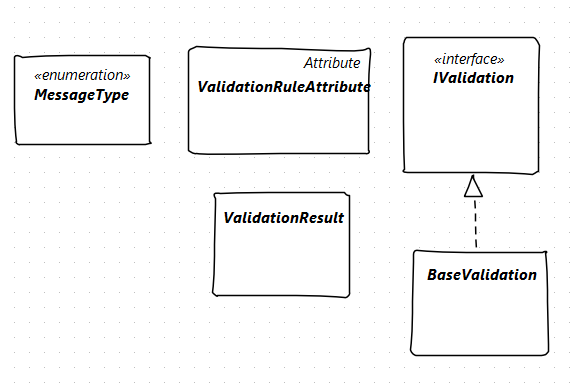
### 

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | |
|  |  |
|  |  | | |

## Validation

This component implement a mechanism that invoke all validation algorithms on the MSN side, get validation result and report to user.

### UML



IValidation interface declare a method to collect all validation methods on the MSN side.

BaseValidation class implement IValidation and create algorithm to get validation results.

ValidationResult class define a struct data contains information like type, error/warning/info message, module id. Type of validation include all information such as ERROR, INFO, WARNING

ValidationRuleAttribute class define attribute to mark whether methods enable validation or not.

MessegeType data type defines some type of validation like ERROR, INFO, WARNING.

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

|  |  |
| --- | --- |
| **Location File Name** | BaseValidation.cs |
| **Description** | This BaseValidation class is a base class that is implemented by other classes used to validate CDF and intermediate data. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | UserInterface | Protected | Contain component user interface used to report error/warning/info to user. |
| (2) | Repository | Protected | Contain component repository used to get data from CDF/BSWMDT/Translation files. |
| (3) | BasicConfiguration | Protected | Contain basic configuration information such as module name, module id,... |
| (4) | IntermediateData | Protected | Contain component intermediate data used to get data from |
| (5) | ForceBothPreAndPostValidation | Protected | Force both pre and post validation |
| (6) | ForceAllContinueIfFailed | Protected | Force all continue if failed |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ValidatePostIntermediate | Public | Run methods validation after all methods in intermediate component invoked. |
| (2) | ValidatePreIntermediate | Public | Run methods validation before all methods in intermediate component invoked. |
| (3) | BaseValidation | Protected | Base validation constructor without arguments |
| (4) | BaseValidation | Protected | Base validation constructor with arguments |
| (5) | BaseValidation | Protected | Base validation constructor with argument userInterface |
| (6) | CheckForMandatoryParameters | Protected | Check mandatory parameters based on input parameters from MCAL module |
| (7) | CheckGeneratedStruct | Protected | Check whether number of fields of sub-structs is the same or not |
| (8) | Check\_ERR\_SPECIFIC\_02 | Protected | To check empty field name in entity structure name |
| (9) | CheckSameFieldsOfSubstructs | Protected | Check whether the fields are the same in substructs or not. |
| (10) | CheckSameFieldsOfSubstructs | Protected | Check whether the fields are the same in substructs or not. |
| (11) | CheckRequiredFieldsOfSubstructs | Protected | To check required field name in entity structure name |
| (12) | CheckRequiredFieldsOfSubstructs | Protected | Check required fields of substructs |
| (13) | GetAllStructureData | Protected | Get all structure data name, an abstract method to be override by specific module to provide structure data in post build source |
| (14) | run | Private | Run all methods in |
| (15) | isMissingContainer | Private | Check whether madatory container exists or not. |
| (16) | addErrorMessage | Private | Create error message template for checking mandatory parameters |
| (17) | checkSameFieldsOfSubstructs | Private | Check same fields of substructs |
| (18) | findFieldsOfAnyLastNodeStruct | Private | Find sample fields of any last node struct |
| (19) | checkRequiredFieldsOfSubstructs | Private | Find sample fields of any last node struct |
| (20) | exitWithType | Private | Update exit code with type such as ERROR, WARN, SUCCESS |

#### Fields

##### UserInterface

GENERIC\_GDD\_CLS\_021\_001:

|  |  |
| --- | --- |
| **Variable Name** | UserInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | Contain component user interface used to report error/warning/info to user. |

##### Repository

GENERIC\_GDD\_CLS\_021\_002:

|  |  |
| --- | --- |
| **Variable Name** | Repository |
| **Type** | IRepository |
| **Range** | None |
| **Description** | Contain component repository used to get data from CDF/BSWMDT/Translation files. |

##### BasicConfiguration

GENERIC\_GDD\_CLS\_021\_003:

|  |  |
| --- | --- |
| **Variable Name** | BasicConfiguration |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | Contain basic configuration information such as module name, module id,... |

##### IntermediateData

GENERIC\_GDD\_CLS\_021\_004:

|  |  |
| --- | --- |
| **Variable Name** | IntermediateData |
| **Type** | IIntermediateData |
| **Range** | None |
| **Description** | Contain component intermediate data used to get data from |

##### ForceBothPreAndPostValidation

GENERIC\_GDD\_CLS\_021\_005:

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

##### ForceAllContinueIfFailed

GENERIC\_GDD\_CLS\_021\_006:

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

#### Methods

##### ValidatePostIntermediate

GENERIC\_GDD\_CLS\_021\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidatePostIntermediate | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Run methods validation after all methods in intermediate component invoked.  Algorithm: CALL run method with parameter runPre = false | | |

{Ref: N/A}

##### ValidatePreIntermediate

GENERIC\_GDD\_CLS\_021\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidatePreIntermediate | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Run methods validation before all methods in intermediate component invoked.  Algorithm: CALL run method with parameter runPre = true | | |

{Ref: N/A}

##### BaseValidation

GENERIC\_GDD\_CLS\_021\_009:

|  |  |
| --- | --- |
| **Method Name** | BaseValidation |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Base validation constructor without arguments |

{Ref: N/A}

##### BaseValidation

GENERIC\_GDD\_CLS\_021\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseValidation | | |
| **Arguments** | userInterface | User interface | |
| Range: | None |
| repository | Repository | |
| Range: | None |
| basicConfiguration | Basic configuration | |
| Range: | None |
| intermediateData | Intermediate data | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base validation constructor with arguments  Algorithm: Init values of component such as UserInterface, Repository, BasicConfiguration, IntermediateData | | |

{Ref: N/A}

##### BaseValidation

GENERIC\_GDD\_CLS\_021\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseValidation | | |
| **Arguments** | userInterface | User Interface | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base validation constructor with argument userInterface  Algorithm: Init value of userInterface | | |

{Ref: N/A}

##### CheckForMandatoryParameters

GENERIC\_GDD\_CLS\_021\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckForMandatoryParameters | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| Hashtable mandatoryParams | Mandatory Parameter | |
| Range: | None |
| IList< string > mandatoryContainers | Mandatory containers | |
| Range: | None |
| **Return** | return List of string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check mandatory parameters based on input parameters from MCAL module  Algorithm: FOREACH def name in mandatoryParams  Get list of containers by this def name  IF NOT found any container in containers THEN  Add an error for missing container to collection "result"  ELSE  FOREACH defname in containers  IF NOT all the mandatory parameters have value in each container THEN  Add an error for each missing parameter to collection "result"  End  End  End  End  Return collection "result" | | |

{Ref: N/A}

##### CheckGeneratedStruct

GENERIC\_GDD\_CLS\_021\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckGeneratedStruct | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether number of fields of sub-structs is the same or not  Algorithm: Scan all required struct from MCAL module input  If have any fields is not the same between sub-structs  Report error ERRXXX001 | | |

{Ref: N/A}

##### CheckRequiredFieldsOfSubstructs

GENERIC\_GDD\_CLS\_021\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckRequiredFieldsOfSubstructs | | |
| **Arguments** | string structName | Struct name | |
| Range: | None |
| List< string > requiredFields | Required fields | |
| Range: | None |
| **Return** | return List of string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | To check required field name in entity structure name  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\_021\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckRequiredFieldsOfSubstructs | | |
| **Arguments** | string dataPath | None | |
| Range: | None |
| string structName | Struct name | |
| Range: | None |
| List< string > requiredFields | Required fields | |
| Range: | None |
| **Return** | return List of string | None | |
| Range: | None |
| **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}

##### CheckSameFieldsOfSubstructs

GENERIC\_GDD\_CLS\_021\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckSameFieldsOfSubstructs | | |
| **Arguments** | string structName | Struct name | |
| Range: | None |
| **Return** | return List of string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether the fields are the same in substructs or not.  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\_021\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CheckSameFieldsOfSubstructs | | |
| **Arguments** | string dataPath | None | |
| Range: | None |
| string structName | Struct name | |
| Range: | None |
| **Return** | return List of string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether the fields are the same in substructs or not.  Algorithm: Get 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\_02

GENERIC\_GDD\_CLS\_021\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Check\_ERR\_SPECIFIC\_02 | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | To check empty field name in entity structure name  Algorithm: Scan all required structs from MCAL module input  If have any required field in sub-struct is missing  Report error ERRXXX002 | | |

{Ref: }

##### GetAllStructureData

GENERIC\_GDD\_CLS\_021\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAllStructureData | | |
| **Arguments** | None | | |
| **Return** | return null | None | |
| Range: | None |
| **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: Implementation will be provided by MCAL module | | |

{Ref: N/A}

##### addErrorMessage

GENERIC\_GDD\_CLS\_021\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | addErrorMessage | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string param | Module Id | |
| Range: | None |
| string defName | Definition name | |
| Range: | None |
| string path | Path of conainer | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Create error message template for checking mandatory parameters  Algorithm: Print error information such as file name, parameter name, path of container if have any | | |

{Ref: N/A}

##### checkRequiredFieldsOfSubstructs

GENERIC\_GDD\_CLS\_021\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkRequiredFieldsOfSubstructs | | |
| **Arguments** | string rootStructName | Struct data | |
| Range: | None |
| structData | None | |
| Range: | None |
| List< string > requiredFields | None | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **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  Add 1 error to collection "result"  End  End  Ends   Return error collection "result" | | |

{Ref: N/A}

##### checkSameFieldsOfSubstructs

GENERIC\_GDD\_CLS\_021\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkSameFieldsOfSubstructs | | |
| **Arguments** | string rootStructName | Root struct name | |
| Range: | None |
| structData | Struct data | |
| Range: | None |
| List< string > fields | Fields | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **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  Add 1 error to collection "result"  End  End  End   Return error collection "result" | | |

{Ref: N/A}

##### exitWithType

GENERIC\_GDD\_CLS\_021\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | exitWithType | | |
| **Arguments** | type | Type | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Update exit code with type such as ERROR, WARN, SUCCESS  Algorithm: 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\_021\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | findFieldsOfAnyLastNodeStruct | | |
| **Arguments** | structData | Struct data | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **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  End  Ends | | |

{Ref: N/A}

##### isMissingContainer

GENERIC\_GDD\_CLS\_021\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | isMissingContainer | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string containerName | Module Id | |
| Range: | None |
| IList< string > mandatoryContainers | Mandatory parameters | |
| Range: | None |
| **Return** | return List of string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether madatory container exists or not.  Algorithm: If container is mandatory which is not configed in CDF files  return true | | |

{Ref: N/A}

##### run

GENERIC\_GDD\_CLS\_021\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | run | | |
| **Arguments** | bool runPre | Pre-validation | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Run all methods in  Algorithm: Scan all validation methods that are decorated by ValidationRuleAttribute  IF runPre = true THEN  Select only pre-validation methods  ELSE  Select only post-validation methods  FOREACH validation method  Run the valiation to get ValidationResult  IF ValidationResult is ERROR THEN  Report error message and Exit  ELSE IF ValidationResult is WARNING THEN  Report warning message  ELSE IF ValiadtionResult is INFO THEN  Report info message  End | | |

{Ref: N/A}

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

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

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Code | Public | Contain code of error |
| (2) | Message | Public | Contain list of message of error/warn/info |
| (3) | ModuleId | Public | Contain module id such as 80 for Can,... |
| (4) | Type | Public | Contain message types such as ERROR/WARN/INFO |
| (5) | Success | Public | Contain success status in |

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

##### Code

GENERIC\_GDD\_CLS\_022\_001:

|  |  |
| --- | --- |
| **Variable Name** | Code |
| **Type** | int |
| **Range** | None |
| **Description** | Contain code of error |

##### Message

GENERIC\_GDD\_CLS\_022\_002:

|  |  |
| --- | --- |
| **Variable Name** | Message |
| **Type** | List< string > |
| **Range** | None |
| **Description** | Contain list of message of error/warn/info |

##### ModuleId

GENERIC\_GDD\_CLS\_022\_003:

|  |  |
| --- | --- |
| **Variable Name** | ModuleId |
| **Type** | int |
| **Range** | None |
| **Description** | Contain module id such as 80 for Can,... |

##### Type

GENERIC\_GDD\_CLS\_022\_004:

|  |  |
| --- | --- |
| **Variable Name** | Type |
| **Type** | MessageType |
| **Range** | None |
| **Description** | Contain message types such as ERROR/WARN/INFO |

##### Success

GENERIC\_GDD\_CLS\_022\_005:

|  |  |
| --- | --- |
| **Variable Name** | Success |
| **Type** | ValidationResult |
| **Range** | None |
| **Description** | Contain success status in |

#### Methods

##### AddMessage

GENERIC\_GDD\_CLS\_022\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AddMessage | | |
| **Arguments** | string msgFormat | Message format | |
| Range: | None |
| params object[] param | Parameters to fill in message format | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Add new message constructed from message format | | |

{Ref: N/A}

##### IsError

GENERIC\_GDD\_CLS\_022\_007:

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

{Ref: N/A}

##### ValidationResult

GENERIC\_GDD\_CLS\_022\_008:

|  |  |
| --- | --- |
| **Method Name** | ValidationResult |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Validation result constructor without arguments  Algorithm: Init value of Type attribute |

{Ref: N/A}

##### ValidationResult

GENERIC\_GDD\_CLS\_022\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidationResult | | |
| **Arguments** | int code | Code | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Validation result constructor with arguments as code  Algorithm: Init values such as Code, Type | | |

{Ref: N/A}

##### ValidationResult

GENERIC\_GDD\_CLS\_022\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidationResult | | |
| **Arguments** | type | Type | |
| Range: | None |
| int code | Code | |
| Range: | None |
| int moduleId | Module id | |
| Range: | None |
| string msgFormat | Message format | |
| Range: | None |
| params object[] param | Parameters to fill in message format | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Validation result constructor with arguments.  Algorithm: Init values such as Code, Module Id, Type, Message | | |

{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 | This attribute mark that all methods run before |
| (2) | PostIntermediationValidation | Public | This attribute mark that all methods run after |
| (3) | ContinueIfFailed | Public | This attribute mark that whether method is allowed run or not. |
| (4) | EnableMethod | Public | This attribute mark that whether method is allowed run or not. |
| (5) | Order | Public | This attribute mark order priority of each validation method in a group |
| (6) | preIntermediationValidation | Private | This attribute mark that all methods run before |
| (7) | postIntermediationValidation | Private | This attribute mark that all methods run after |
| (8) | continueIfFailed | Private | This attribute mark that program still continue although result of method is failed |
| (9) | enableMethod | Private | This attribute mark that whether method is allowed run or not. |

1. Method List

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

#### Fields

##### PreIntermediationValidation

GENERIC\_GDD\_CLS\_023\_001:

|  |  |
| --- | --- |
| **Variable Name** | PreIntermediationValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | This attribute mark that all methods run before |

##### PostIntermediationValidation

GENERIC\_GDD\_CLS\_023\_002:

|  |  |
| --- | --- |
| **Variable Name** | PostIntermediationValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | This attribute mark that all methods run after |

##### ContinueIfFailed

GENERIC\_GDD\_CLS\_023\_003:

|  |  |
| --- | --- |
| **Variable Name** | ContinueIfFailed |
| **Type** | bool |
| **Range** | None |
| **Description** | This attribute mark that whether method is allowed run or not. |

##### EnableMethod

GENERIC\_GDD\_CLS\_023\_004:

|  |  |
| --- | --- |
| **Variable Name** | EnableMethod |
| **Type** | bool |
| **Range** | None |
| **Description** | This attribute mark that whether method is allowed run or not. |

##### Order

GENERIC\_GDD\_CLS\_023\_006:

|  |  |
| --- | --- |
| **Variable Name** | Order |
| **Type** | float |
| **Range** | None |
| **Description** | This attribute mark order priority of each validation method in a group |

##### preIntermediationValidation

GENERIC\_GDD\_CLS\_023\_007:

|  |  |
| --- | --- |
| **Variable Name** | preIntermediationValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | This attribute mark that all methods run before |

##### postIntermediationValidation

GENERIC\_GDD\_CLS\_023\_008:

|  |  |
| --- | --- |
| **Variable Name** | postIntermediationValidation |
| **Type** | bool |
| **Range** | None |
| **Description** | This attribute mark that all methods run after |

##### continueIfFailed

GENERIC\_GDD\_CLS\_023\_009:

|  |  |
| --- | --- |
| **Variable Name** | continueIfFailed |
| **Type** | bool |
| **Range** | None |
| **Description** | This attribute mark that program still continue although result of method is failed |

##### enableMethod

GENERIC\_GDD\_CLS\_023\_010:

|  |  |
| --- | --- |
| **Variable Name** | enableMethod |
| **Type** | bool |
| **Range** | None |
| **Description** | This attribute mark that whether method is allowed run or not. |

#### Methods

##### ValidationRuleAttribute

GENERIC\_GDD\_CLS\_023\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidationRuleAttribute | | |
| **Arguments** | float order | Order of method running | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Validation rule attribute constructor with argument order  Algorithm: Init value order | | |

{Ref: N/A}

##### ValidationRuleAttribute

GENERIC\_GDD\_CLS\_023\_013:

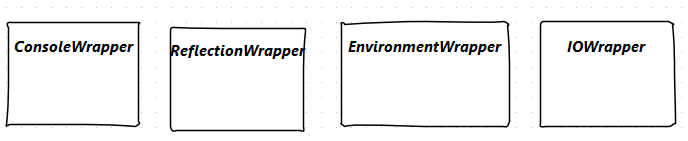
|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ValidationRuleAttribute | | |
| **Arguments** | bool preIntermediationValidation | Pre intermediation validation | |
| Range: | None |
| bool postIntermediationValidation | Post intermediation validation | |
| Range: | None |
| bool continueIfFailed | Continue if failed | |
| Range: | None |
| bool enableMethod | Enable/Disable method running | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Validation rule attribute constructor with arguments  Algorithm: Init values such as preIntermediationValidation, postIntermediationValidation, continueIfFailed  enableMethod | | |

{Ref: N/A}

## DotNetApiWrapper

Generation Tool is developed by NET APIs, so this component manage errors that occur on NET API and concentrated error code. The MCAL module use Wrapper API instead of direct NET API.

