

# **FUJITSU Software Agile<sup>+</sup> Relief V1.1.1**

A horizontal decorative band with a dark red background. It features several glowing, white, curved lines that sweep across the band, creating a sense of motion and depth. The lines are layered, with some appearing closer and more vibrant than others, and they intersect to form a complex, abstract pattern.

## **Agile<sup>+</sup> Relief Manual**

## Preface

Agile+ Relief is the application that daily monitors the quality data after the source program compiled by C/C++ or Java is parsed and visualizes the position of quality problems.

Use Agile+ Relief J when parsing the source program of Java.

For the use of Agile+ Relief J, please refer to [Agile+ Relief J Command Manual].

Use Agile+ Relief C/C++ when parsing the source program of C/C++.

For the use of Agile+ Relief C/C++, please refer to [Agile+ Relief C/C++ Command Manual].

Agile+ Relief does not correspond to cooperation of Agile+ Relief C/C++ CERT Option.

This manual contains explanations of the usage of the Agile+ Relief.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States and other countries.

MISRA and its logo are registered trademarks of MISRA Ltd, held on behalf of the MISRA Consortium.

CERT is registered trademark of Carnegie Mellon University in the United States.

Interstage is the registered trademark of Fujitsu Limited.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

The names of other products and services referred to in this document are trademarks of their respective developers and manufacturers.

Particular technologies disclosed in this document may be subject to the Foreign Exchange and Foreign Trade Control Law. Export of this document or any part thereof, or supply of this document or any part thereof to a foreign resident, must be undertaken only in compliance with the provisions of the law.

FUJITSU LIMITED

### Notes

- Transmission or copying of this document in whole or in part is prohibited.
- The content of this document is subject to change without prior notice.

All rights reserved, Copyright(C) 2019-2023 FUJITSU LIMITED.

## Contents

1. What is Agile+ Relief?.....	1
1.1. Summary .....	1
1.1.1. Product Objectives.....	1
1.1.2. Usage scenarios.....	1
1.1.3. Functions.....	3
1.2. Structure of System and I/O File.....	5
2. How to use Agile+ Relief .....	7
2.1. Classification of Messages.....	7
2.2. Organization Mechanism.....	8
2.3. Procedures .....	9
2.3.1. Procedures under source analysis mode.....	9
2.3.1.1. Setting Check Base.....	9
2.3.1.2. Setting Output Content.....	10
2.3.1.3. Setting Objective for Check.....	10
2.3.1.4. Setting message before managed.....	11
2.3.1.5. Output quality info results.....	11
2.3.1.6. Confirm quality info results.....	11
2.3.1.7. Process messages.....	11
2.3.1.8. Combine the quality info results of multiple organizations....	12
2.3.2. Procedures under analysis result consolidation mode.....	13
2.3.2.1. Setting Check Base.....	13
2.3.2.2. Setting Output Content.....	13
2.3.2.3. Extracting Consolidation Object.....	14
2.3.2.4. Setting Consolidation Object.....	17
2.3.2.5. Setting message before managed.....	17
2.3.2.6. Output quality info results.....	17
2.3.2.7. Confirm quality info results.....	18
2.3.2.8. Process messages.....	18
2.3.2.9. Combine the quality info results of multiple organizations....	18
2.4. Execute Sample Project.....	18
2.4.1. Source analysis mode – Execute C/C++ Sample Project.....	18
2.4.1.1. Structure of sample project.....	19
2.4.1.2. Create check base.....	20
2.4.1.3. Create project structure definition file.....	22
2.4.1.4. Execute Batch Execution Command.....	22
2.4.1.5. Execute Sample of Different Folder Structure.....	23
2.4.2. Source analysis mode – Execute Java Sample Project.....	24
2.4.2.1. Structure of sample project.....	24
2.4.2.2. Create Check Base.....	25
2.4.2.3. Create project structure definition file.....	25
2.4.2.4. Execute Batch Execution Command.....	26
2.4.3. Analysis result consolidation mode – Execute Sample Project.....	27
2.4.3.1. Structure of sample project.....	27
2.4.3.2. Create check base.....	27
2.4.3.3. Analyze Source Program.....	28
2.4.3.4. Consolidate Analysis Result.....	29
2.5. Notes When Using.....	30
2.6. Suppress comment.....	30