### UML



|  |  |
| --- | --- |
| Class | Description |
| ConsoleWrapper | Wrap Console class in NET and handle related exceptions |
| ReflectionWrapper | Wrap Reflection class in NET and handle related exceptions |
| EnvironmentWrapper | Wrap Environment class in NET and handle related exceptions |
| IOWrapper | Wrap Directory, File, Xdocument, Path classes in NET and handle related exceptions |

### 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 | Contain user interface is used to report error/warn/info messages to user. |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | WriteLine | Public | Call .NET API to write to console |

#### Fields

##### ui

GENERIC\_GDD\_CLS\_024\_001:

|  |  |
| --- | --- |
| **Variable Name** | ui |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | Contain user interface is used to report error/warn/info messages to user. |

#### Methods

##### WriteLine

GENERIC\_GDD\_CLS\_024\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | WriteLine | | |
| **Arguments** | string value | string value | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Call .NET API to write to console  Algorithm: This method 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 |

#### Fields

#### Methods

##### Exit

GENERIC\_GDD\_CLS\_025\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Exit | | |
| **Arguments** | int exitCode | int value | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Call .NET API to exit application  Algorithm: This method 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 | Contain user interface used to report error/warn/info messages to user. |

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\_026\_001:

|  |  |
| --- | --- |
| **Variable Name** | ui |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | Contain user interface used to report error/warn/info messages to user. |

#### Methods

##### CreateDirectory

GENERIC\_GDD\_CLS\_026\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CreateDirectory | | |
| **Arguments** | string path | string path | |
| Range: | None |
| int? exitCode | int? value | |
| Range: | Null or not |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase create directory  Algorithm: This method call .NET API Directory.CreateDirectory(path) to create a new directory.  If exception(s) is(are) thrown, 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] CMN\_GFD\_ERR\_018, CMN\_GFD\_ERR\_050, CMN\_GFD\_ERR\_051, GENERIC\_TFD\_X2x\_GT\_ERR\_018, GENERIC\_TFD\_X2x\_GT\_ERR\_050}

##### DirectoryExists

GENERIC\_GDD\_CLS\_026\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | DirectoryExists | | |
| **Arguments** | string directory | string path | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | This method call .NET API to check if a directory exists  Algorithm: Call .NET API Directory.Exists(directory) to check if a directory exists  If exception is thrown, record ERR000050 and exit application | | |

{Ref: [1] CMN\_GFD\_ERR\_050}

##### FileExists

GENERIC\_GDD\_CLS\_026\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | FileExists | | |
| **Arguments** | string file | string file | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Call .NET API to check if a file exists.  Algorithm: Call .NET API File.Exists(file) to get current directory of executing application.  If exception is thrown, record ERR000050 and exit application | | |

{Ref: [1] CMN\_GFD\_ERR\_050}

##### GetCurrentDirectory

GENERIC\_GDD\_CLS\_026\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetCurrentDirectory | | |
| **Arguments** | int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | This method call .NET API to get current directory of executing application  Algorithm: Call .NET API Directory.GetCurrentDirectory() to get current directory of executing application.  If exception is thrown, record ERR000050 and exit application | | |

{Ref: [1] CMN\_GFD\_ERR\_050}

##### GetFileName

GENERIC\_GDD\_CLS\_026\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetFileName | | |
| **Arguments** | string fileName | string file | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase get file name and extension path string  Algorithm: This method call .NET API Path.GetFileName(fileName) to get the file name  and extension of the specified path string.  If exception(s) is(are) thrown, record the error and exit application  Handle .NET exceptions:  ArgumentException: record error ERR000053 | | |

{Ref: [1] CMN\_GFD\_ERR\_053}

##### GetFiles

GENERIC\_GDD\_CLS\_026\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetFiles | | |
| **Arguments** | string path | string path | |
| Range: | Null or not |
| string searchPattern | string path | |
| Range: | TopDirectoryOnly AllDirectories |
| SearchOption searchOption | int? exitCode | |
| Range: | Null or not |
| int? exitCode | None | |
| Range: | None |
| **Return** | return a string array | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase get files  Algorithm: This method call .NET API Directory.GetFiles(path, searchPattern, searchOption) to search  file names by pattern.  If exception(s) is(are) thrown, record the error and exit application  Handle .NET exceptions:  UnauthorizedAccessException: record error ERR000050  PathTooLongException: record error ERR000051  IOException: record error ERR000018 | | |

{Ref: [1] CMN\_GFD\_ERR\_018, CMN\_GFD\_ERR\_050, CMN\_GFD\_ERR\_051, GENERIC\_TFD\_X2x\_GT\_ERR\_051}

##### GetFullPath

GENERIC\_GDD\_CLS\_026\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetFullPath | | |
| **Arguments** | string path | string path | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase get the absolute path  Algorithm: This method call .NET API Path.GetFullPath(path) to returns the absolute path for the specified path string  If exception(s) is(are) thrown, record the error and exit application  Handle .NET exceptions:  SecurityException: record error ERR000050  PathTooLongException: record error ERR000051 | | |

{Ref: [1] CMN\_GFD\_ERR\_050, CMN\_GFD\_ERR\_051}

##### LoadXML

GENERIC\_GDD\_CLS\_026\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | LoadXML | | |
| **Arguments** | string file | string file | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a XDocument | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase load XML file.  Algorithm: This method call .NET API XDocument.Load(file) to load a XML file.  If exception(s) is(are) thrown, record the error and exit application  Handle .NET exceptions:  FormatException: record error ERR000012  XmlException: record error ERR000012  UnauthorizedAccessException: record error ERR000050 | | |

{Ref: [1] CMN\_GFD\_ERR\_012, CMN\_GFD\_ERR\_050}

##### ReadAllLines

GENERIC\_GDD\_CLS\_026\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ReadAllLines | | |
| **Arguments** | string file | string path | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a string array | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase read content of a file  Algorithm: This method 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, record the error and exit application  Handle .NET exceptions:  UnauthorizedAccessException: record error ERR000050  SecurityException: record error ERR000050  IOException: record error ERR000053 | | |

{Ref: [1] CMN\_GFD\_ERR\_050, CMN\_GFD\_ERR\_053, GENERIC\_TFD\_X2x\_GT\_ERR\_053}

##### StreamWrite

GENERIC\_GDD\_CLS\_026\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StreamWrite | | |
| **Arguments** | StreamWriter streamWriter | StreamWriter streamWriter | |
| Range: | Null or not |
| string content | string content | |
| Range: | Null or not |
| string path | string path | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase write content to streamWriter  Algorithm: This method 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, record the error and exit application  Handle .NET exceptions:  IOException: record error ERR000018 | | |

{Ref: [1] CMN\_GFD\_ERR\_018}

##### StreamWriteLine

GENERIC\_GDD\_CLS\_026\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | StreamWriteLine | | |
| **Arguments** | StreamWriter streamWriter | StreamWriter streamWriter | |
| Range: | Null or not |
| string content | string content | |
| Range: | Null or not |
| string path | string path | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase write content to streamWriter  Algorithm: This method 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, record the error and exit application  Handle .NET exceptions:  IOException: record error ERR000018 | | |

{Ref: [1] CMN\_GFD\_ERR\_018}

##### Exit

GENERIC\_GDD\_CLS\_026\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Exit | | |
| **Arguments** | int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Call to exit program  Algorithm: This method receive parameter as exit code value from calling method.  If parameter does not provide, it will 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 | Contain user interface used to report error/warn/info messages to user. |

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\_027\_001:

|  |  |
| --- | --- |
| **Variable Name** | ui |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | Contain user interface used to report error/warn/info messages to user. |

#### Methods

##### LoadFile

GENERIC\_GDD\_CLS\_027\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | LoadFile | | |
| **Arguments** | string path | string file | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase load the contents of an assembly file.  Algorithm: This method call .NET API Assembly.LoadFile(path) to load the contents  of an assembly file on the specified path.  If exception(s) is(are) thrown, record the error and exit application  Handle .NET exceptions:  FileLoadException: record error ERR000055  BadImageFormatException: record error ERR000056 | | |

{Ref: [1] CMN\_GFD\_ERR\_055, CMN\_GFD\_ERR\_056, GENERIC\_TFD\_X2x\_GT\_ERR\_055}

##### LoadFrom

GENERIC\_GDD\_CLS\_027\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | LoadFrom | | |
| **Arguments** | string path | string file | |
| Range: | Null or not |
| int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Handle kind of .NET exceptions incase load an assembly given its file name or path.  Algorithm: This method call .NET API Assembly.LoadFrom(path) to load an assembly  given its file name or path.  If exception(s) is(are) thrown, 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] CMN\_GFD\_ERR\_050, CMN\_GFD\_ERR\_051, CMN\_GFD\_ERR\_055, CMN\_GFD\_ERR\_056, GENERIC\_TFD\_X2x\_GT\_ERR\_056}

##### Exit

GENERIC\_GDD\_CLS\_027\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Exit | | |
| **Arguments** | int? exitCode | int? exitCode | |
| Range: | Null or not |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Call to exit program  Algorithm: This method receive parameter as exit code value from calling method.  If parameter does not provide, it will exit program without status code. | | |

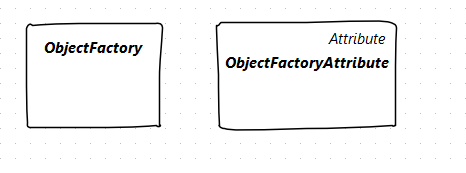
{Ref: N/A}

## ObjectFactory

This component resolve an interface into one or many instances of concrete classes which actually handle requests at runtime. This mechanism is like dependency injection pattern.

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).

### UML



The ObjectFactory class scan dynamic class library for each MCAL module and load all interfaces, classes, methods into cache. Afterward they are processed by components such as Validation, Intermediate, Generation.

The ObjectFactoryAttribute class is used to mark whether class is allowed loading or not.

### 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 | Contain module name information |
| (2) | SupportedDevices | Public | Contain all devices are supported by MCAL module |
| (3) | gentoolExecutableTypes | Private | Contain information executable type under |
| (4) | registeredTypes | Private | Contain types that are loaded from DLLs such as interface, class |
| (5) | cache | Private | A cache of initialized instances for interfaces |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | RegisterAssemblyTypes | 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) | ObjectFactory | Private | Object factory constructor without arguments |
| (5) | createInstance | Private | Create object instance from a concrete type |
| (6) | checkConditionType | Private | Check type to meet Autosar and Device conditions |
| (7) | getCacheObjectsOfInterface | Private | Get cache object instance based on specified interface |
| (8) | parseOtherConditions | Private | Parsing other conditions of |

#### Fields

##### ModuleName

GENERIC\_GDD\_CLS\_028\_001:

|  |  |
| --- | --- |
| **Variable Name** | ModuleName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain module name information |

##### SupportedDevices

GENERIC\_GDD\_CLS\_028\_002:

|  |  |
| --- | --- |
| **Variable Name** | SupportedDevices |
| **Type** | List< string > |
| **Range** | None |
| **Description** | Contain all devices are supported by MCAL module |

##### gentoolExecutableTypes

GENERIC\_GDD\_CLS\_028\_003:

|  |  |
| --- | --- |
| **Variable Name** | gentoolExecutableTypes |
| **Type** | HashSet< Type > |
| **Range** | None |
| **Description** | Contain information executable type under |

##### registeredTypes

GENERIC\_GDD\_CLS\_028\_004:

|  |  |
| --- | --- |
| **Variable Name** | registeredTypes |
| **Type** | Dictionary< Type, List< Type > > |
| **Range** | None |
| **Description** | Contain types that are loaded from DLLs such as interface, class |

##### cache

GENERIC\_GDD\_CLS\_028\_005:

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

#### Methods

##### GetInstance< T >

GENERIC\_GDD\_CLS\_028\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetInstance< T > | | |
| **Arguments** | None | | |
| **Return** | return instance object with interface T | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get an instance object with interface is T  Algorithm: Get all instance objects with interface is T  IF found  Return first object  ELSE  Exit Gentool | | |

{Ref: N/A}

##### GetInstances< T >

GENERIC\_GDD\_CLS\_028\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetInstances< T > | | |
| **Arguments** | None | | |
| **Return** | return all instance objects with interface T | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get all instance objects with interface is T  Algorithm: Get all instance objects with interface is T  IF found  RETURN all instance objects  ELSE  Exit Gentool | | |

{Ref: N/A}

##### RegisterAssemblyTypes

GENERIC\_GDD\_CLS\_028\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | RegisterAssemblyTypes | | |
| **Arguments** | string[] assemblyFilePaths | Assembly file paths | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Register all types from files specified by assemblyFilePaths  Algorithm: FOREACH file in assemblyFilePaths  Load all types from assembly file  Store all types in registeredTypes  End | | |

{Ref: N/A}

##### ObjectFactory

GENERIC\_GDD\_CLS\_028\_009:

|  |  |
| --- | --- |
| **Method Name** | ObjectFactory |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Object factory constructor without arguments  Algorithm: Load types of executable MCALConfGen  For each type  Get interface of type  Store to dictionary registeredTypes[interface] = type  End |

{Ref: N/A}

##### checkConditionType

GENERIC\_GDD\_CLS\_028\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | checkConditionType | | |
| **Arguments** | Type concreteType | Concrete type | |
| Range: | None |
| List< Type > allRelatedTypes | All related types | |
| Range: | None |
| Type interfaceType | Interface | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check type to meet Autosar and Device conditions  Algorithm: IF AutoSarVersion is "\*" OR is basicConfig.AutoSARVersion AND Device is "\*"  OR basicConfig.DeviceNames contain Device THEN  RETURN true  ELSE  RETURN false | | |

{Ref: N/A}

##### createInstance

GENERIC\_GDD\_CLS\_028\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | createInstance | | |
| **Arguments** | Type concreteType | Concrete type | |
| Range: | None |
| **Return** | return a Object | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Create object instance from a concrete type  Algorithm: From concrete type, get Instance fields  IF found  Return value of the field  ELSE  Exit Gentool | | |

{Ref: N/A}

##### getCacheObjectsOfInterface

GENERIC\_GDD\_CLS\_028\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getCacheObjectsOfInterface | | |
| **Arguments** | Type interfaceType | Interface | |
| Range: | None |
| **Return** | return List of CacheObject | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get cache object instance based on specified interface  Algorithm: IF cache stores object instances of the interface THEN  RETURN object instances  ELSE  Scan all types that have the interface in registeredTypes  With each type, create a new instance and add to cache  End  Return object instances from cache | | |

{Ref: N/A}

##### parseOtherConditions

GENERIC\_GDD\_CLS\_028\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseOtherConditions | | |
| **Arguments** | string otherConditions | Other conditions | |
| Range: | None |
| **Return** | return Dictionary of conditions, key is parameter/reference name and value is expected value | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing other conditions of  Algorithm: IF otherConditions is empty string THEN  Return empty dictionary  ELSE  Split otherConditions into multiple conditions (separator is ;)  FOREACH condition in other condition  Split condition by separator '=' and add to result dictionary  End  Return result dictionary | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **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 ObjectFactoryAttribute's fields of the classes. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | Contain intance of object such as class, interface |
| (2) | TypeOf | Public | Contain type of object |

1. Method List

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

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_029\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | Object |
| **Range** | None |
| **Description** | Contain intance of object such as class, interface |

##### TypeOf

GENERIC\_GDD\_CLS\_029\_002:

|  |  |
| --- | --- |
| **Variable Name** | TypeOf |
| **Type** | Type |
| **Range** | None |
| **Description** | Contain type of object |

#### Methods

##### CacheObject

GENERIC\_GDD\_CLS\_029\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CacheObject | | |
| **Arguments** | Type typeOf | Type | |
| Range: | None |
| object instance | Object instance | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Cache object constructor with arguments  Algorithm: Init values such as typeOf, instance | | |

{Ref: N/A}

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

|  |  |
| --- | --- |
| **Location File Name** | ObjectFactoryAttribute.cs |
| **Description** | This class contains attribute to be resolved by ObjectFactory for its implemented inferface, it should be decorated by ObjectFactoryAttribute. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | AutoSarVersion | Public | Contain information autosar version that is supported by tool |
| (2) | Device | Public | Contain information device that is supported by tool |
| (3) | Version | Public | Contain information version that is supported by tool |
| (4) | ModuleNames | Public | Contain information module name in MCAL module |
| (5) | OtherConditions | Public | Other conditions Format: |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ObjectFactoryAttribute | Public | Object factory attribute constructor without arguments |

#### Fields

##### AutoSarVersion

GENERIC\_GDD\_CLS\_030\_001:

|  |  |
| --- | --- |
| **Variable Name** | AutoSarVersion |
| **Type** | string |
| **Range** | None |
| **Description** | Contain information autosar version that is supported by tool |

##### Device

GENERIC\_GDD\_CLS\_030\_002:

|  |  |
| --- | --- |
| **Variable Name** | Device |
| **Type** | string |
| **Range** | None |
| **Description** | Contain information device that is supported by tool |

##### Version

GENERIC\_GDD\_CLS\_030\_003:

|  |  |
| --- | --- |
| **Variable Name** | Version |
| **Type** | string |
| **Range** | None |
| **Description** | Contain information version that is supported by tool |

##### ModuleNames

GENERIC\_GDD\_CLS\_030\_004:

|  |  |
| --- | --- |
| **Variable Name** | ModuleNames |
| **Type** | string |
| **Range** | None |
| **Description** | Contain information module name in MCAL module |

##### OtherConditions

GENERIC\_GDD\_CLS\_030\_005:

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

#### Methods

##### ObjectFactoryAttribute

GENERIC\_GDD\_CLS\_030\_006:

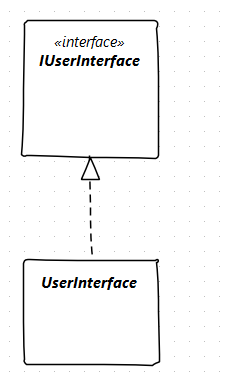
|  |  |
| --- | --- |
| **Method Name** | ObjectFactoryAttribute |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Object factory attribute constructor without arguments |

{Ref: N/A}

## UserInterface

This component manage status of Generation Tool like success/fail end and display information such as error, warining, info to user.

### UML



UserInterface class implement features of Tool such as display error/info/warning information, end status of Tool as success or fail. In addition, this class bear the responsibility for writing log, formating error/info/warning message.

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

|  |  |
| --- | --- |
| **Location File Name** | UserInterface.cs |
| **Description** | This class that implements IUserInterface interface to record Error, Warning, Info information and exit Gentool |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | Contain UserInterface, which is created by ObjectFactory |
| (2) | ErrorCount | Public | Contain number of error used to report to user |
| (3) | WarningCount | Public | Contain number of warning used to report to user |
| (4) | runtimeConfig | Private | Contain RuntimeConfig, which is created by |
| (5) | basicConfig | Private | Contain BasicConfiguration, which is created by |
| (6) | swLog | Private | Contain stream writer to write log when need |
| (7) | logContent | Private | Coontain log content and write content to log file |
| (8) | errorCount | Private | Contain number of error used to report to user |
| (9) | warningCount | Private | Contain number of warning used to report to user |
| (10) | inExitMode | Private | Contain information exit mode that record status of tool such as success or no success |
| (11) | exitCodeDic | Private | Match each class corresponding to report types |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Error | Public | Report error information to user |
| (2) | Warn | Public | Report warning information to user |
| (3) | Info | Public | Report information to user |
| (4) | OpenFile | Public | Open file from path to check able writing to file. |
| (5) | Exit | Public | Exit tool with status as success or fail |
| (6) | Exit | Public | Exit Gentool with parameter exit code |
| (7) | ErrorFileNotFound | Public | Report error when file does not exists based file path input from user |
| (8) | ErrorModuleNotFound | Public | Report error message if module name is not provided by user. |
| (9) | InfoErrorCode | Public | Report proper information based on error code |
| (10) | getDateTime | Private | Get current date time with format "dd MMM yyyy - hh:mm:ss" |
| (11) | getCallerClass | Private | Find class information cause error/info/warn to report to user |
| (12) | report | Private | create report template by concatenating module id, error code and message |
| (13) | textWrap | Private | Format error/info/warn message such as wrap text to ensure content is not more 80 chararters. |
| (14) | writeLog | Private | Writing content to log file |

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_031\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly UserInterface |
| **Range** | None |
| **Description** | Contain UserInterface, which is created by ObjectFactory |

##### ErrorCount

GENERIC\_GDD\_CLS\_031\_002:

|  |  |
| --- | --- |
| **Variable Name** | ErrorCount |
| **Type** | int |
| **Range** | None |
| **Description** | Contain number of error used to report to user |

##### WarningCount

GENERIC\_GDD\_CLS\_031\_003:

|  |  |
| --- | --- |
| **Variable Name** | WarningCount |
| **Type** | int |
| **Range** | None |
| **Description** | Contain number of warning used to report to user |

##### runtimeConfig

GENERIC\_GDD\_CLS\_031\_004:

|  |  |
| --- | --- |
| **Variable Name** | runtimeConfig |
| **Type** | IRuntimeConfiguration |
| **Range** | None |
| **Description** | Contain RuntimeConfig, which is created by |

##### basicConfig

GENERIC\_GDD\_CLS\_031\_005:

|  |  |
| --- | --- |
| **Variable Name** | basicConfig |
| **Type** | IBasicConfiguration |
| **Range** | None |
| **Description** | Contain BasicConfiguration, which is created by |

##### swLog

GENERIC\_GDD\_CLS\_031\_006:

|  |  |
| --- | --- |
| **Variable Name** | swLog |
| **Type** | StreamWriter |
| **Range** | None |
| **Description** | Contain stream writer to write log when need |

##### logContent

GENERIC\_GDD\_CLS\_031\_007:

|  |  |
| --- | --- |
| **Variable Name** | logContent |
| **Type** | StringBuilder |
| **Range** | None |
| **Description** | Coontain log content and write content to log file |

##### errorCount

GENERIC\_GDD\_CLS\_031\_008:

|  |  |
| --- | --- |
| **Variable Name** | errorCount |
| **Type** | int |
| **Range** | None |
| **Description** | Contain number of error used to report to user |

##### warningCount

GENERIC\_GDD\_CLS\_031\_009:

|  |  |
| --- | --- |
| **Variable Name** | warningCount |
| **Type** | int |
| **Range** | None |
| **Description** | Contain number of warning used to report to user |

##### inExitMode

GENERIC\_GDD\_CLS\_031\_010:

|  |  |
| --- | --- |
| **Variable Name** | inExitMode |
| **Type** | bool |
| **Range** | None |
| **Description** | Contain information exit mode that record status of tool such as success or no success |

##### exitCodeDic

GENERIC\_GDD\_CLS\_031\_011:

|  |  |
| --- | --- |
| **Variable Name** | exitCodeDic |
| **Type** | Dictionary< Type, int > |
| **Range** | None |
| **Description** | Match each class corresponding to report types |

#### Methods

##### Error

GENERIC\_GDD\_CLS\_031\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Error | | |
| **Arguments** | int moduleId | Module id | |
| Range: | None |
| int errorCode | Error code | |
| Range: | None |
| string message | Message | |
| Range: | None |
| params object[] parameters | Parameters | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Algorithm** |  | | |

{Ref: N/A}

##### ErrorFileNotFound

GENERIC\_GDD\_CLS\_031\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ErrorFileNotFound | | |
| **Arguments** | string fileName | File name | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Report error when file does not exists based file path input from user  Algorithm: Report error ERR000001  CALL Exit method with exitCode EXIT\_DUE\_CMD\_ERROR | | |

{Ref: [1] CMN\_GFD\_ERR\_001}

##### ErrorModuleNotFound

GENERIC\_GDD\_CLS\_031\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ErrorModuleNotFound | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Report error message if module name is not provided by user.  Algorithm: Report error ERR000031  Report info INF000005  CALL infoErrorCode method to record info based on error code EXIT\_DUE\_CMD\_ERROR  Environment.Exit with 1 | | |

{Ref: [1] CMN\_GFD\_ERR\_031, CMN\_GFD\_INF\_005, GENERIC\_TFD\_X2x\_GT\_ERR\_031}

##### Exit

GENERIC\_GDD\_CLS\_031\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Exit | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Exit tool with status as success or fail  Algorithm: Identify exitcode from caller class  IF caller class NOT found in exitCodeDic THEN  exitcode = EXIT\_SUCCESSFUL  End  Call Exit method with parameter exitcode | | |

{Ref: GENERIC\_TFD\_X2x\_GT\_USG\_013}

##### Exit

GENERIC\_GDD\_CLS\_031\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Exit | | |
| **Arguments** | int exitCode | Exit code | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Exit Gentool with parameter exit code  Algorithm: Use basicConfig.OutputFolder if runtimeConfig.FolderOutput NOT existed  IF runtimeConfig.DryRun == false THEN  Write to log file  Report error ERR000004 when can NOT write log file  End   Report info INF000005  CALL infoErrorCode method to record info based on exitCode  IF exitCode = EXIT\_SUCCESSFUL then  Environment.Exit with 0  ELSE  Environment.Exit with 1 | | |

{Ref: [1] CMN\_GFD\_ERR\_004, CMN\_GFD\_INF\_005, GENERIC\_TFD\_X2x\_GT\_ERR\_004}

##### Info

GENERIC\_GDD\_CLS\_031\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Info | | |
| **Arguments** | int moduleId | Module id | |
| Range: | None |
| int errorCode | Error code | |
| Range: | None |
| string message | Message | |
| Range: | None |
| params object[] parameters | Parameters | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Algorithm** |  | | |

{Ref: N/A}

##### InfoErrorCode

GENERIC\_GDD\_CLS\_031\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | InfoErrorCode | | |
| **Arguments** | int errorcode | Errorcode | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Algorithm** |  | | |

{Ref: [1] CMN\_GFD\_INF\_006, CMN\_GFD\_INF\_007, CMN\_GFD\_INF\_008, CMN\_GFD\_INF\_009, CMN\_GFD\_INF\_010, CMN\_GFD\_INF\_011, CMN\_GFD\_INF\_012, CMN\_GFD\_INF\_014, GENERIC\_TFD\_X2x\_GT\_INF\_006, GENERIC\_TFD\_X2x\_GT\_INF\_007, GENERIC\_TFD\_X2x\_GT\_INF\_008, GENERIC\_TFD\_X2x\_GT\_INF\_009, GENERIC\_TFD\_X2x\_GT\_INF\_010, GENERIC\_TFD\_X2x\_GT\_INF\_011, GENERIC\_TFD\_X2x\_GT\_INF\_012, GENERIC\_TFD\_X2x\_GT\_INF\_014}

##### OpenFile

GENERIC\_GDD\_CLS\_031\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | OpenFile | | |
| **Arguments** | string path | Path | |
| Range: | None |
| **Return** | return a StreamWriter | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: [1] CMN\_GFD\_INF\_004, GENERIC\_TFD\_X2x\_GT\_INF\_004}

##### Warn

GENERIC\_GDD\_CLS\_031\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Warn | | |
| **Arguments** | int moduleId | Module id | |
| Range: | None |
| int errorCode | Error code | |
| Range: | None |
| string message | Message | |
| Range: | None |
| params object[] parameters | Parameters | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: N/A}

##### getCallerClass

GENERIC\_GDD\_CLS\_031\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getCallerClass | | |
| **Arguments** | None | | |
| **Return** | return a Type | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** |  | | |

{Ref: N/A}

##### getDateTime

GENERIC\_GDD\_CLS\_031\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getDateTime | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **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\_031\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | report | | |
| **Arguments** | int moduleId | Module id | |
| Range: | None |
| int errorCode | Error code | |
| Range: | None |
| string msgType | Message type: ERR, WRN and INF | |
| Range: | None |
| string msg | Message format | |
| Range: | None |
| params object[] parameters | Parameters | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Algorithm** |  | | |

{Ref: N/A}

##### textWrap

GENERIC\_GDD\_CLS\_031\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | textWrap | | |
| **Arguments** | string prefix | Prefix | |
| Range: | None |
| string inputString | Input string | |
| Range: | None |
| int width | Width | |
| Range: | None |
| char separator | Separator | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Algorithm** |  | | |

{Ref: N/A}

##### writeLog

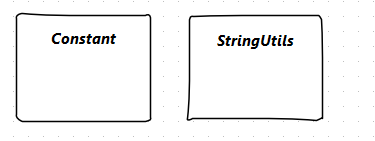
GENERIC\_GDD\_CLS\_031\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | writeLog | | |
| **Arguments** | None | | |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Algorithm** |  | | |

{Ref: N/A}

## Util

### UML



This component contain util classes like Contant class to define some contants is used in others component, StringUtils class circumscribe gain methods such as checking whether value follow to Syntax C or not, converting hex value to integer,…

### 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) | TWO | Public | TWO |
| (14) | ONE | Public | ONE |
| (15) | ZERO | Public | ZERO |
| (16) | TRUE | Public | TRUE |
| (17) | FALSE | Public | FALSE |
| (18) | IOREG | Public | IOREG |
| (19) | OPTION | Public | OPTION |
| (20) | INPUT\_FILE | Public | Input file |
| (21) | AR\_RELEASE\_VERSION | Public | Ar release version |
| (22) | SW\_VERSION | Public | Ar software version |
| (23) | VENDOR\_ID | Public | Vendor id |
| (24) | MODULE\_ID | Public | Module id |
| (25) | EXIT\_DUE\_CMD\_ERROR | Public | Exit due to command line error |
| (26) | EXIT\_SUCCESSFUL | Public | Exit successful |
| (27) | EXIT\_SUCCESSFUL\_WITH\_WARNINGS | Public | Exit successful with warnings |
| (28) | EXIT\_DUE\_INPUT\_FILES\_ERROR | Public | Exit due to input files error |
| (29) | EXIT\_DUE\_STRUCTURE\_GENERATION\_ERROR | Public | Exit due to structure generation error |
| (30) | EXIT\_DUE\_OVERWRITTEN\_FILE\_ERROR | Public | Exit due to overwritten file error |
| (31) | EXIT\_DUE\_UNCOMPATIBLE\_DLL | Public | Exit due to uncompatible with DLL file |
| (32) | EXIT\_DUE\_CRASH | Public | Exit due to unknown error |
| (33) | EXIT\_DUE\_DLL\_NOT\_FOUND | Public | Exit due to DLL file not found |
| (34) | AR\_PACKAGE | Public | Ar package |
| (35) | AR\_PACKAGES | Public | Ar packages |
| (36) | ELEMENTS | Public | Elements |
| (37) | SHORT\_NAME | Public | Short name |
| (38) | ECUC\_VALUE\_COLLECTION | Public | Ecuc value collection |
| (39) | BOOL\_DATA\_TYPE\_MARKED | Public | Mark for ECUC BOOLEAN PARAM DEF attribute CDF |
| (40) | FLOAT\_DATA\_TYPE\_MARKED | Public | Mark for ECUC\_FLOAT\_PARAM\_DEF attribute CDF |
| (41) | ECUC\_INTEGER\_PARAM\_DEF | Public | Mark for ECUC\_INTEGER\_PARAM\_DEF attribute CDF |
| (42) | ECUC\_STRING\_PARAM\_DEF | Public | Mark for ECUC\_STRING\_PARAM\_DEF attribute CDF |
| (43) | ECUC\_LINKER\_SYMBOL\_DEF | Public | Mark for ECUC\_LINKER\_SYMBOL\_DEF attribute CDF |
| (44) | ECUC\_FUNCTION\_NAME\_DEF | Public | Mark for ECUC\_FUNCTION\_NAME\_DEF attribute CDF |
| (45) | ECUC\_ENUMERATION\_PARAM\_DEF | Public | Mark for ECUC\_ENUMERATION\_PARAM\_DEF attribute CDF |
| (46) | ECUC\_ENUMERATION\_LITERAL\_DEF | Public | Mark for ECUC\_ENUMERATION\_LITERAL\_DEF attribute CDF |
| (47) | ECUC\_SYMBOLIC\_NAME\_REFERENCE\_DEF | Public | Mark for ECUC\_SYMBOLIC\_NAME\_REFERENCE\_DEF attribute CDF |
| (48) | ECUC\_URI\_REFERENCE\_DEF | Public | Mark for ECUC\_URI\_REFERENCE\_DEF attribute CDF |
| (49) | ECUC\_ADD\_INFO\_PARAM\_DEF | Public | Mark for ECUC\_ADD\_INFO\_PARAM\_DEF attribute CDF |
| (50) | AR\_VERSION\_4\_2\_2 | Public | AutoSar Version 4.2.2 |
| (51) | AR\_VERSION\_4\_0\_3 | Public | AutoSar Version 4.0.3 |
| (52) | TOOL\_VERSION | Public | Generation tool version control |
| (53) | ALL\_OBJECT | Public | Select all object related to Autosar Version and Device Name. |
| (54) | DEVICE\_NAME\_D1L | Public | List of device name D1L family. |
| (55) | DEVICE\_NAME\_D1L1 | Public | List of device name D1L1 series. |
| (56) | DEVICE\_NAME\_D1L2 | Public | List of device name D1L2 series. |
| (57) | DEVICE\_NAME\_D1L2H | Public | List of device name D1L2H series. |
| (58) | DEVICE\_NAME\_D1M | Public | List of device name D1M family. |
| (59) | DEVICE\_NAME\_D1M1 | Public | List of device name D1M1 series. |
| (60) | DEVICE\_NAME\_D1M1H | Public | List of device name D1M1H series. |
| (61) | DEVICE\_NAME\_D1M1V2 | Public | List of device name D1M1V2 series. |
| (62) | DEVICE\_NAME\_D1M1A | Public | List of device name D1M1A series. |
| (63) | DEVICE\_NAME\_D1M2 | Public | List of device name D1M2 series. |
| (64) | DEVICE\_NAME\_D1M2H | Public | List of device name D1M2H series. |
| (65) | DEVICE\_NAME\_D1S | Public | List of device name D1S family. |
| (66) | DEVICE\_NAME\_D1S1 | Public | List of device name D1S1 series. |
| (67) | ModuleOption | Public | Option of module such as -m, -module |
| (68) | ConfigFileOption | Public | None |
| (69) | OutputOption | Public | None |
| (70) | FilterOption | Public | None |
| (71) | HelpOption | Public | None |
| (72) | DryrunOption | Public | None |
| (73) | LogOption | Public | None |
| (74) | OutputIncSrcOption | Public | None |
| (75) | FILTER\_RENESAS | Public | None |
| (76) | OUTPUT | Public | None |
| (77) | FILTER | Public | None |
| (78) | OUTPUT\_PATH | Public | None |
| (79) | ON\_CONST | Public | None |
| (80) | OFF\_CONST | Public | None |
| (81) | FILTER\_NAME | Public | None |
| (82) | Extension | Public | None |
| (83) | MODULE\_NAME | Public | None |
| (84) | MICROFAMILY\_NAME | Public | None |
| (85) | MODULE\_REQUIRED | Public | None |
| (86) | PROJECT\_TITLE | Public | None |
| (87) | MODULE\_TOOL\_ID | Public | None |
| (88) | VENDOR\_API\_INFIX | Public | None |
| (89) | DEFINITION\_REF | Public | None |
| (90) | PARAMETER\_VALUES | Public | None |
| (91) | REFERENCE\_VALUES | Public | None |
| (92) | UUID | Public | None |
| (93) | VALUE | Public | None |
| (94) | VALUE\_REF | Public | None |
| (95) | DEST | Public | None |
| (96) | ECUC\_BOOLEAN\_PARAM\_DEF | Public | None |
| (97) | ECUC\_FLOAT\_PARAM\_DEF | Public | None |
| (98) | IMPLEMENT\_SHORT\_NAME | Public | None |
| (99) | IMPLEMENT\_PROGRAMMING\_LANGUAGE | Public | None |
| (100) | IMPLEMENT\_SW\_VERSION | Public | None |
| (101) | IMPLEMENT\_VENDOR\_ID | Public | None |
| (102) | IMPLEMENT\_AR\_RELEASE\_VERSION | Public | None |
| (103) | IMPLEMENT\_VENDOR\_API\_INFIX | Public | None |
| (104) | IMPLEMENT\_BEHAVIOR\_REF | Public | None |
| (105) | IMPLEMENT\_UUID | Public | None |
| (106) | DESCRIPTION\_SHORT\_NAME | Public | None |
| (107) | DESCRIPTION\_MODULE\_ID | Public | None |
| (108) | DESCRIPTION\_UUID | Public | None |
| (109) | IMPLEMENT\_VENDOR\_SPECIFIC\_MODULE\_DEF\_REFS | Public | None |
| (110) | MODULE\_DESCRIPTION\_REF | Public | None |
| (111) | NULL\_PTR | Public | None |
| (112) | DEM\_CONF\_DEM\_EVENT\_PARAMETER | Public | None |
| (113) | VERSION\_INFORMATION\_CFG | Public | None |
| (114) | VERSION\_INFORMATION\_CBK | Public | None |
| (115) | VERSION\_INFORMATION\_PBCFG | Public | None |
| (116) | VERSION\_INFORMATION\_LT | Public | None |
| (117) | CFG\_AR\_RELEASE\_MAJOR\_VERSION | Public | None |
| (118) | CFG\_AR\_RELEASE\_MINOR\_VERSION | Public | None |
| (119) | CFG\_AR\_RELEASE\_REVISION\_VERSION | Public | None |
| (120) | CFG\_SW\_MAJOR\_VERSION | Public | None |
| (121) | CFG\_SW\_MINOR\_VERSION | Public | None |
| (122) | AR\_RELEASE\_MAJOR\_VERSION\_VALUE | Public | None |
| (123) | AR\_RELEASE\_MINOR\_VERSION\_VALUE | Public | None |
| (124) | AR\_RELEASE\_REVISION\_VERSION\_VALUE | Public | None |
| (125) | SW\_MAJOR\_VERSION\_VALUE | Public | None |
| (126) | SW\_MINOR\_VERSION\_VALUE | Public | None |
| (127) | SW\_PATCH\_VERSION\_VALUE | Public | None |
| (128) | VENDOR\_ID\_VALUE | Public | None |
| (129) | MODULE\_ID\_VALUE | Public | None |
| (130) | CBK\_AR\_RELEASE\_MAJOR\_VERSION | Public | None |
| (131) | CBK\_AR\_RELEASE\_MINOR\_VERSION | Public | None |
| (132) | CBK\_AR\_RELEASE\_REVISION\_VERSION | Public | None |
| (133) | CBK\_SW\_MAJOR\_VERSION | Public | None |
| (134) | CBK\_SW\_MINOR\_VERSION | Public | None |
| (135) | PBCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION | Public | None |
| (136) | PBCFG\_C\_AR\_RELEASE\_MINOR\_VERSION | Public | None |
| (137) | PBCFG\_C\_AR\_RELEASE\_REVISION\_VERSION | Public | None |
| (138) | PBCFG\_C\_SW\_MAJOR\_VERSION | Public | None |
| (139) | PBCFG\_C\_SW\_MINOR\_VERSION | Public | None |
| (140) | LCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION | Public | None |
| (141) | LCFG\_C\_AR\_RELEASE\_MINOR\_VERSION | Public | None |
| (142) | LCFG\_C\_AR\_RELEASE\_REVISION\_VERSION | Public | None |
| (143) | LCFG\_C\_SW\_MAJOR\_VERSION | Public | None |
| (144) | LCFG\_C\_SW\_MINOR\_VERSION | Public | None |
| (145) | LINK\_TIME | Public | None |
| (146) | CALL\_BACK | Public | None |
| (147) | POST\_BUILD | Public | None |
| (148) | DeviceNameD1xRSCAN | Public | None |
| (149) | DeviceNameD1xCANFD | Public | None |
| (150) | DEVICE\_NAME\_D1S1L | Public | None |
| (151) | DEVICE\_NAME\_BL2M\_100 | Public | None |
| (152) | DEVICE\_NAME\_BL3M\_144 | Public | None |
| (153) | DEVICE\_NAME\_MLEV2\_4M\_272 | Public | None |