3.	Contents of Quality Info Result .....	32
3.1.	Types of Output HTML Files.....	32
3.2.	Example of HTML File Output.....	33
3.3.	Relation Among Output HTML Files.....	41
3.4.	Types of Output CSV Files.....	41
3.5.	Template File Update.....	42
3.6.	Description of Definition File for Output Content.....	48
4.	Description of Various Definition Files.....	53
4.1.	Project Structure Definition.....	53
4.1.1.	Project Structure Definition File.....	54
4.1.2.	Project Structure Definition Listing File.....	71
4.1.3.	Example to Designate Target Folder for Project Creation.....	73
4.1.3.1.	Example to Designate Target Folder under source analysis mode.	73
4.1.3.2.	Example to Designate Target Folder under analysis result consolidation mode.....	75
4.2.	Definition of Check Base.....	77
4.2.1.	Check indication definition File and Check rule definition File.	77
4.2.1.1.	The checking rule definition in case of C/C++.....	77
4.2.1.2.	The checking rule definition in case of Java.....	80
4.2.2.	Dangerous /Attention Field Definition file.....	80
4.2.3.	Diagnosis Indicator Definition File.....	84
4.2.4.	Indication feature definition file.....	85
4.3.	Definition File for Output Content.....	87
5.	How to use arqm command .....	88
5.1.	Command Format.....	88
5.2.	Option .....	88
5.3.	Before Execution.....	88
5.4.	Project Analysis Result Information.....	89
5.5.	Set the message before managed.....	89
5.6.	Return Value.....	89
5.7.	Error Message.....	89
6.	To Agile+ Relief C/C++ MISRA Option User.....	97
6.1.	Replacement of Template File.....	97
6.2.	Description of Project Structure Definition File.....	98
7.	To IPA/SEC-C/C++ Coding Manner Check User.....	99
7.1.	Replacement of Template File.....	99
7.2.	Description of Project Structure Definition File.....	100
8.	How to Use the Indication Message Viewer.....	101
8.1.	Download reference files of project analysis results.....	101
8.2.	Startup .....	102
8.3.	Open the project analysis result reference file.....	102
8.4.	Confirm Messages.....	105
9.	Agile+ Relief diagnosis report generation tool .....	109
9.1.	Composition of Agile+ Relief diagnosis report.....	109
9.1.1.	Title .....	109
9.1.2.	General diagnosis report.....	110
9.1.3.	List of indication messages diagnosis report.....	111
9.1.4.	Source code metrics diagnosis report.....	112
9.2.	Preparation of Agile+ Relief diagnosis report generation tool use ..	112
9.2.1.	Customize Agile+ Relief Diagnosis report template.....	112

9.3.	Generate Agile+ Relief diagnosis report.....	114
9.4.	Cautions .....	114
10.	Improvements.....	115
10.1.	Improvement in V.1.1.1.....	115
10.1.1.	Support to C/C+.....	115
10.1.2.	Correspondence to the latest OS/MW environment.....	115
11.	To the customers using V1L10.....	116
11.1.	Change of Project Structure Definition File.....	116

# 1. What is Agile+ Relief?

## 1.1. Summary

### 1.1.1. Product Objectives

In agile development, according to the concept of self-discipline improvement and iterative development, team management is based on optimizing and changing quality standards. Therefore, in a large-scale project that has introduced Agile, it is necessary to introduce a unified and quantitative criteria management as overall quality control without affecting the activities of each team.

Agile+ Relief provides unified and quantitative criteria management as overall quality control.

### 1.1.2. Usage scenarios

The users and usage scenarios envisaged about Agile+ Relief are as follows.