1. Method List

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

#### Fields

##### ON

GENERIC\_GDD\_CLS\_032\_001:

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

##### OFF

GENERIC\_GDD\_CLS\_032\_002:

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

##### STD\_ON

GENERIC\_GDD\_CLS\_032\_003:

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

##### STD\_OFF

GENERIC\_GDD\_CLS\_032\_004:

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

##### ECUC\_CONTAINER\_VALUE

GENERIC\_GDD\_CLS\_032\_005:

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

##### CONTAINERS

GENERIC\_GDD\_CLS\_032\_006:

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

##### SUB\_CONTAINERS

GENERIC\_GDD\_CLS\_032\_007:

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

##### ECUC\_MODULE\_CONFIGURATION\_VALUES

GENERIC\_GDD\_CLS\_032\_008:

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

##### BSW\_IMPLEMENTATION

GENERIC\_GDD\_CLS\_032\_009:

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

##### BSW\_MODULE\_DESCRIPTION

GENERIC\_GDD\_CLS\_032\_010:

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

##### TRANSLATION\_FILE\_PATH

GENERIC\_GDD\_CLS\_032\_011:

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

##### DEVICE\_FILE\_PATH

GENERIC\_GDD\_CLS\_032\_012:

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

##### TWO

GENERIC\_GDD\_CLS\_032\_013:

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

##### ONE

GENERIC\_GDD\_CLS\_032\_014:

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

##### ZERO

GENERIC\_GDD\_CLS\_032\_015:

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

##### TRUE

GENERIC\_GDD\_CLS\_032\_016:

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

##### FALSE

GENERIC\_GDD\_CLS\_032\_017:

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

##### IOREG

GENERIC\_GDD\_CLS\_032\_018:

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

##### OPTION

GENERIC\_GDD\_CLS\_032\_019:

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

##### INPUT\_FILE

GENERIC\_GDD\_CLS\_032\_020:

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

##### AR\_RELEASE\_VERSION

GENERIC\_GDD\_CLS\_032\_021:

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

##### SW\_VERSION

GENERIC\_GDD\_CLS\_032\_022:

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

##### VENDOR\_ID

GENERIC\_GDD\_CLS\_032\_023:

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

##### MODULE\_ID

GENERIC\_GDD\_CLS\_032\_024:

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

##### EXIT\_DUE\_CMD\_ERROR

GENERIC\_GDD\_CLS\_032\_025:

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

##### EXIT\_SUCCESSFUL

GENERIC\_GDD\_CLS\_032\_026:

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

##### EXIT\_SUCCESSFUL\_WITH\_WARNINGS

GENERIC\_GDD\_CLS\_032\_027:

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

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

GENERIC\_GDD\_CLS\_032\_028:

|  |  |
| --- | --- |
| **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\_032\_029:

|  |  |
| --- | --- |
| **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\_032\_030:

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

##### EXIT\_DUE\_UNCOMPATIBLE\_DLL

GENERIC\_GDD\_CLS\_032\_031:

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

##### EXIT\_DUE\_CRASH

GENERIC\_GDD\_CLS\_032\_032:

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

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

GENERIC\_GDD\_CLS\_032\_033:

|  |  |
| --- | --- |
| **Variable Name** | EXIT\_DUE\_DLL\_NOT\_FOUND |
| **Type** | const int |
| **Range** | None |
| **Description** | Exit due to DLL file not found |

##### AR\_PACKAGE

GENERIC\_GDD\_CLS\_032\_034:

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

##### AR\_PACKAGES

GENERIC\_GDD\_CLS\_032\_035:

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

##### ELEMENTS

GENERIC\_GDD\_CLS\_032\_036:

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

##### SHORT\_NAME

GENERIC\_GDD\_CLS\_032\_037:

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

##### ECUC\_VALUE\_COLLECTION

GENERIC\_GDD\_CLS\_032\_038:

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

##### BOOL\_DATA\_TYPE\_MARKED

GENERIC\_GDD\_CLS\_032\_039:

|  |  |
| --- | --- |
| **Variable Name** | BOOL\_DATA\_TYPE\_MARKED |
| **Type** | const string |
| **Range** | None |
| **Description** | Mark for ECUC BOOLEAN PARAM DEF attribute CDF |

##### FLOAT\_DATA\_TYPE\_MARKED

GENERIC\_GDD\_CLS\_032\_040:

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

##### ECUC\_INTEGER\_PARAM\_DEF

GENERIC\_GDD\_CLS\_032\_041:

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

##### ECUC\_STRING\_PARAM\_DEF

GENERIC\_GDD\_CLS\_032\_042:

|  |  |
| --- | --- |
| **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\_032\_043:

|  |  |
| --- | --- |
| **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\_032\_044:

|  |  |
| --- | --- |
| **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\_032\_045:

|  |  |
| --- | --- |
| **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\_032\_046:

|  |  |
| --- | --- |
| **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\_032\_047:

|  |  |
| --- | --- |
| **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\_032\_048:

|  |  |
| --- | --- |
| **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\_032\_049:

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

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

GENERIC\_GDD\_CLS\_032\_050:

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

##### AR\_VERSION\_4\_0\_3

GENERIC\_GDD\_CLS\_032\_051:

|  |  |
| --- | --- |
| **Variable Name** | AR\_VERSION\_4\_0\_3 |
| **Type** | const string |
| **Range** | None |
| **Description** | AutoSar Version 4.0.3 |

##### TOOL\_VERSION

GENERIC\_GDD\_CLS\_032\_052:

|  |  |
| --- | --- |
| **Variable Name** | TOOL\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | Generation tool version control |

##### ALL\_OBJECT

GENERIC\_GDD\_CLS\_032\_053:

|  |  |
| --- | --- |
| **Variable Name** | ALL\_OBJECT |
| **Type** | const string |
| **Range** | None |
| **Description** | Select all object related to Autosar Version and Device Name. |

##### DEVICE\_NAME\_D1L

GENERIC\_GDD\_CLS\_032\_054:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1L |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1L family. |

##### DEVICE\_NAME\_D1L1

GENERIC\_GDD\_CLS\_032\_055:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1L1 |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1L1 series. |

##### DEVICE\_NAME\_D1L2

GENERIC\_GDD\_CLS\_032\_056:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1L2 |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1L2 series. |

##### DEVICE\_NAME\_D1L2H

GENERIC\_GDD\_CLS\_032\_057:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1L2H |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1L2H series. |

##### DEVICE\_NAME\_D1M

GENERIC\_GDD\_CLS\_032\_058:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1M |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1M family. |

##### DEVICE\_NAME\_D1M1

GENERIC\_GDD\_CLS\_032\_059:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1M1 |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1M1 series. |

##### DEVICE\_NAME\_D1M1H

GENERIC\_GDD\_CLS\_032\_060:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1M1H |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1M1H series. |

##### DEVICE\_NAME\_D1M1V2

GENERIC\_GDD\_CLS\_032\_061:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1M1V2 |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1M1V2 series. |

##### DEVICE\_NAME\_D1M1A

GENERIC\_GDD\_CLS\_032\_062:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1M1A |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1M1A series. |

##### DEVICE\_NAME\_D1M2

GENERIC\_GDD\_CLS\_032\_063:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1M2 |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1M2 series. |

##### DEVICE\_NAME\_D1M2H

GENERIC\_GDD\_CLS\_032\_064:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1M2H |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1M2H series. |

##### DEVICE\_NAME\_D1S

GENERIC\_GDD\_CLS\_032\_065:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1S |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1S family. |

##### DEVICE\_NAME\_D1S1

GENERIC\_GDD\_CLS\_032\_066:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1S1 |
| **Type** | const string |
| **Range** | None |
| **Description** | List of device name D1S1 series. |

##### ModuleOption

GENERIC\_GDD\_CLS\_032\_067:

|  |  |
| --- | --- |
| **Variable Name** | ModuleOption |
| **Type** | readonly string[] |
| **Range** | None |
| **Description** | Option of module such as -m, -module |

##### ConfigFileOption

GENERIC\_GDD\_CLS\_032\_068:

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

##### OutputOption

GENERIC\_GDD\_CLS\_032\_069:

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

##### FilterOption

GENERIC\_GDD\_CLS\_032\_070:

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

##### HelpOption

GENERIC\_GDD\_CLS\_032\_071:

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

##### DryrunOption

GENERIC\_GDD\_CLS\_032\_072:

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

##### LogOption

GENERIC\_GDD\_CLS\_032\_073:

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

##### OutputIncSrcOption

GENERIC\_GDD\_CLS\_032\_074:

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

##### FILTER\_RENESAS

GENERIC\_GDD\_CLS\_032\_075:

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

##### OUTPUT

GENERIC\_GDD\_CLS\_032\_076:

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

##### FILTER

GENERIC\_GDD\_CLS\_032\_077:

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

##### OUTPUT\_PATH

GENERIC\_GDD\_CLS\_032\_078:

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

##### ON\_CONST

GENERIC\_GDD\_CLS\_032\_079:

|  |  |
| --- | --- |
| **Variable Name** | ON\_CONST |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### OFF\_CONST

GENERIC\_GDD\_CLS\_032\_080:

|  |  |
| --- | --- |
| **Variable Name** | OFF\_CONST |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### FILTER\_NAME

GENERIC\_GDD\_CLS\_032\_081:

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

##### Extension

GENERIC\_GDD\_CLS\_032\_082:

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

##### MODULE\_NAME

GENERIC\_GDD\_CLS\_032\_083:

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

##### MICROFAMILY\_NAME

GENERIC\_GDD\_CLS\_032\_084:

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

##### MODULE\_REQUIRED

GENERIC\_GDD\_CLS\_032\_085:

|  |  |
| --- | --- |
| **Variable Name** | MODULE\_REQUIRED |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### PROJECT\_TITLE

GENERIC\_GDD\_CLS\_032\_086:

|  |  |
| --- | --- |
| **Variable Name** | PROJECT\_TITLE |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### MODULE\_TOOL\_ID

GENERIC\_GDD\_CLS\_032\_087:

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

##### VENDOR\_API\_INFIX

GENERIC\_GDD\_CLS\_032\_088:

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

##### DEFINITION\_REF

GENERIC\_GDD\_CLS\_032\_089:

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

##### PARAMETER\_VALUES

GENERIC\_GDD\_CLS\_032\_090:

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

##### REFERENCE\_VALUES

GENERIC\_GDD\_CLS\_032\_091:

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

##### UUID

GENERIC\_GDD\_CLS\_032\_092:

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

##### VALUE

GENERIC\_GDD\_CLS\_032\_093:

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

##### VALUE\_REF

GENERIC\_GDD\_CLS\_032\_094:

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

##### DEST

GENERIC\_GDD\_CLS\_032\_095:

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

##### ECUC\_BOOLEAN\_PARAM\_DEF

GENERIC\_GDD\_CLS\_032\_096:

|  |  |
| --- | --- |
| **Variable Name** | ECUC\_BOOLEAN\_PARAM\_DEF |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### ECUC\_FLOAT\_PARAM\_DEF

GENERIC\_GDD\_CLS\_032\_097:

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

##### IMPLEMENT\_SHORT\_NAME

GENERIC\_GDD\_CLS\_032\_098:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_SHORT\_NAME |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### IMPLEMENT\_PROGRAMMING\_LANGUAGE

GENERIC\_GDD\_CLS\_032\_099:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_PROGRAMMING\_LANGUAGE |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### IMPLEMENT\_SW\_VERSION

GENERIC\_GDD\_CLS\_032\_100:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_SW\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### IMPLEMENT\_VENDOR\_ID

GENERIC\_GDD\_CLS\_032\_101:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_VENDOR\_ID |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### IMPLEMENT\_AR\_RELEASE\_VERSION

GENERIC\_GDD\_CLS\_032\_102:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_AR\_RELEASE\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### IMPLEMENT\_VENDOR\_API\_INFIX

GENERIC\_GDD\_CLS\_032\_103:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_VENDOR\_API\_INFIX |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### IMPLEMENT\_BEHAVIOR\_REF

GENERIC\_GDD\_CLS\_032\_104:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_BEHAVIOR\_REF |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### IMPLEMENT\_UUID

GENERIC\_GDD\_CLS\_032\_105:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_UUID |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DESCRIPTION\_SHORT\_NAME

GENERIC\_GDD\_CLS\_032\_106:

|  |  |
| --- | --- |
| **Variable Name** | DESCRIPTION\_SHORT\_NAME |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DESCRIPTION\_MODULE\_ID

GENERIC\_GDD\_CLS\_032\_107:

|  |  |
| --- | --- |
| **Variable Name** | DESCRIPTION\_MODULE\_ID |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DESCRIPTION\_UUID

GENERIC\_GDD\_CLS\_032\_108:

|  |  |
| --- | --- |
| **Variable Name** | DESCRIPTION\_UUID |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### IMPLEMENT\_VENDOR\_SPECIFIC\_MODULE\_DEF\_REFS

GENERIC\_GDD\_CLS\_032\_109:

|  |  |
| --- | --- |
| **Variable Name** | IMPLEMENT\_VENDOR\_SPECIFIC\_MODULE\_DEF\_REFS |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### MODULE\_DESCRIPTION\_REF

GENERIC\_GDD\_CLS\_032\_110:

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

##### NULL\_PTR

GENERIC\_GDD\_CLS\_032\_111:

|  |  |
| --- | --- |
| **Variable Name** | NULL\_PTR |
| **Type** | readonly string[] |
| **Range** | None |
| **Description** | None |

##### DEM\_CONF\_DEM\_EVENT\_PARAMETER

GENERIC\_GDD\_CLS\_032\_112:

|  |  |
| --- | --- |
| **Variable Name** | DEM\_CONF\_DEM\_EVENT\_PARAMETER |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### VERSION\_INFORMATION\_CFG

GENERIC\_GDD\_CLS\_032\_113:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_CFG |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### VERSION\_INFORMATION\_CBK

GENERIC\_GDD\_CLS\_032\_114:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_CBK |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### VERSION\_INFORMATION\_PBCFG

GENERIC\_GDD\_CLS\_032\_115:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_PBCFG |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### VERSION\_INFORMATION\_LT

GENERIC\_GDD\_CLS\_032\_116:

|  |  |
| --- | --- |
| **Variable Name** | VERSION\_INFORMATION\_LT |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CFG\_AR\_RELEASE\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_032\_117:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_AR\_RELEASE\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CFG\_AR\_RELEASE\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_032\_118:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_AR\_RELEASE\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CFG\_AR\_RELEASE\_REVISION\_VERSION

GENERIC\_GDD\_CLS\_032\_119:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_AR\_RELEASE\_REVISION\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CFG\_SW\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_032\_120:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_SW\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CFG\_SW\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_032\_121:

|  |  |
| --- | --- |
| **Variable Name** | CFG\_SW\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### AR\_RELEASE\_MAJOR\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_032\_122:

|  |  |
| --- | --- |
| **Variable Name** | AR\_RELEASE\_MAJOR\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### AR\_RELEASE\_MINOR\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_032\_123:

|  |  |
| --- | --- |
| **Variable Name** | AR\_RELEASE\_MINOR\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### AR\_RELEASE\_REVISION\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_032\_124:

|  |  |
| --- | --- |
| **Variable Name** | AR\_RELEASE\_REVISION\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### SW\_MAJOR\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_032\_125:

|  |  |
| --- | --- |
| **Variable Name** | SW\_MAJOR\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### SW\_MINOR\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_032\_126:

|  |  |
| --- | --- |
| **Variable Name** | SW\_MINOR\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### SW\_PATCH\_VERSION\_VALUE

GENERIC\_GDD\_CLS\_032\_127:

|  |  |
| --- | --- |
| **Variable Name** | SW\_PATCH\_VERSION\_VALUE |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### VENDOR\_ID\_VALUE

GENERIC\_GDD\_CLS\_032\_128:

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

##### MODULE\_ID\_VALUE

GENERIC\_GDD\_CLS\_032\_129:

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

##### CBK\_AR\_RELEASE\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_032\_130:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_AR\_RELEASE\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CBK\_AR\_RELEASE\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_032\_131:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_AR\_RELEASE\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CBK\_AR\_RELEASE\_REVISION\_VERSION

GENERIC\_GDD\_CLS\_032\_132:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_AR\_RELEASE\_REVISION\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CBK\_SW\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_032\_133:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_SW\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CBK\_SW\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_032\_134:

|  |  |
| --- | --- |
| **Variable Name** | CBK\_SW\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### PBCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_032\_135:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### PBCFG\_C\_AR\_RELEASE\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_032\_136:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_AR\_RELEASE\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### PBCFG\_C\_AR\_RELEASE\_REVISION\_VERSION

GENERIC\_GDD\_CLS\_032\_137:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_AR\_RELEASE\_REVISION\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### PBCFG\_C\_SW\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_032\_138:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_SW\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### PBCFG\_C\_SW\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_032\_139:

|  |  |
| --- | --- |
| **Variable Name** | PBCFG\_C\_SW\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### LCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_032\_140:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_AR\_RELEASE\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### LCFG\_C\_AR\_RELEASE\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_032\_141:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_AR\_RELEASE\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### LCFG\_C\_AR\_RELEASE\_REVISION\_VERSION

GENERIC\_GDD\_CLS\_032\_142:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_AR\_RELEASE\_REVISION\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### LCFG\_C\_SW\_MAJOR\_VERSION

GENERIC\_GDD\_CLS\_032\_143:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_SW\_MAJOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### LCFG\_C\_SW\_MINOR\_VERSION

GENERIC\_GDD\_CLS\_032\_144:

|  |  |
| --- | --- |
| **Variable Name** | LCFG\_C\_SW\_MINOR\_VERSION |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### LINK\_TIME

GENERIC\_GDD\_CLS\_032\_145:

|  |  |
| --- | --- |
| **Variable Name** | LINK\_TIME |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### CALL\_BACK

GENERIC\_GDD\_CLS\_032\_146:

|  |  |
| --- | --- |
| **Variable Name** | CALL\_BACK |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### POST\_BUILD

GENERIC\_GDD\_CLS\_032\_147:

|  |  |
| --- | --- |
| **Variable Name** | POST\_BUILD |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DeviceNameD1xRSCAN

GENERIC\_GDD\_CLS\_032\_148:

|  |  |
| --- | --- |
| **Variable Name** | DeviceNameD1xRSCAN |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DeviceNameD1xCANFD

GENERIC\_GDD\_CLS\_032\_149:

|  |  |
| --- | --- |
| **Variable Name** | DeviceNameD1xCANFD |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DEVICE\_NAME\_D1S1L

GENERIC\_GDD\_CLS\_032\_150:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_D1S1L |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DEVICE\_NAME\_BL2M\_100

GENERIC\_GDD\_CLS\_032\_151:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_BL2M\_100 |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DEVICE\_NAME\_BL3M\_144

GENERIC\_GDD\_CLS\_032\_152:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_BL3M\_144 |
| **Type** | const string |
| **Range** | None |
| **Description** | None |

##### DEVICE\_NAME\_MLEV2\_4M\_272

GENERIC\_GDD\_CLS\_032\_153:

|  |  |
| --- | --- |
| **Variable Name** | DEVICE\_NAME\_MLEV2\_4M\_272 |
| **Type** | const string |
| **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) | IsNameSyntaxAsC | Public | Check whether variable name follow to C coding rule or not. |
| (11) | DoubleToString | Public | This method is used to convert double to string and ignore CultureInfo |
| (12) | IsNullPointer | Public | Check whether value is null or not |
| (13) | replaceStringByTemplate | Private | Replace placeholder in a string |
| (14) | AppendDirectorySeparatorChar | Private | Append directory separator "\" to a path |
| (15) | parseShortName | Private | Parse short name in CDF to 2 parts: name and surfix number |

#### Fields

#### Methods

##### CompareShortName

GENERIC\_GDD\_CLS\_033\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | CompareShortName | | |
| **Arguments** | string shrName1 | short name for comparation | |
| Range: | None |
| string shrName2 | short name for comparation | |
| Range: | None |
| **Return** | compration result | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Compare 2 short name for sorting  Algorithm: Parse short name into name and number  IF names are the same THEN compare numbers | | |

{Ref: N/A}

##### DoubleToString

GENERIC\_GDD\_CLS\_033\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | DoubleToString | | |
| **Arguments** | double value | value of double type | |
| Range: | None |
| **Return** | coverted string without culture | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | This method is used to convert double to string and ignore CultureInfo  Algorithm: Use option CultureInfo.InvariantCulture to ignore culture when covert double to string. | | |

{Ref: N/A}

##### FindString

GENERIC\_GDD\_CLS\_033\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | FindString | | |
| **Arguments** | string text | Text | |
| Range: | None |
| string expression | Expression | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Find substring in text that match a regular expression  Algorithm: IF text or expr is NULL or empty THEN  Return emtpy  End   Scan text  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\_033\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetElementLastInArray | | |
| **Arguments** | string path | Path | |
| Range: | None |
| char delimeter | Delimeter | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get last element when spliting path into substrings by delimeter  Algorithm: Split path into substrings by delimeter.  RETURN last substring. | | |

{Ref: N/A}

##### GetRelativePath

GENERIC\_GDD\_CLS\_033\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetRelativePath | | |
| **Arguments** | string toPath | To path | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get relative path from current directory  Algorithm: Get relative path from current directory  Replace all "/" by "\" | | |

{Ref: GENERIC\_TFD\_X2x\_GT\_USG\_014}

##### GetStringFromTemplate

GENERIC\_GDD\_CLS\_033\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetStringFromTemplate | | |
| **Arguments** | string inputString | Input string | |
| Range: | None |
| Dictionary< string, string > dictionary | Mapping placeholders and their values | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Replace placeholders in a string  Algorithm: Scan each placeholder in dictionary  Replace placeholder in inputString by its value defined in dictionary  End | | |

{Ref: N/A}

##### IsHex

GENERIC\_GDD\_CLS\_033\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsHex | | |
| **Arguments** | string value | Value | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check if string is in hexa format  Algorithm: Using regular expression to check if string is in hexa format | | |

{Ref: N/A}

##### IsInvalidName

GENERIC\_GDD\_CLS\_033\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsInvalidName | | |
| **Arguments** | string name | Name | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check invalid name  Algorithm: Invalid name when  - name is null or empty  - name starts with "-"  - name contains "\*", "?" or "|" | | |

{Ref: N/A}

##### IsNameSyntaxAsC

GENERIC\_GDD\_CLS\_033\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsNameSyntaxAsC | | |
| **Arguments** | string name | A variable name | |
| Range: | None |
| **Return** | bool | None | |
| Range: | true/false |
| **Generated Value** | None | | |
| **Description** | Check whether variable name follow to C coding rule or not.  Algorithm: Name of variable have to follow to C coding rule. | | |

{Ref: N/A}

##### IsNullPointer

GENERIC\_GDD\_CLS\_033\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsNullPointer | | |
| **Arguments** | string value | value need checking | |
| Range: | None |
| **Return** | true if value is null, else false | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Check whether value is null or not  Algorithm: Check whether value is null or not by compare NULL, NULL\_PTR or empty | | |

{Ref: N/A}

##### ParseHexToInt

GENERIC\_GDD\_CLS\_033\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseHexToInt | | |
| **Arguments** | string inputString | Input string | |
| Range: | None |
| **Return** | return a int | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Convert hex string to integer  Algorithm: Upper case inputString  Remove all "0x", "0X", "U" and "L" characters from inputString  Using C# util function to convert inputString to integer value | | |

{Ref: N/A}

##### ParseHexToUInt

GENERIC\_GDD\_CLS\_033\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ParseHexToUInt | | |
| **Arguments** | string inputString | Input string | |
| Range: | None |
| **Return** | return a int | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Convert hex string to unsigned integer  Algorithm: Upper case inputString  Remove all "0x", "0X", "U" and "L" characters from inputString  Using C# util function to convert inputString to unsigned integer value | | |

{Ref: N/A}

##### AppendDirectorySeparatorChar

GENERIC\_GDD\_CLS\_033\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AppendDirectorySeparatorChar | | |
| **Arguments** | string path | Path | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Append directory separator "\" to a path  Algorithm: Append directory separator "\" to a path when  - path doesn't have extension at the end  - path doesn't have directory separator "\" at the end | | |

{Ref: N/A}

##### parseShortName

GENERIC\_GDD\_CLS\_033\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | parseShortName | | |
| **Arguments** | string shrName | short name in CDF | |
| Range: | None |
| out string name | name of parameter | |
| Range: | None |
| out int number | surfix number | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Parse short name in CDF to 2 parts: name and surfix number  Algorithm: Parse short name in CDF to 2 parts: name and surfix number  IF surfix number is empty THEN set it as zero | | |

{Ref: N/A}

##### replaceStringByTemplate

GENERIC\_GDD\_CLS\_033\_015:

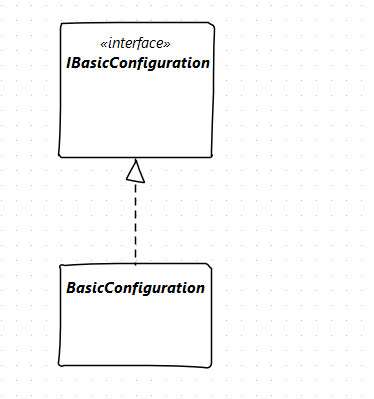
|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | replaceStringByTemplate | | |
| **Arguments** | string inputString | Dictionary | |
| Range: | None |
| Dictionary< string, string > dictionary | inputString | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Replace placeholder in a string  Algorithm: inputString has format $${placeholder}:{fix\_length\_after\_placeholder}:{alignment}:{suffix}$$  RETURN a string by following steps  - Result = Replace placeholder by its value defined in dictionary  - Result = Result + suffix  - Add more spaces to Result so that: number of space + suffix length = fix\_length\_after\_placeholder  - Align Result to left, right or center based on alignment value (L, R, C) | | |

{Ref: N/A}

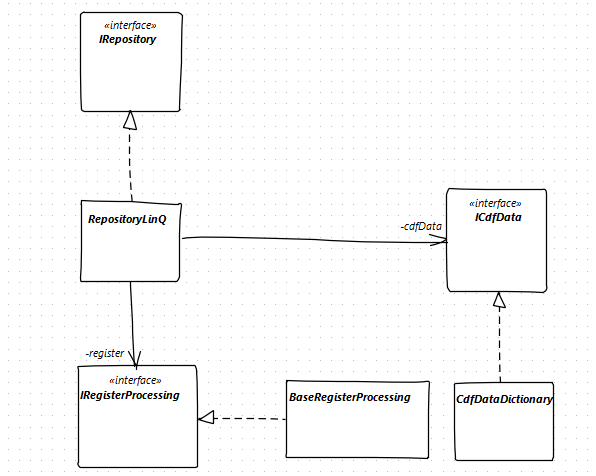
## Data

This component define data structure of Generation Tool to manage input data from user such as command line options, cdf, bswmdt, translation files.

### UML



The BasicConfiguration class define a data structure to handle information of MCAL module like module id, device name, module name, dependency modules, micro family name,…

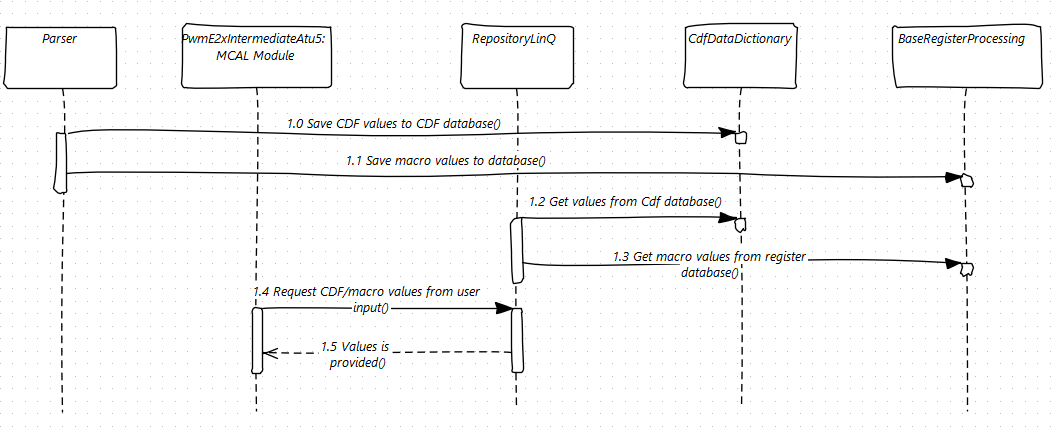


The CdfDataDictionary class define a database that contains CDF values that is parsed by Parser component. Afterward, the RepositoryLinQ class can access those values via provided API.

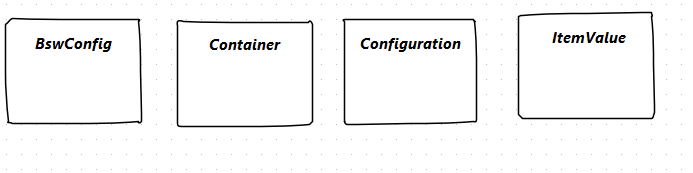
The BaseRegisterProcessing class circumscribe a database to handle macro values that is parsed from Parser component. Those values can also be accessed by RepositoryLinQ via supplied API.

The RepositoryLinQ class provide APIs to MCAL modules like getting containers in CDF file, getting child containers of container, getting parameter values of containers, getting marco values in translation files,….Those APIs help to reduce effort accessing data from user input regarding MCAL modules.

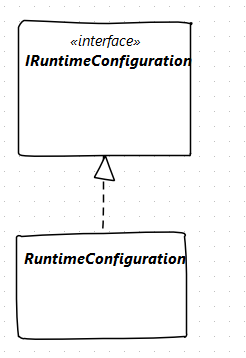
Below is sample to describe data flows between related classes.



Some classes that define structures are used to manage data in CDF files



|  |  |  |
| --- | --- | --- |
| Class | Description | Example |
| BswConfig | Store required values in BSWMDT file such as vendor id, sw version, ar release version | <SW-VERSION>**1.2.10**</SW-VERSION>  <VENDOR-ID>**59**</VENDOR-ID>  <AR-RELEASE-VERSION>**4.2.2**</AR-RELEASE-VERSION> |
| Container | Store container information in CDF file such as short name, definition ref, parameter values, reference values | <ECUC-CONTAINER-VALUE UUID="7fdb4340-43f0-4b76-b8bb-8a1ed5cb0e20">  <SHORT-NAME>**LinGeneral**</SHORT-NAME>  <DEFINITION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/**Renesas/EcucDefs\_Lin/Lin/LinGeneral**</DEFINITION-REF>  <PARAMETER-VALUES/>  </ECUC-CONTAINER-VALUE> |
| Configuration | Store configuration each module in CDF file. | <DEFINITION-REF DEST="ECUC-MODULE-DEF**">/Renesas/EcucDefs\_Lin/Lin</**DEFINITION-REF>  <IMPLEMENTATION-CONFIG-VARIANT>VARIANT-POST-BUILD</IMPLEMENTATION-CONFIG-VARIANT>  <MODULE-DESCRIPTION-REF DEST="BSW-IMPLEMENTATION**">/Renesas/EcucDefs\_Lin/Lin\_Impl**</MODULE-DESCRIPTION-REF>  <CONTAINERS> |
| ItemValue | Store a parameter value or a reference value in a container like definition ref and value. | <ECUC-NUMERICAL-PARAM-VALUE>  <DEFINITION-REF DEST="ECUC-INTEGER-PARAM-DEF">/**Renesas/EcucDefs\_Lin/Lin/LinGlobalConfig/LinChannel/LinChannelBaudRate**</DEFINITION-REF>  <VALUE>**9600**</VALUE>  </ECUC-NUMERICAL-PARAM-VALUE> |



The RuntimeConfiguration class contains valid command line options such as -h, -o, -m, -l,.. and file paths like translation file path, cdf file path.

### Renesas::Generator::MCALConfGen::Data::BasicConfiguration::BasicConfiguration

|  |  |
| --- | --- |
| **Location File Name** | BasicConfiguration.cs |
| **Description** | This class implements IBasicConfiguration interface to store basic information of Gentool. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | BasicConfiguration instance used by ObjectFactory to get a new BasicConfiguration object |
| (2) | ModuleName | Public | Contain MCAL module name such as Adc, Can, Dio, Lin,... |
| (3) | MicroFamilyName | Public | Contain micro family name such as E2x, U2x, D1x,... |
| (4) | AutoSARVersion | Public | Contain AutoSAR version such as 4.2.2, 4.0.3,... |
| (5) | DeviceNames | Public | Contain device names information such as R7F701421, R7F701402,... |
| (6) | ExeDirectory | Public | Contain executable directory such as D:\, E:\,... |
| (7) | ModuleId | Public | Contain module id as 80(Can) |
| (8) | OutputFolder | Public | Contain output folder path such as D:\output\ |
| (9) | ExecutionName | Public | Contain execution name as MCALConfGen.exe |
| (10) | ModuleRequired | Public | Contain modules required such as Dem, Os, Mcu,... |
| (11) | ProjectTitle | Public | Contain project title information |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | BasicConfiguration | Private | Basic configuration constructor without parameters |

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_034\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly BasicConfiguration |
| **Range** | None |
| **Description** | BasicConfiguration instance used by ObjectFactory to get a new BasicConfiguration object |

##### ModuleName

GENERIC\_GDD\_CLS\_034\_002:

|  |  |
| --- | --- |
| **Variable Name** | ModuleName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain MCAL module name such as Adc, Can, Dio, Lin,... |

##### MicroFamilyName

GENERIC\_GDD\_CLS\_034\_003:

|  |  |
| --- | --- |
| **Variable Name** | MicroFamilyName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain micro family name such as E2x, U2x, D1x,... |

##### AutoSARVersion

GENERIC\_GDD\_CLS\_034\_004:

|  |  |
| --- | --- |
| **Variable Name** | AutoSARVersion |
| **Type** | string |
| **Range** | None |
| **Description** | Contain AutoSAR version such as 4.2.2, 4.0.3,... |

##### DeviceNames

GENERIC\_GDD\_CLS\_034\_005:

|  |  |
| --- | --- |
| **Variable Name** | DeviceNames |
| **Type** | List< string > |
| **Range** | None |
| **Description** | Contain device names information such as R7F701421, R7F701402,... |

##### ExeDirectory

GENERIC\_GDD\_CLS\_034\_006:

|  |  |
| --- | --- |
| **Variable Name** | ExeDirectory |
| **Type** | string |
| **Range** | None |
| **Description** | Contain executable directory such as D:\, E:\,... |

##### ModuleId

GENERIC\_GDD\_CLS\_034\_007:

|  |  |
| --- | --- |
| **Variable Name** | ModuleId |
| **Type** | int |
| **Range** | None |
| **Description** | Contain module id as 80(Can) |

##### OutputFolder

GENERIC\_GDD\_CLS\_034\_008:

|  |  |
| --- | --- |
| **Variable Name** | OutputFolder |
| **Type** | string |
| **Range** | None |
| **Description** | Contain output folder path such as D:\output\ |

##### ExecutionName

GENERIC\_GDD\_CLS\_034\_009:

|  |  |
| --- | --- |
| **Variable Name** | ExecutionName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain execution name as MCALConfGen.exe |

##### ModuleRequired

GENERIC\_GDD\_CLS\_034\_010:

|  |  |
| --- | --- |
| **Variable Name** | ModuleRequired |
| **Type** | List< string > |
| **Range** | None |
| **Description** | Contain modules required such as Dem, Os, Mcu,... |

##### ProjectTitle

GENERIC\_GDD\_CLS\_034\_011:

|  |  |
| --- | --- |
| **Variable Name** | ProjectTitle |
| **Type** | string |
| **Range** | None |
| **Description** | Contain project title information |

#### Methods

##### BasicConfiguration

GENERIC\_GDD\_CLS\_034\_012:

|  |  |
| --- | --- |
| **Method Name** | BasicConfiguration |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Basic configuration constructor without parameters  Algorithm: Init values such as Module Id, AutoSARVersion, DeviceNames, ExeDirectory |

{Ref: N/A}

### Renesas::Generator::MCALConfGen::Data::CDFData::BaseRegisterProcessing

|  |  |
| --- | --- |
| **Location File Name** | BaseRegisterProcessing.cs |
| **Description** | This BaseRegisterProcessing class implements IRegisterProcessing interface to parses and stores information of translation and device header files. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | BaseRegisterProcessing instance used by ObjectFactory to get a new BaseRegisterProcessing object |
| (2) | macroLabelValue | Private | Contain definition macro include (macro - value) such as (RENESAS\_CSIG0TIR - INTC1ICCSIG0IR) |
| (3) | macroAddressValue | Private | Contain macro value include (macro value - address) such as (INTC1ICCSIG0IR - 0xfffeea38UL) |
| (4) | userInterface | Private | Contain user interface used to report error/message/info to user |
| (5) | translationHeaderFilePath | Private | Contain path of translation file that contain macro definition. |
| (6) | deviceHeaderFilePath | Private | Contain path of device file that contain address values. |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetMacroAddressValue | Public | Get macro address value based on definition macro value. |
| (2) | GetMacroLabelValue | Public | Get macro label value based on definition macro |
| (3) | IsRegisterExist | Public | Check whether definition macro does exist or not |
| (4) | SaveMacroAddressValue | Public | Parsing device file and saving macro address value |
| (5) | SaveMacroLabelValue | Public | Parsing translation file and save macro label value |
| (6) | BaseRegisterProcessing | Protected | Base register processing constructor without arguments |
| (7) | BaseRegisterProcessing | Protected | Base register processing constructor with argument |

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_035\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly BaseRegisterProcessing |
| **Range** | None |
| **Description** | BaseRegisterProcessing instance used by ObjectFactory to get a new BaseRegisterProcessing object |

##### macroLabelValue

GENERIC\_GDD\_CLS\_035\_002:

|  |  |
| --- | --- |
| **Variable Name** | macroLabelValue |
| **Type** | IDictionary< string, string > |
| **Range** | None |
| **Description** | Contain definition macro include (macro - value) such as (RENESAS\_CSIG0TIR - INTC1ICCSIG0IR) |

##### macroAddressValue

GENERIC\_GDD\_CLS\_035\_003:

|  |  |
| --- | --- |
| **Variable Name** | macroAddressValue |
| **Type** | IDictionary< string, string > |
| **Range** | None |
| **Description** | Contain macro value include (macro value - address) such as (INTC1ICCSIG0IR - 0xfffeea38UL) |

##### userInterface

GENERIC\_GDD\_CLS\_035\_004:

|  |  |
| --- | --- |
| **Variable Name** | userInterface |
| **Type** | IUserInterface |
| **Range** | None |
| **Description** | Contain user interface used to report error/message/info to user |

##### translationHeaderFilePath

GENERIC\_GDD\_CLS\_035\_012:

|  |  |
| --- | --- |
| **Variable Name** | translationHeaderFilePath |
| **Type** | String |
| **Range** | None |
| **Description** | Contain path of translation file that contain macro definition. |