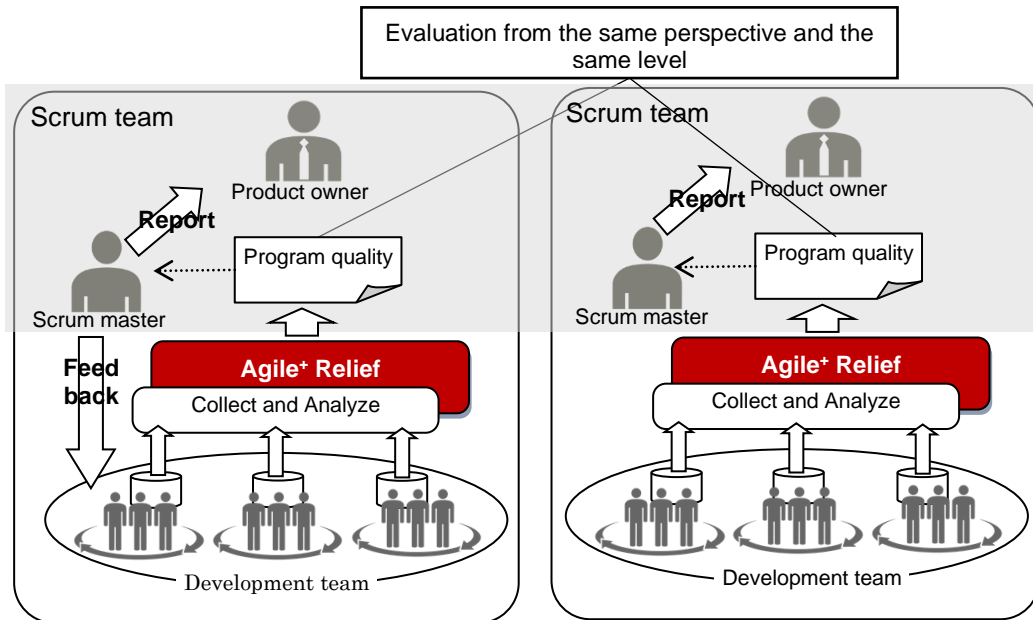
The users are envisaged as follows.

- Product Manager (QMO)  
People who work with the product owner to confirm whether the product satisfied system quality requirements. Engaged in quality management activities [Quality Management Office] in organizational project management activities.
- Product Owner  
People who have a QMO-like role for the scrum team and make acceptance decisions on outputs.
- Scrum Master  
People who is responsible for ensure that the development team run sprint smoothly. They explain the current status of the development team to the product owner to gain their understanding. They also explain the substance and purpose of the development process to stakeholders to gain their understanding.
- Development Team  
Responsible for developing software and providing software by sprint basis. Determine and implement new adoption and improvement methods of development process and development method.

The usage scenarios are envisaged as follows.

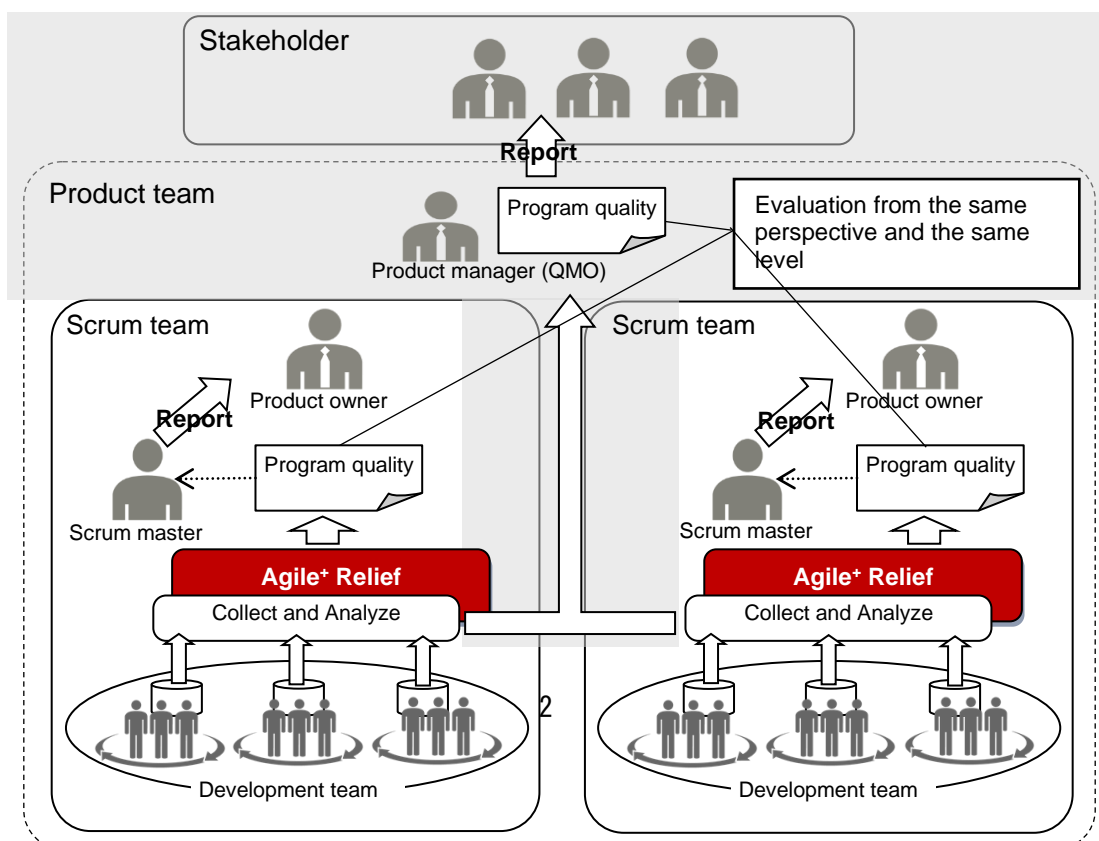
- Scene 1: Quality evaluation of the scrum team