##### deviceHeaderFilePath

GENERIC\_GDD\_CLS\_035\_013:

|  |  |
| --- | --- |
| **Variable Name** | deviceHeaderFilePath |
| **Type** | String |
| **Range** | None |
| **Description** | Contain path of device file that contain address values. |

#### Methods

##### GetMacroAddressValue

GENERIC\_GDD\_CLS\_035\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetMacroAddressValue | | |
| **Arguments** | string macroValue | Macro value | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get macro address value based on definition macro value.  Algorithm: IF macroValue NOT found THEN  Record error ERR000023 and exit  ELSE IF address of macro is null or empty THEN  Record error ERR000024 and exit  ELSE IF address of macro is not hexa value THEN  Record error ERR000038 and exit  ELSE  Return address of macro | | |

{Ref: [1] CMN\_GFD\_ERR\_023, CMN\_GFD\_ERR\_024, CMN\_GFD\_ERR\_038, GENERIC\_TFD\_X2x\_GT\_ERR\_023, GENERIC\_TFD\_X2x\_GT\_ERR\_024, GENERIC\_TFD\_X2x\_GT\_ERR\_038}

##### GetMacroLabelValue

GENERIC\_GDD\_CLS\_035\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetMacroLabelValue | | |
| **Arguments** | string macroDefinition | Macro definition | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get macro label value based on definition macro  Algorithm: If macroDefinition NOT found THEN  Record error ERR000021 and exit  ELSE IF value of macro is null or empty THEN  Record error ERR000022 and exit  ELSE  Return value of macro | | |

{Ref: [1] CMN\_GFD\_ERR\_021, CMN\_GFD\_ERR\_022, GENERIC\_TFD\_X2x\_GT\_ERR\_021, GENERIC\_TFD\_X2x\_GT\_ERR\_022}

##### IsRegisterExist

GENERIC\_GDD\_CLS\_035\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsRegisterExist | | |
| **Arguments** | string macroDefinition | Macro definition | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether definition macro does exist or not  Algorithm: IF definition macro found THEN  RETURN true  ELSE  RETURN false | | |

{Ref: N/A}

##### SaveMacroAddressValue

GENERIC\_GDD\_CLS\_035\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveMacroAddressValue | | |
| **Arguments** | string filesInput | File name | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing device file and saving macro address value  Algorithm: IF file NOT existed THEN  Report error ERR000001 and exit  ELSE  With each line in file  IF line NOT a comment THEN  Parsing macro value and address  Add macro value and address to collection macroAddressValue  End  End With  If have any address of definition macro duplication  Report to error ERR000037  end | | |

{Ref: [1] CMN\_GFD\_ERR\_001, CMN\_GFD\_ERR\_037, GENERIC\_TFD\_X2x\_GT\_ERR\_037}

##### SaveMacroLabelValue

GENERIC\_GDD\_CLS\_035\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveMacroLabelValue | | |
| **Arguments** | string fileName | File name | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Parsing translation file and save macro label value  Algorithm: IF file NOT existed THEN  Record error ERR000001 and exit  ELSE  FOREACH line in file  IF line NOT a comment THEN  Parsing macro label and value  Add macro label and value to collection macroLabelValue  End  End   IF have any definition macro duplication  Report error ERR000020 and exit  End | | |

{Ref: [1] CMN\_GFD\_ERR\_001, CMN\_GFD\_ERR\_020, GENERIC\_TFD\_X2x\_GT\_ERR\_020}

##### BaseRegisterProcessing

GENERIC\_GDD\_CLS\_035\_010:

|  |  |
| --- | --- |
| **Method Name** | BaseRegisterProcessing |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Base register processing constructor without arguments |

{Ref: N/A}

##### BaseRegisterProcessing

GENERIC\_GDD\_CLS\_035\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseRegisterProcessing | | |
| **Arguments** | userInterface | User interface | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base register processing constructor with argument  Algorithm: Init value for component user interface | | |

{Ref: N/A}

### Renesas::Generator::MCALConfGen::Data::CDFData::CdfDataDictionary

|  |  |
| --- | --- |
| **Location File Name** | CdfDataDictionary.cs |
| **Description** | This CdfDataDictionary class that implements ICdfData interface to save and retrive infomration of CDF files. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | CdfDataDictionary instance used by ObjectFactory to get a new CdfDataDictionary object |
| (2) | dataContext | Private | A 2-levels dictionary data context Module name 1 ------ |
| (3) | moduleConfig | Private | Mapping between 1 module ------ |
| (4) | configurations | Private | Contain all configurations of MCAL module |
| (5) | cdfFiles | Private | Contain data from CDF files as database |
| (6) | bswConfigs | Private | Contain basic software configuration of MCAL module as database |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Add | Public | Add a container of a module into database |
| (2) | GetContainers | Public | Get all containers of MCAL module |
| (3) | SaveConfiguration | Public | Save configuration in database with key as MCAL module name |
| (4) | GetConfiguration | Public | Get configuration by MCAL module name from database as moduleConfig |
| (5) | GetConfigurations | Public | Get all configurations by MCAL module name from database as moduleConfig |
| (6) | SaveBswConfiguration | Public | Store bswmdt configuration in database as bswConfigs |
| (7) | GetBswConfigurations | Public | Get all bsw configurations of MCAL module |
| (8) | GetBswConfiguration | Public | Get bsw configuration of MCAL module in database as bswConfigs |
| (9) | GetModules | Public | Get name of MCAL module and all dependence module such as Mcu, Dem, CanIf,... |
| (10) | SaveCdfFiles | Public | Store CDF files in database as cdfFiles |
| (11) | GetCdfFile | Public | Get CDF files from database as cdfFiles |
| (12) | GetContainerByUUID | Public | Get container by UUID from database as CdfData |
| (13) | AddModuleConfiguration | Public | Store module configuration in database as moduleConfig |
| (14) | CdfDataDictionary | Private | Cdf data dictionary constructor without arguments |
| (15) | getContainerByShortName | Private | Get container by short name |
| (16) | getModuleShortNameKeys | Private | Get list of module short name |

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_036\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly CdfDataDictionary |
| **Range** | None |
| **Description** | CdfDataDictionary instance used by ObjectFactory to get a new CdfDataDictionary object |

##### dataContext

GENERIC\_GDD\_CLS\_036\_002:

|  |  |
| --- | --- |
| **Variable Name** | dataContext |
| **Type** | Dictionary< string, Dictionary< string, Container |
| **Range** | None |
| **Description** | A 2-levels dictionary data context Module name 1 ------ |

##### moduleConfig

GENERIC\_GDD\_CLS\_036\_003:

|  |  |
| --- | --- |
| **Variable Name** | moduleConfig |
| **Type** | Dictionary< string, List< Configuration |
| **Range** | None |
| **Description** | Mapping between 1 module ------ |

##### configurations

GENERIC\_GDD\_CLS\_036\_004:

|  |  |
| --- | --- |
| **Variable Name** | configurations |
| **Type** | Dictionary< string, Configuration |
| **Range** | None |
| **Description** | Contain all configurations of MCAL module |

##### cdfFiles

GENERIC\_GDD\_CLS\_036\_005:

|  |  |
| --- | --- |
| **Variable Name** | cdfFiles |
| **Type** | Dictionary< string, string > |
| **Range** | None |
| **Description** | Contain data from CDF files as database |

##### bswConfigs

GENERIC\_GDD\_CLS\_036\_006:

|  |  |
| --- | --- |
| **Variable Name** | bswConfigs |
| **Type** | Dictionary< string, BswConfig |
| **Range** | None |
| **Description** | Contain basic software configuration of MCAL module as database |

#### Methods

##### Add

GENERIC\_GDD\_CLS\_036\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Add | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| container | Container | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Add a container of a module into database  Algorithm: 2-levels dictionary dataContext  Module name 1 --------- Container Short Name 1 --------- Container Object 1  +-------- Container Short Name 2 --------- Container Object 2  Module name 2 --------- Container Short Name 3 --------- Container Object 3  +-------- Container Short Name 4 --------- Container Object 4   Add to dataContext: moduleName + container's ShortName + container | | |

{Ref: N/A}

##### AddModuleConfiguration

GENERIC\_GDD\_CLS\_036\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AddModuleConfiguration | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| config | Configuration of MCAL module | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Store module configuration in database as moduleConfig  Algorithm: if have no any configuration in database as moduleConfig  Add configuration to database as moduleConfig  end  In case not exists any configuration in database  Init array of configuration in database with key as module name | | |

{Ref: N/A}

##### GetBswConfiguration

GENERIC\_GDD\_CLS\_036\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetBswConfiguration | | |
| **Arguments** | string moduleName | Module | |
| Range: | None |
| **Return** | return a BswConfig | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get bsw configuration of MCAL module in database as bswConfigs  Algorithm: IF module is null or empty THEN  IF have any config in bswConfigs THEN  RETURN first config  Else  RETURN null  ELSE  IF found moduleName in bswConfigs THEN  RETURN first found  ELSE  RETURN null | | |

{Ref: N/A}

##### GetBswConfigurations

GENERIC\_GDD\_CLS\_036\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetBswConfigurations | | |
| **Arguments** | None | | |
| **Return** | return a List of bsw configurations | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get all bsw configurations of MCAL module  Algorithm: RETURN all bsw configurations in database as bswConfigs | | |

{Ref: N/A}

##### GetCdfFile

GENERIC\_GDD\_CLS\_036\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetCdfFile | | |
| **Arguments** | string shortName | Short name | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get CDF files from database as cdfFiles  Algorithm: IF moduleName is NOT null or empty THEN  IF found shortName in cdfFiles THEN  RETURN found file  ELSE  RETURN empty  ELSE  RETURN empty | | |

{Ref: N/A}

##### GetConfiguration

GENERIC\_GDD\_CLS\_036\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetConfiguration | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return a Configuration | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get configuration by MCAL module name from database as moduleConfig  Algorithm: IF found moduleName in database as moduleConfig THEN  RETURN value from database  Else  RETURN null | | |

{Ref: N/A}

##### GetConfigurations

GENERIC\_GDD\_CLS\_036\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetConfigurations | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return a Configuration | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get all configurations by MCAL module name from database as moduleConfig  Algorithm: IF found moduleName in database as moduleConfig THEN  RETURN values from database  Else  RETURN null | | |

{Ref: N/A}

##### GetContainerByUUID

GENERIC\_GDD\_CLS\_036\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainerByUUID | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string uuid | Uuid | |
| Range: | None |
| **Return** | return a Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get container by UUID from database as CdfData  Algorithm: Scan all containers under moduleName  IF any of them has Uuid that equal to uuid THEN  RETURN first found container  End  End | | |

{Ref: N/A}

##### GetContainers

GENERIC\_GDD\_CLS\_036\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainers | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get all containers of MCAL module  Algorithm: IF moduleName is NOT null or empty THEN  Get all containers that are right under moduleName (or "{moduleName}d\*$")  to add to a containers collection "result"   FOREACH container in dataContext  Recursively get all of its children containers to add to "result"  End   RETURN "result"  ELLSE  FOREACH module name of dataContext  Get all containers that are right under moduleName (or "{moduleName}d\*$")  to add to a containers collection "result"   For each container  Recursively get all of its children containers to add to "result"  End  End   RETURN "result" | | |

{Ref: N/A}

##### GetModules

GENERIC\_GDD\_CLS\_036\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetModules | | |
| **Arguments** | None | | |
| **Return** | return List of string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get name of MCAL module and all dependence module such as Mcu, Dem, CanIf,...  Algorithm: Get name of MCAL module and all dependence module from database as moduleConfig | | |

{Ref: N/A}

##### SaveBswConfiguration

GENERIC\_GDD\_CLS\_036\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveBswConfiguration | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| config | Config | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Store bswmdt configuration in database as bswConfigs  Algorithm: IF moduleName is NOT null or empty and config is NOT null THEN  Add config to bswConfigs with key is moduleName  End | | |

{Ref: }

##### SaveCdfFiles

GENERIC\_GDD\_CLS\_036\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveCdfFiles | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string fileName | File name | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Store CDF files in database as cdfFiles  Algorithm: IF both moduleName and fileName are NOT null or empty THEN  Add fileName into database cdfFiles with key is moduleName  End | | |

{Ref: N/A}

##### SaveConfiguration

GENERIC\_GDD\_CLS\_036\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | SaveConfiguration | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| config | Config | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Save configuration in database with key as MCAL module name  Algorithm: IF moduleName is NOT null or empty and config is NOT null THEN  Save config into configurations with key moduleName  End | | |

{Ref: N/A}

##### CdfDataDictionary

GENERIC\_GDD\_CLS\_036\_020:

|  |  |
| --- | --- |
| **Method Name** | CdfDataDictionary |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Cdf data dictionary constructor without arguments |

{Ref: N/A}

##### getContainerByShortName

GENERIC\_GDD\_CLS\_036\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getContainerByShortName | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string shortName | Short name | |
| Range: | None |
| **Return** | return a Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get container by short name  Algorithm: Get containers of moduleName by calling method GetContainers  IF found 1 or more containers THEN  RETURN first container  ELSE  RETURN null | | |

{Ref: N/A}

##### getModuleShortNameKeys

GENERIC\_GDD\_CLS\_036\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getModuleShortNameKeys | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return List of string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get list of module short name  Algorithm: IF moduleName is NOT null or empty THEN  Find where moduleConfig contain key moduleName  to add to list of found module short name  Return "result"  Else  RETURN "result" | | |

{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 | Contain uuid of tag Implement in BSWMDT |
| (2) | ImplementSortName | Public | Contain short name of tag Implement in BSWMDT |
| (3) | ProgrammingLanguage | Public | Contain programming language of tag Implement in BSWMDT |
| (4) | SwVersion | Public | Contain software version of tag Implement in BSWMDT |
| (5) | VendorId | Public | Contain vendor id of tag Implement in BSWMDT |
| (6) | ArReleaseVersion | Public | Contain ar release version of tag Implement in BSWMDT |
| (7) | BehaviorRef | Public | Contain behavior reference of tag Implement in BSWMDT |
| (8) | VendorApiInfix | Public | Contain vendor api infix of tag Implement in BSWMDT |
| (9) | DescriptionUuid | Public | Contain uuid of tag Description in BSWMDT |
| (10) | DescriptionShortName | Public | Contain short name of tag Description in BSWMDT |
| (11) | ModuleId | Public | Contain module id of tag Description in BSWMDT |
| (12) | VendorSpecificModuleDefRef | Public | Contain information Vendor Specific Module DefRef |

1. Method List

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

#### Fields

##### ImplementUuid

GENERIC\_GDD\_CLS\_037\_001:

|  |  |
| --- | --- |
| **Variable Name** | ImplementUuid |
| **Type** | string |
| **Range** | None |
| **Description** | Contain uuid of tag Implement in BSWMDT |

##### ImplementSortName

GENERIC\_GDD\_CLS\_037\_002:

|  |  |
| --- | --- |
| **Variable Name** | ImplementSortName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain short name of tag Implement in BSWMDT |

##### ProgrammingLanguage

GENERIC\_GDD\_CLS\_037\_003:

|  |  |
| --- | --- |
| **Variable Name** | ProgrammingLanguage |
| **Type** | string |
| **Range** | None |
| **Description** | Contain programming language of tag Implement in BSWMDT |

##### SwVersion

GENERIC\_GDD\_CLS\_037\_004:

|  |  |
| --- | --- |
| **Variable Name** | SwVersion |
| **Type** | string |
| **Range** | None |
| **Description** | Contain software version of tag Implement in BSWMDT |

##### VendorId

GENERIC\_GDD\_CLS\_037\_005:

|  |  |
| --- | --- |
| **Variable Name** | VendorId |
| **Type** | string |
| **Range** | None |
| **Description** | Contain vendor id of tag Implement in BSWMDT |

##### ArReleaseVersion

GENERIC\_GDD\_CLS\_037\_006:

|  |  |
| --- | --- |
| **Variable Name** | ArReleaseVersion |
| **Type** | string |
| **Range** | None |
| **Description** | Contain ar release version of tag Implement in BSWMDT |

##### BehaviorRef

GENERIC\_GDD\_CLS\_037\_007:

|  |  |
| --- | --- |
| **Variable Name** | BehaviorRef |
| **Type** | string |
| **Range** | None |
| **Description** | Contain behavior reference of tag Implement in BSWMDT |

##### VendorApiInfix

GENERIC\_GDD\_CLS\_037\_008:

|  |  |
| --- | --- |
| **Variable Name** | VendorApiInfix |
| **Type** | string |
| **Range** | None |
| **Description** | Contain vendor api infix of tag Implement in BSWMDT |

##### DescriptionUuid

GENERIC\_GDD\_CLS\_037\_009:

|  |  |
| --- | --- |
| **Variable Name** | DescriptionUuid |
| **Type** | string |
| **Range** | None |
| **Description** | Contain uuid of tag Description in BSWMDT |

##### DescriptionShortName

GENERIC\_GDD\_CLS\_037\_010:

|  |  |
| --- | --- |
| **Variable Name** | DescriptionShortName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain short name of tag Description in BSWMDT |

##### ModuleId

GENERIC\_GDD\_CLS\_037\_011:

|  |  |
| --- | --- |
| **Variable Name** | ModuleId |
| **Type** | string |
| **Range** | None |
| **Description** | Contain module id of tag Description in BSWMDT |

##### VendorSpecificModuleDefRef

GENERIC\_GDD\_CLS\_037\_012:

|  |  |
| --- | --- |
| **Variable Name** | VendorSpecificModuleDefRef |
| **Type** | string |
| **Range** | None |
| **Description** | Contain 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 | Contain Uuid in tag ECUC-MODULE-CONFIGURATION-VALUES in CDF files. |
| (2) | ShortName | Public | Contain short name in tag ECUC-MODULE-CONFIGURATION-VALUES in CDF files. |
| (3) | DefinitionRef | Public | Contain DEFINITION-REF information in tag ECUC-MODULE-CONFIGURATION-VALUES in CDF files. |
| (4) | ModuleDescriptionRef | Public | Contain content of tag MODULE-DESCRIPTION-REF in CDF files. |

1. Method List

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

#### Fields

##### Uuid

GENERIC\_GDD\_CLS\_038\_001:

|  |  |
| --- | --- |
| **Variable Name** | Uuid |
| **Type** | string |
| **Range** | None |
| **Description** | Contain Uuid in tag ECUC-MODULE-CONFIGURATION-VALUES in CDF files. |

##### ShortName

GENERIC\_GDD\_CLS\_038\_002:

|  |  |
| --- | --- |
| **Variable Name** | ShortName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain short name in tag ECUC-MODULE-CONFIGURATION-VALUES in CDF files. |

##### DefinitionRef

GENERIC\_GDD\_CLS\_038\_003:

|  |  |
| --- | --- |
| **Variable Name** | DefinitionRef |
| **Type** | string |
| **Range** | None |
| **Description** | Contain DEFINITION-REF information in tag ECUC-MODULE-CONFIGURATION-VALUES in CDF files. |

##### ModuleDescriptionRef

GENERIC\_GDD\_CLS\_038\_004:

|  |  |
| --- | --- |
| **Variable Name** | ModuleDescriptionRef |
| **Type** | string |
| **Range** | None |
| **Description** | Contain content of tag MODULE-DESCRIPTION-REF in CDF files. |

#### Methods

### Renesas::Generator::MCALConfGen::Data::CDFData::Items::Container

|  |  |
| --- | --- |
| **Location File Name** | Container.cs |
| **Description** | This class that stores information of a CDF tag ECUC-CONTAINER-VALUE |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Uuid | Public | Contain Uuid in tag ECUC-CONTAINER-VALUE in CDF files. |
| (2) | ShortName | Public | Contain short name in tag ECUC-CONTAINER-VALUE in CDF files. |
| (3) | DefinitionRef | Public | Contain definition reference in tag ECUC-CONTAINER-VALUE in CDF files. |
| (4) | ParameterValues | Public | Contain parameter values in tag ECUC-CONTAINER-VALUE/PARAMETER-VALUES in CDF files. |
| (5) | ReferenceValues | Public | Contain reference values in tag ECUC-CONTAINER-VALUE/REFERENCE-VALUES in CDF files. |
| (6) | Parent | Public | Contain parent of container information |
| (7) | Path | Public | Contain path information of container as CanConfigSet/CanController/CanControllerBaudrateConfig |
| (8) | Childs | Public | Contain childs of container information |

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 list of children by definition reference name |
| (9) | GetChildByDefName | Public | Get a child by definition reference name |
| (10) | Compare | Public | Compare 2 containers by their def names and short names |

#### Fields

##### Uuid

GENERIC\_GDD\_CLS\_039\_001:

|  |  |
| --- | --- |
| **Variable Name** | Uuid |
| **Type** | string |
| **Range** | None |
| **Description** | Contain Uuid in tag ECUC-CONTAINER-VALUE in CDF files. |

##### ShortName

GENERIC\_GDD\_CLS\_039\_002:

|  |  |
| --- | --- |
| **Variable Name** | ShortName |
| **Type** | string |
| **Range** | None |
| **Description** | Contain short name in tag ECUC-CONTAINER-VALUE in CDF files. |

##### DefinitionRef

GENERIC\_GDD\_CLS\_039\_003:

|  |  |
| --- | --- |
| **Variable Name** | DefinitionRef |
| **Type** | string |
| **Range** | None |
| **Description** | Contain definition reference in tag ECUC-CONTAINER-VALUE in CDF files. |

##### ParameterValues

GENERIC\_GDD\_CLS\_039\_004:

|  |  |
| --- | --- |
| **Variable Name** | ParameterValues |
| **Type** | IList< ItemValue |
| **Range** | None |
| **Description** | Contain parameter values in tag ECUC-CONTAINER-VALUE/PARAMETER-VALUES in CDF files. |

##### ReferenceValues

GENERIC\_GDD\_CLS\_039\_005:

|  |  |
| --- | --- |
| **Variable Name** | ReferenceValues |
| **Type** | IList< ItemValue |
| **Range** | None |
| **Description** | Contain reference values in tag ECUC-CONTAINER-VALUE/REFERENCE-VALUES in CDF files. |

##### Parent

GENERIC\_GDD\_CLS\_039\_006:

|  |  |
| --- | --- |
| **Variable Name** | Parent |
| **Type** | Container |
| **Range** | None |
| **Description** | Contain parent of container information |

##### Path

GENERIC\_GDD\_CLS\_039\_007:

|  |  |
| --- | --- |
| **Variable Name** | Path |
| **Type** | string |
| **Range** | None |
| **Description** | Contain path information of container as CanConfigSet/CanController/CanControllerBaudrateConfig |

##### Childs

GENERIC\_GDD\_CLS\_039\_008:

|  |  |
| --- | --- |
| **Variable Name** | Childs |
| **Type** | List< Container |
| **Range** | None |
| **Description** | Contain childs of container information |

#### Methods

##### Compare

GENERIC\_GDD\_CLS\_039\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Compare | | |
| **Arguments** | con1 | None | |
| Range: | None |
| con2 | None | |
| Range: | None |
| **Return** | return a int | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Compare 2 containers by their def names and short names  Algorithm: IF both are null THEN  RETURN 0  ELSE IF container1 is null AND container2 NOT null THEN  RETURN -1  ELSE IF container1 NOT null AND container2 is null THEN  RETURN 1  ELSE  IF container1.DefinitionRef and container2.DefinitionRef NOT equal THEN  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\_039\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Equals | | |
| **Arguments** | object target | Target object | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check if this object and target are equal  Algorithm: IF target is null or not instance of Container THEN  RETURN false  ELSE  IF Uuid and short name of this object are equal to target object THEN  RETURN true  ELSE  RETURN false | | |

{Ref: N/A}

##### GetChildByDefName

GENERIC\_GDD\_CLS\_039\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetChildByDefName | | |
| **Arguments** | string defName | Def name | |
| Range: | None |
| int sortOption | Sort option | |
| Range: | None |
| **Return** | return a | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get a child by definition reference name  Algorithm: Get a list of child by def name by calling method GetChildsByDefName  IF not found THEN  Return null  ELSE  Return first child | | |

{Ref: N/A}

##### GetChildsByDefName

GENERIC\_GDD\_CLS\_039\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetChildsByDefName | | |
| **Arguments** | string defName | Def name | |
| Range: | None |
| int sortOption | Sort option | |
| Range: | None |
| **Return** | return a List of | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get list of children by definition reference 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\_039\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetDescendant | | |
| **Arguments** | None | | |
| **Return** | return a List of | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get all descendant containers (including this container)  Algorithm: list.Add(this container)   Recursively scan all descendant containers of this container  list.Add(descendant container)  End   Return list | | |

{Ref: N/A}

##### GetHashCode

GENERIC\_GDD\_CLS\_039\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetHashCode | | |
| **Arguments** | None | | |
| **Return** | return a int | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get hash code of container  Algorithm: IF Uuid is NOT null or empty THEN  RETURN hash code of Uuid  ELSE  RETURN hash code from base method Object.GetHashCode() | | |

{Ref: N/A}

##### GetParameterValue

GENERIC\_GDD\_CLS\_039\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetParameterValue | | |
| **Arguments** | string paramName | Param name | |
| Range: | None |
| **Return** | return a | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get a parameter value from name  Algorithm: Scan all ParameterValues of this container  IF definition name of a parameter item is equal to paramName THEN  RETURN first found parameter item  End  End   Return null if NOT found | | |

{Ref: N/A}

##### GetParameterValues

GENERIC\_GDD\_CLS\_039\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetParameterValues | | |
| **Arguments** | string paramName | Param name | |
| Range: | None |
| **Return** | return a List of | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get all parameters' values from name  Algorithm: Scan all ParameterValues of this container  If definition name of a parameter item is equal to paramName THEN  RETURN a list of found parameter items  End  End   Return null if NOT found | | |

{Ref: N/A}

##### GetReferenceValue

GENERIC\_GDD\_CLS\_039\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetReferenceValue | | |
| **Arguments** | string paramName | Param name | |
| Range: | None |
| **Return** | return a | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get a reference value from name  Algorithm: Scan all ReferenceValues of this container  IF definition name of a reference item is equal to paramName THEN  RETURN first found reference item  End  End   Return null if NOT found | | |

{Ref: N/A}

##### GetReferenceValues

GENERIC\_GDD\_CLS\_039\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetReferenceValues | | |
| **Arguments** | string paramName | Param name | |
| Range: | None |
| **Return** | return a List of | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get all references' values from name  Algorithm: Scan all ReferenceValues of this container  IF definition name of a reference item is equal to paramName THEN  RETURN a list of found reference items  End  End   Return null if NOT found | | |

{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 | Contain definition reference in tag DEFINITION-REF. E.g: /Renesas/EcucDefs\_Can/Can/CanConfigSet/CanController/CanBusoffProcessing |
| (2) | value | Private | Contain value of definition reference in tag VALUE. E.g: POLLING |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | ItemValue | Public | Item value constructor with arguments |
| (2) | DefinitionRef | Public | Get definition reference information |
| (3) | Value< T > | Public | Get value of definition reference information |
| (4) | InternalValue | Public | Get internal value of definition reference information Only used by Generic and for testing |

#### Fields

##### definitionRef

GENERIC\_GDD\_CLS\_040\_001:

|  |  |
| --- | --- |
| **Variable Name** | definitionRef |
| **Type** | readonly string |
| **Range** | None |
| **Description** | Contain definition reference in tag DEFINITION-REF. E.g: /Renesas/EcucDefs\_Can/Can/CanConfigSet/CanController/CanBusoffProcessing |

##### value

GENERIC\_GDD\_CLS\_040\_002:

|  |  |
| --- | --- |
| **Variable Name** | value |
| **Type** | readonly object |
| **Range** | None |
| **Description** | Contain value of definition reference in tag VALUE. E.g: POLLING |

#### Methods

##### DefinitionRef

GENERIC\_GDD\_CLS\_040\_003:

|  |  |
| --- | --- |
| **Method Name** | DefinitionRef |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Get definition reference information |

{Ref: N/A}

##### InternalValue

GENERIC\_GDD\_CLS\_040\_004:

|  |  |
| --- | --- |
| **Method Name** | InternalValue |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Get internal value of definition reference information Only used by Generic and for testing |

{Ref: N/A}

##### ItemValue

GENERIC\_GDD\_CLS\_040\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ItemValue | | |
| **Arguments** | string definitionRef | None | |
| Range: | None |
| object value | None | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Item value constructor with arguments  Algorithm: Init values such as definitionRef, value | | |

{Ref: N/A}

##### Value< T >

GENERIC\_GDD\_CLS\_040\_006:

|  |  |
| --- | --- |
| **Method Name** | Value< T > |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Get value of definition reference information |

{Ref: N/A}

### Renesas::Generator::MCALConfGen::Data::CDFData::RepositoryLinQ

|  |  |
| --- | --- |
| **Location File Name** | RepositoryLinQ.cs |
| **Description** | This RepositoryLinQ class that implements IRepository interfaces and use basic operations of CdfDataDictionary to provide more convenience operations to upper layers. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | RepositoryLinQ instance used by ObjectFactory to get a new RepositoryLinQ object |
| (2) | cdfData | Private | Contain data from cdf file as a database |
| (3) | register | Private | Contain component Register Processing to access register value and definition macro |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetCdfFileName | Public | Get CDF file name that stored in database |
| (2) | GetContainerByDefName | Public | Get a container by definition reference E.g: <DEFINITION-REF> /Renesas/EcucDefs\_Can/Can/CanConfigSet/CanController </DEFINITION-REF> GetContainerByDefName("Can", "CanController") |
| (3) | GetContainersByDefName | Public | Get all container by definition reference E.g: <DEFINITION-REF> /Renesas/EcucDefs\_Can/Can/CanConfigSet/CanController </DEFINITION-REF> GetContainersByDefName("Can", "CanController") |
| (4) | GetContainerByShortNamePath | Public | Get a container by short name path E.g: path: CanConfigSet/CanController/CanControllerBaudrateConfig |
| (5) | GetContainers | Public | Get all containers by module name E.g: GetContainers("Can"), GetContainers("Dio"), GetContainers("Dem") |
| (6) | GetMacroAddressValue | Public | Get macro address value from database as Register |
| (7) | GetAddressByMacroDefinition | Public | Get address by macro definition from database as Register |
| (8) | GetMacroLabelValue | Public | Get value based on definition macro from database |
| (9) | GetParameterValue | Public | Get value of parameter based on definition reference name |
| (10) | GetParameterValues | Public | Get parameter values by definition reference name |
| (11) | GetReferenceValue | Public | Get reference value of container based on definition reference |
| (12) | GetChilds | Public | Get all children container of a container which is identified by module name and short name |
| (13) | GetContainer | Public | Get container based on module name and short name |
| (14) | GetVersionInformation | Public | Get version information based on components name from BSWMDT E.g: Components name: AR-RELEASE-VERSION SW-VERSION VENDOR-ID MODULE-ID VENDOR-API-INFIX |
| (15) | IsExistedPath | Public | Check whether path does exist or not. |
| (16) | GetStartOfDbToc | Public | Get start of db toc |
| (17) | IsExistModuleName | Public | Check whether module name does exist or not. |
| (18) | GetConfiguration | Public | Get configuration based on module name include on information as follow: Short name Definition reference |
| (19) | GetConfigurations | Public | Get all configurations based on module name include on information as follow: Short name Definition reference |
| (20) | IsRegisterExist | Public | Check whether register does exist or not. |
| (21) | ComputeOnOffParams | Public | Compute values of bool parameter such as STD\_ON or STD\_OFF |
| (22) | ComputeDemEventParams | Public | Process values of dem event paramaters. |
| (23) | GetVersionInformation | Public | To compute value of all parameters Version Information |
| (24) | RepositoryLinQ | Protected | Repository LinQ constructor without arguments. |
| (25) | RepositoryLinQ | Protected | Repository LinQ constructor with arguments |
| (26) | initVersionInformation | Private | Init values autosar version requirement and software version |

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_041\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly RepositoryLinQ |
| **Range** | None |
| **Description** | RepositoryLinQ instance used by ObjectFactory to get a new RepositoryLinQ object |

##### cdfData

GENERIC\_GDD\_CLS\_041\_002:

|  |  |
| --- | --- |
| **Variable Name** | cdfData |
| **Type** | ICdfData |
| **Range** | None |
| **Description** | Contain data from cdf file as a database |

##### register

GENERIC\_GDD\_CLS\_041\_003:

|  |  |
| --- | --- |
| **Variable Name** | register |
| **Type** | IRegisterProcessing |
| **Range** | None |
| **Description** | Contain component Register Processing to access register value and definition macro |

#### Methods

##### ComputeDemEventParams

GENERIC\_GDD\_CLS\_041\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ComputeDemEventParams | | |
| **Arguments** | Dictionary< string, string > demEventParamsGroup | A dictionary contains key as parameter name and value as definition macro | |
| Range: | None |
| **Return** | BaseIntermediateItem[] | None | |
| Range: | None |
| **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\_041\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ComputeOnOffParams | | |
| **Arguments** | Dictionary< string, string > onOffParamsGroup | A dictionary contains key as parameter name and value as generated definition macro | |
| Range: | None |
| **Return** | BaseIntermediateItem[] | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Compute values of bool parameter such as STD\_ON or STD\_OFF  Algorithm: Based on parameter name, get value from CDFs.  Set STD\_ON If value is true and else set false | | |

{Ref: N/A}

##### GetAddressByMacroDefinition

GENERIC\_GDD\_CLS\_041\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetAddressByMacroDefinition | | |
| **Arguments** | string renesasMacroName | None | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get address by macro definition from database as Register  Algorithm: Get macro value from macro definition by calling method GetMacroLabelValue  IF macro value is NOT null or empty THEN  Get macro address from macro value by calling method GetMacroAddressValue  RETURN macro address  ELSE  RETURN empty | | |

{Ref: N/A}

##### GetCdfFileName

GENERIC\_GDD\_CLS\_041\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetCdfFileName | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get CDF file name that stored in database  Algorithm: Get CDF file name based on moduleName from database as cdfFiles | | |

{Ref: N/A}

##### GetChilds

GENERIC\_GDD\_CLS\_041\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetChilds | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string shortName | Short name | |
| Range: | None |
| **Return** | return List of Container | None | |
| Range: | None |
| **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 THEN  Recursively find children of the container  RETURN found children  ELSE  RETURN empty list | | |

{Ref: N/A}

##### GetConfiguration

GENERIC\_GDD\_CLS\_041\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetConfiguration | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return a Configuration | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get configuration based on module name include on information as follow: Short name Definition reference  Algorithm: Get configuration of MCAL module from database as cdfData | | |

{Ref: N/A}

##### GetConfigurations

GENERIC\_GDD\_CLS\_041\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetConfigurations | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return all Configuration | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get all configurations based on module name include on information as follow: Short name Definition reference  Algorithm: Get all configurations of MCAL module from database as cdfData based on moduleName | | |

{Ref: N/A}

##### GetContainer

GENERIC\_GDD\_CLS\_041\_011:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainer | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string shortName | Short name | |
| Range: | None |
| **Return** | return a Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get container based on module name and short name  Algorithm: Get a list of containers in cdfData by moduleName  FOREACH container in container list  IF container's ShortName = shortName THEN  Return container  End  End   Return null if NOT found a container | | |

{Ref: N/A}

##### GetContainerByDefName

GENERIC\_GDD\_CLS\_041\_012:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainerByDefName | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string defName | Def name | |
| Range: | None |
| int sortOption | Sort option | |
| Range: | None |
| **Return** | return a Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get a container by definition reference E.g: <DEFINITION-REF> /Renesas/EcucDefs\_Can/Can/CanConfigSet/CanController </DEFINITION-REF> GetContainerByDefName("Can", "CanController")  Algorithm: CALL GetContainersByDefName to get a list of containers  IF found a container THEN  RETURN first found  ELSE  RETURN null | | |

{Ref: N/A}

##### GetContainerByShortNamePath

GENERIC\_GDD\_CLS\_041\_013:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainerByShortNamePath | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string path | Path | |
| Range: | None |
| **Return** | return a Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get a container by short name path E.g: path: CanConfigSet/CanController/CanControllerBaudrateConfig  Algorithm: Get all containers in cdfData by moduleName  IF found a container where container.Path = path THEN  RETURN first found  ELSE  RETURN null | | |

{Ref: N/A}

##### GetContainers

GENERIC\_GDD\_CLS\_041\_014:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainers | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return a Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get all containers by module name E.g: GetContainers("Can"), GetContainers("Dio"), GetContainers("Dem")  Algorithm: Get all containers in cdfData by moduleName | | |

{Ref: N/A}

##### GetContainersByDefName

GENERIC\_GDD\_CLS\_041\_015:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetContainersByDefName | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string defName | Def name | |
| Range: | None |
| int sortOption | Sort option | |
| Range: | None |
| **Return** | return List of Container | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get all container by definition reference E.g: <DEFINITION-REF> /Renesas/EcucDefs\_Can/Can/CanConfigSet/CanController </DEFINITION-REF> GetContainersByDefName("Can", "CanController")  Algorithm: IF defName is NOT null or empty THEN  Get list of containers from cdfData by moduleName where last element of container's DefinitionRef  is defName  Else  Get list of containers from cdfData by moduleName   IF sortOption = 1 THEN  Sort the list of containers  End   RETURN the list of containers | | |

{Ref: N/A}

##### GetMacroAddressValue

GENERIC\_GDD\_CLS\_041\_016:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetMacroAddressValue | | |
| **Arguments** | string macroValue | Macro value | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get macro address value from database as Register  Algorithm: CALL GetMacroAddressValue from component RegisterProcessing to get address value | | |

{Ref: N/A}

##### GetMacroLabelValue

GENERIC\_GDD\_CLS\_041\_017:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetMacroLabelValue | | |
| **Arguments** | string renesasMacroName | None | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get value based on definition macro from database  Algorithm: CALL GetMacroLabelValue from component RegisterProcessing to get label value | | |

{Ref: N/A}

##### GetParameterValue

GENERIC\_GDD\_CLS\_041\_018:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetParameterValue | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string defName | Definition Reference name | |
| Range: | None |
| **Return** | return a ItemValue | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get value of parameter based on definition reference name  Algorithm: Get all containers in database as cdfData based on moduleName  IF found any parameter of container has defName THEN  RETURN first found parameter  ELSE  RETURN null | | |

{Ref: N/A}

##### GetParameterValues

GENERIC\_GDD\_CLS\_041\_019:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetParameterValues | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string defName | Definition reference name | |
| Range: | None |
| **Return** | return a list of ItemValue | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get parameter values by definition reference name  Algorithm: Get all containers in cdfData by moduleName  Scan all container in container list  Get parameter value by defName and add to a collection "result"  End   RETURN collection "result" | | |

{Ref: N/A}

##### GetReferenceValue

GENERIC\_GDD\_CLS\_041\_020:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetReferenceValue | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| string defName | Definition Reference name | |
| Range: | None |
| **Return** | return a ItemValue | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get reference value of container based on definition reference  Algorithm: Get all containers in cdfData by moduleName  IF found any reference of container has defName THEN  RETURN first found reference  ELSE  RETURN null | | |