Agile+ Relief is used to the program quality evaluation based on uniform and quantitative standards within each scrum team. Agile+ Relief realizes unified and quantitative standard managements as quality control of the scrum team by collecting and analyzing the program quality information of the development team.



- Scene 2: Quality evaluation of the entire project

Evaluate the program quality based on uniform and quantitative standards of the product by collecting and analyzing the program quality information of each scrum team mentioned above.



### 1.1.3. Functions

Linking with Agile+ Relief C/C++ or Agile+ Relief J, Agile+ Relief provides a continuous supervision of quality data for the analysis of source program in C/C++ or Java and puts forward a visual display for the quality defects.

The provided functions are described as follows:

#### 1) Define Check Base

Define the unified and quantitative Check Base within the organization of quality management and perform the check according to the predefined check base.

Check Base means: the rules for check, dangerous fields of metrics information, and the indices for quality status judgment according to the indication results.

The scrum master /product owner /product manager can grasp the evaluation (quality status) based on established check base easily. Meanwhile, since the required indication messages are to be output, the developers can confirm the indication messages correctly and exhaustively.

#### 2) Generation of HTML File for Quality Info and Evaluation Result

According to check base, the evaluation will be done to the quality data after the analysis of source program and include file, and the result will be generated in the form of HTML file.

For the format of HTML file, there is no need for special software. Generated contents include daily quality status according to check base, the disagreement between source program and indication messages for check.

#### 3) Execute Command Uniformly (arqm command)

Here, it provides two functions. One is [source analysis mode], that is to input source program to perform analysis by Agile+ Relief C/C++ or Agile+ Relief J and consolidation of analysis result. The other is [analysis result consolidation mode], that is to input the analysis result of Agile+ Relief C/C++ or Agile+ Relief J analysis command (pgr5, pgrjava etc.) to perform only the consolidation of analysis result.

That means in [source analysis mode], even in the case of increase or reduction of source program and project, once the settings are done there is no need to modify for the execution of command. In this way, no more settings are required, and no check miss might occur for development asset. It also works with version control tools to automatically retrieve the latest target source programs.

In [analysis result consolidation mode], analysis is performed using the analysis command (pgr5, pgrjava etc.) of Agile+ Relief C/C++ or Agile+ Relief J. Therefore, multiple analysis can be performed at the same time, so that the processing time from analysis to result consolidation will be reduced (\*1). Thus, the result of large-scale asset that has long processing time of result consolidation can be viewed on time.

(\*1) When executing multiple analysis processes at the same time, correspondent number of Agile+ Relief C/C++ license or Agile+ Relief J license are required.

#### 4) Settings for Recording the Processing

For the messages detected by batch execution command, two approaches for recording the



necessity of source program processing are provided.

One approach is: Through the indication message viewer, judge whether the source program processing is required for detected messages. If it does not require processing the source program, then it is recorded as "Not require processing". Note: Indication message viewer can be used in Agile+ Relief C/C++ only.

Another approach is: Set to "Before Managed" status by batch execution command (arqm command). This status indicates that source programming processing is not required for the last message of last time.

Even if the source program has been modified, the recording that mentioned above will be continued. Therefore, it is possible to judge that no further confirmation is required in the future work according to the messages marked as "Not require processing" and "Before Managed".