{Ref: N/A}

##### GetStartOfDbToc

GENERIC\_GDD\_CLS\_041\_021:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetStartOfDbToc | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **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\_041\_022:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetVersionInformation | | |
| **Arguments** | string name | Name | |
| Range: | None |
| string moduleName | Module Name | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get version information based on components name from BSWMDT E.g: Components name: AR-RELEASE-VERSION SW-VERSION VENDOR-ID MODULE-ID VENDOR-API-INFIX  Algorithm: Get basic software config in cdfData by moduleName  IF found a field in basic software config based that match with name THEN  RETURN value of found field  ELSE  RETURN null | | |

{Ref: N/A}

##### GetVersionInformation

GENERIC\_GDD\_CLS\_041\_023:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetVersionInformation | | |
| **Arguments** | None | | |
| **Return** | Return BaseIntermediateItem: Name: GetVersionInformation Value: empty | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | To compute value of all parameters 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\_041\_024:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsExistModuleName | | |
| **Arguments** | string moduleName | Module name | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether module name does exist or not.  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\_041\_025:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsExistedPath | | |
| **Arguments** | string path | Path | |
| Range: | None |
| string moduleName | Module name | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether path does exist or not.  Algorithm: Get all containers in cdfData by moduleName  FOREACH container in container list  IF found a container that container.Path = path THEN  RETURN true  ELSE  RETURN false  End | | |

{Ref: N/A}

##### IsRegisterExist

GENERIC\_GDD\_CLS\_041\_026:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IsRegisterExist | | |
| **Arguments** | string renesasMacroDefinition | None | |
| Range: | None |
| **Return** | return a bool | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Check whether register does exist or not.  Algorithm: CALL IsRegisterExist from component Register Processing  to check whether macro definition does exist or not. | | |

{Ref: N/A}

##### RepositoryLinQ

GENERIC\_GDD\_CLS\_041\_027:

|  |  |
| --- | --- |
| **Method Name** | RepositoryLinQ |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Repository LinQ constructor without arguments. |

{Ref: N/A}

##### RepositoryLinQ

GENERIC\_GDD\_CLS\_041\_028:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | RepositoryLinQ | | |
| **Arguments** | cdfData | Cdf data | |
| Range: | None |
| register | Register | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Repository LinQ constructor with arguments  Algorithm: Init values such as cdfData, register | | |

{Ref: N/A}

##### initVersionInformation

GENERIC\_GDD\_CLS\_041\_029:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | initVersionInformation | | |
| **Arguments** | string[] autosarReqInfo | None | |
| Range: | None |
| string[] softWareInfo | None | |
| Range: | None |
| string vendorId | None | |
| Range: | None |
| string moduleId | None | |
| Range: | None |
| **Return** | Dictionary | None | |
| Range: | None |
| **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}

### 

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

X2x

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

X2x

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

X2xX2x

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

X2xX2x

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

X2x

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

X2xX2x

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | |
|  |  |
|  |  | | |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |
|  |  | | |

X2x

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

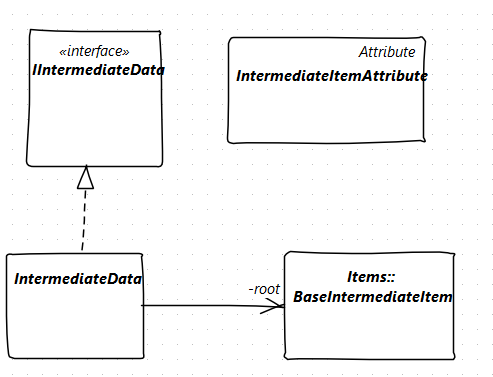
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

## Intermediate Data

This component store values that calculated by MCAL module to database. This part is separated from Validation and Generation components, it only has calculation charge and others component will use it’s values to validate or generate to output.

### UML

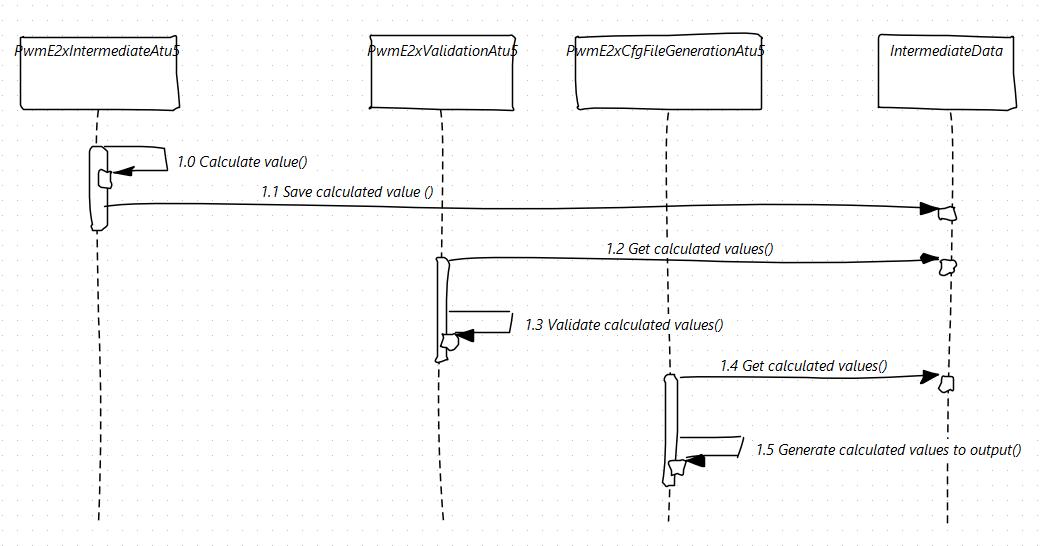


The IntermediateData class implement methods for store and get calculated values from MCAL modules.

The IntermediateItemAttribute class is used to mark for allowing whether a calculated value should be stored or not.

The BaseIntermediateItem class define database structure of Intermediate component

A sample describe task of Intermediate component



### Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateData

|  |  |
| --- | --- |
| **Location File Name** | IntermediateData.cs |
| **Description** | This IntermediateData class to implement IIntermediateData interface to save and retrive intermediate data. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Instance | Public | IntermediateData instance used by ObjectFactory to get a new IntermediateData object |
| (2) | root | Private | Contain intermediate item as root |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | GetRoot | Public | Get intermediate item as root |
| (2) | GetStringDataByPath | Public | Get string data by path |
| (3) | GetItemByPath | Public | Get item by path |
| (4) | Store | Public | Store intermediate item for root node. |
| (5) | IntermediateData | Private | Intermediate data constructor without arguments |

#### Fields

##### Instance

GENERIC\_GDD\_CLS\_042\_001:

|  |  |
| --- | --- |
| **Variable Name** | Instance |
| **Type** | readonly IntermediateData |
| **Range** | None |
| **Description** | IntermediateData instance used by ObjectFactory to get a new IntermediateData object |

##### root

GENERIC\_GDD\_CLS\_042\_002:

|  |  |
| --- | --- |
| **Variable Name** | root |
| **Type** | BaseIntermediateItem |
| **Range** | None |
| **Description** | Contain intermediate item as root |

#### Methods

##### GetItemByPath

GENERIC\_GDD\_CLS\_042\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetItemByPath | | |
| **Arguments** | string path | Path | |
| Range: | None |
| **Return** | return a BaseIntermediateItem | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get item by path  Algorithm: Get BaseIntermediateItem by path from root BaseIntermediateItem | | |

{Ref: N/A}

##### GetRoot

GENERIC\_GDD\_CLS\_042\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetRoot | | |
| **Arguments** | None | | |
| **Return** | return a BaseIntermediateItem | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get intermediate item as root  Algorithm: Return root attribute | | |

{Ref: N/A}

##### GetStringDataByPath

GENERIC\_GDD\_CLS\_042\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetStringDataByPath | | |
| **Arguments** | string path | Path | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get string data by path  Algorithm: Get string data by path from root BaseIntermediateItem | | |

{Ref: N/A}

##### Store

GENERIC\_GDD\_CLS\_042\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Store | | |
| **Arguments** | string path | Path | |
| Range: | None |
| target | Target | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Store intermediate item for root node.  Algorithm: Store BaseIntermediateItem with path into root with key as path | | |

{Ref: N/A}

##### IntermediateData

GENERIC\_GDD\_CLS\_042\_007:

|  |  |
| --- | --- |
| **Method Name** | IntermediateData |
| **Arguments** | None |
| **Return** | None |
| **Generated Value** | None |
| **Description** | Intermediate data constructor without arguments |

{Ref: N/A}

### Renesas::Generator::MCALConfGen::Data::IntermediateData::IntermediateItemAttribute

|  |  |
| --- | --- |
| **Location File Name** | IntermediateItemAttribute.cs |
| **Description** | This IntermediateItemAttribute class that decorates methods which will be invoked to calculate intermediate data. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | DBPath | Public | Contain path of intermediate item as key used to access intermediate items. |
| (2) | Line | Public | contain line number as order of intermediate items in database |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | IntermediateItemAttribute | Public | Intermediate item attribute constructor with arguments |

#### Fields

##### DBPath

GENERIC\_GDD\_CLS\_043\_001:

|  |  |
| --- | --- |
| **Variable Name** | DBPath |
| **Type** | string |
| **Range** | None |
| **Description** | Contain path of intermediate item as key used to access intermediate items. |

##### Line

GENERIC\_GDD\_CLS\_043\_002:

|  |  |
| --- | --- |
| **Variable Name** | Line |
| **Type** | int |
| **Range** | None |
| **Description** | contain line number as order of intermediate items in database |

#### Methods

##### IntermediateItemAttribute

GENERIC\_GDD\_CLS\_043\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | IntermediateItemAttribute | | |
| **Arguments** | int line | line | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Intermediate item attribute constructor with arguments  Algorithm: Init line value | | |

{Ref: N/A}

### Renesas::Generator::MCALConfGen::Data::IntermediateData::Items::BaseIntermediateItem

|  |  |
| --- | --- |
| **Location File Name** | BaseIntermediateItem.cs |
| **Description** | This BaseIntermediateItem class to store an intermediate data. |

1. Field List

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| (1) | Name | Public | Contain name of parameter that will be generated as definition macro. |
| (2) | Value | Public | Contain calculated value that will be generated according to definition macro. |
| (3) | Childs | Public | Contain all items as child of itself. |

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | BaseIntermediateItem | Public | Base intermediate item constructor with arguments |
| (2) | BaseIntermediateItem | Public | Base intermediate item constructor with arguments |
| (3) | AddChild | Public | Add a child as intermediate items. |
| (4) | ToString | Public | Formating display name and value of intermediate item. |
| (5) | GetStringDataByPath | Public | Get intermediate item value based on path |
| (6) | GetItemByPath | Public | Get intermediate item based on path |
| (7) | Store | Public | Store intermediate item in database. |

#### Fields

##### Name

GENERIC\_GDD\_CLS\_044\_001:

|  |  |
| --- | --- |
| **Variable Name** | Name |
| **Type** | string |
| **Range** | None |
| **Description** | Contain name of parameter that will be generated as definition macro. |

##### Value

GENERIC\_GDD\_CLS\_044\_002:

|  |  |
| --- | --- |
| **Variable Name** | Value |
| **Type** | string |
| **Range** | None |
| **Description** | Contain calculated value that will be generated according to definition macro. |

##### Childs

GENERIC\_GDD\_CLS\_044\_003:

|  |  |
| --- | --- |
| **Variable Name** | Childs |
| **Type** | List< BaseIntermediateItem |
| **Range** | None |
| **Description** | Contain all items as child of itself. |

#### Methods

##### AddChild

GENERIC\_GDD\_CLS\_044\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | AddChild | | |
| **Arguments** | child | Child | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Add a child as intermediate items.  Algorithm: Add child to a children list | | |

{Ref: N/A}

##### BaseIntermediateItem

GENERIC\_GDD\_CLS\_044\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseIntermediateItem | | |
| **Arguments** | string name | Name | |
| Range: | None |
| string value | Value | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base intermediate item constructor with arguments  Algorithm: Init values such as name, value. | | |

{Ref: N/A}

##### BaseIntermediateItem

GENERIC\_GDD\_CLS\_044\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | BaseIntermediateItem | | |
| **Arguments** | string name | Name | |
| Range: | None |
| string value | Value | |
| Range: | None |
| List< childs | Childs | |
| Range: | None |
| **Return** | None | | |
| **Generated Value** | None | | |
| **Description** | Base intermediate item constructor with arguments  Algorithm: Init values such as name, value, childs | | |

{Ref: N/A}

##### GetItemByPath

GENERIC\_GDD\_CLS\_044\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetItemByPath | | |
| **Arguments** | string path | Path | |
| Range: | None |
| **Return** | return a | None | |
| Range: |  |
| **Generated Value** | None | | |
| **Description** | Get intermediate item based on 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   For example:  BaseIntermediateItem\_01  Name = "Segment\_01"  Childs = ---------- BaseIntermediateItem\_01\_01  | Name = "Segment\_01\_01"  | Childs = null  +--------- BaseIntermediateItem\_01\_02  Name = "Segment\_01\_02"  Childs = null   Path = "/Segment\_01/Segment\_01\_02" will return BaseIntermediateItem\_01\_02 | | |

{Ref: N/A}

##### GetStringDataByPath

GENERIC\_GDD\_CLS\_044\_008:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | GetStringDataByPath | | |
| **Arguments** | string path | Path | |
| Range: | None |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get intermediate item value based on path  Algorithm: Get BaseIntermediateItem by path by calling method GetItemByPath  IF BaseIntermediateItem is NOT null THEN  RETURN BaseIntermediateItem.Value  ELSE  RETURN empty | | |

{Ref: N/A}

##### Store

GENERIC\_GDD\_CLS\_044\_009:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Store | | |
| **Arguments** | string path | Path | |
| Range: | None |
| target | Target | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Store intermediate item in database.  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\_044\_010:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | ToString | | |
| **Arguments** | None | | |
| **Return** | return a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Formating display name and value of intermediate item.  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}

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | |
|  |  |
|  |  | | |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |
|  |  | | |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

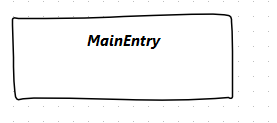
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  |  | |
|  |  |
|  |  | | |
|  |  | | |

## MainEntry

### UML



The MainEntry class defines entry point and main workflow in Generation Tool.

### Renesas::Generator::MCALConfGen::Presentation::MainEntry::MainEntry

|  |  |
| --- | --- |
| **Location File Name** | MainEntry.cs |
| **Description** | This MainEntry class defines entry point of Gentool. |

1. Field List

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

1. Method List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Description** |
| (1) | Main | Public | This is entry point |
| (2) | unhandledExceptionHandler | Private | Throw exception if have any unexpected exception |
| (3) | collectExceptionHierarchyInfo | Private | Collect exception and formating to display to user |
| (4) | printHelpIfNeed | Private | Print help tool based on option -h input from user |
| (5) | registerAssemblyTypes | Private | Store all assembly from MCAL module. |
| (6) | loadConfiguration | Private | Load basic information from specific module such as module name, module id, project title, microfamily. |
| (7) | getModuleName | Private | Get MCAL module name from database |

#### Fields

#### Methods

##### Main

GENERIC\_GDD\_CLS\_046\_001:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | Main | | |
| **Arguments** | string[] args | Args | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | This is entry point  Algorithm: Workflow of tool as follow:  1. Load all assemply of MCAL module  2. Read all configuration from MCAL module  3. Process input from command line  4. Parsing all data from CDF files, BSWMDT, Translation file  5. Validation all parameters parsed from CDf files.  6. Calculate all value of parameter from CDF files.  7. Validation all calculated value by step 6  8. Generate all values from step 6 to files .c, .h | | |

{Ref: GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_006, GENERIC\_TFD\_X2x\_GT\_SAMPUSG\_013}

##### collectExceptionHierarchyInfo

GENERIC\_GDD\_CLS\_046\_002:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | collectExceptionHierarchyInfo | | |
| **Arguments** | Exception | e | |
| Range: | None |
| **Return** | RETURN a string contain information of exception | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Collect exception and formating to display to user  Algorithm: Access exception object and format follow to template. | | |

{Ref: N/A}

##### getModuleName

GENERIC\_GDD\_CLS\_046\_003:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | getModuleName | | |
| **Arguments** | None | | |
| **Return** | RETURN a string | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Get MCAL module name from database  Algorithm: CALL method GetAndValidateModuleName from component Command Line. | | |

{Ref: N/A}

##### loadConfiguration

GENERIC\_GDD\_CLS\_046\_004:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | loadConfiguration | | |
| **Arguments** | Assembly assembly | Specific module assembly | |
| Range: | None |
| **Return** | return a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Load basic information from specific module such as module name, module id, project title, microfamily.  Algorithm: Load assembly and access resource to get module name, module id, project title, module required,  micro family name | | |

{Ref: N/A}

##### printHelpIfNeed

GENERIC\_GDD\_CLS\_046\_005:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | printHelpIfNeed | | |
| **Arguments** | None | | |
| **Return** | RETURN a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Print help tool based on option -h input from user  Algorithm: Print content help of tool according to template | | |

{Ref: N/A}

##### registerAssemblyTypes

GENERIC\_GDD\_CLS\_046\_006:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | registerAssemblyTypes | | |
| **Arguments** | None | | |
| **Return** | RETURN a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Store all assembly from MCAL module.  Algorithm: Scan all .dll in folder  Get .dll MCAL module based on module name  If .dll MCAL module does not found  Report error ERR000045  end  Load all basic configuration based on .dll MCAL module  Store all members of .dll MCAL module | | |

{Ref: [1] CMN\_GFD\_ERR\_045, GENERIC\_TFD\_X2x\_GT\_ERR\_045}

##### unhandledExceptionHandler

GENERIC\_GDD\_CLS\_046\_007:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | unhandledExceptionHandler | | |
| **Arguments** | object sender | Sender | |
| Range: | None |
| UnhandledExceptionEventArgs e | Exception | |
| Range: | None |
| **Return** | RETURN a void | None | |
| Range: | None |
| **Generated Value** | None | | |
| **Description** | Throw exception if have any unexpected exception  Algorithm: Check type of exception, print message then exit  If type of exception as MissingFieldException  report error ERR000043 | | |

{Ref: [1] CMN\_GFD\_ERR\_043, GENERIC\_TFD\_X2x\_GT\_ERR\_043, GENERIC\_TFD\_X2x\_GT\_ECU\_001}

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  | |
|  |  |
|  |  | |
|  |  |
|  |  | | |

# 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

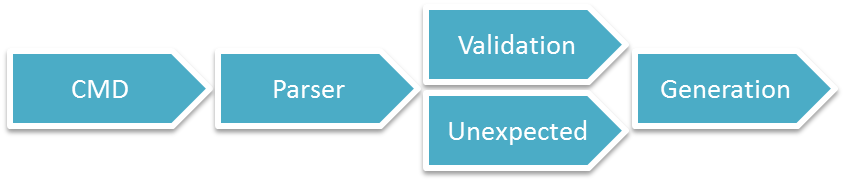
|  |  |
| --- | --- |
| Phase | Implementation |
| Requirement | EAAR-RS-0040.pdf.  Add one more functional requirement to describe how errors should be handled. |
| FD/DD | *RH850\_X2x\_Generic\_GenTool\_ErrorList.xlsx.*   * Add Error handling flow to Generic FD/DD * Group Error, Warning and Info messages 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

|  |  |  |
| --- | --- | --- |
| Document name | Uncovered IDs | Reason |
|  |  |  |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Ver.** | **Change** | **Changed By** |
| 03rd 01 2020 | 1.00 | Initial version | Tu Pham |