#### 5) Diagnosing report generation

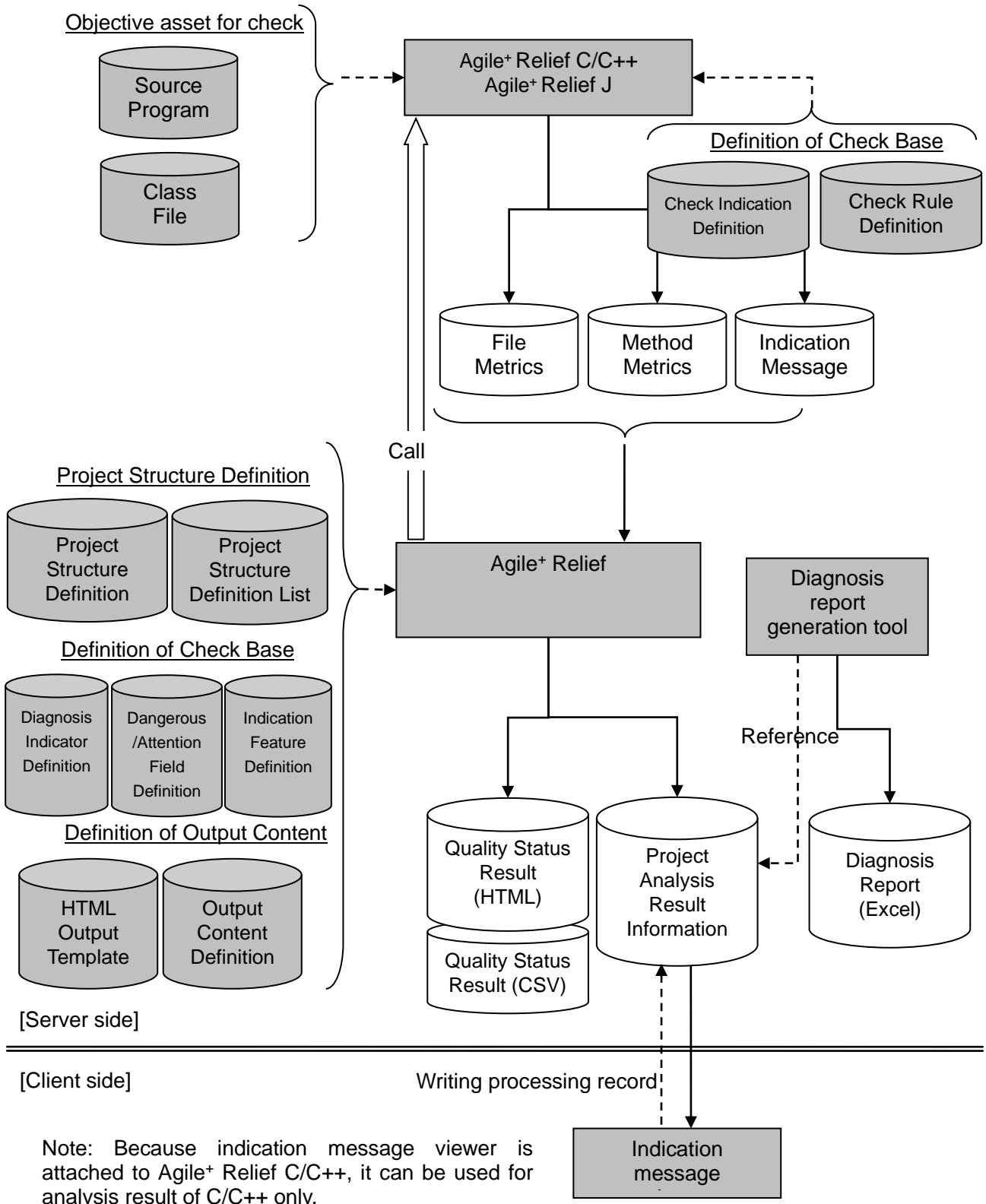
The quality data that analyzes the source program is diagnosed by the following viewpoints, and the result is generated as a diagnosing report.

- Presence of unconfirmed problem part
- Detection tendency to message seen from quality feature and problem feature
- Maintainability/readability seen from source metrics
- Action situation of detected message
- Correlation of File, function/method metrics, and message
- Action situation of warned file and method metrics violation items

The developer can also confirm not only pointed out part but also the problem part along the diagnose result of the above-mentioned viewpoint and can improve the quality of the source program by the manager's generating the diagnosing report and distributing it to the developer. The diagnosing report can be used as material that proves the quality of the source program by recording the result of the confirmation of the developer along the diagnosing report at the same time.

## 1.2. Structure of System and I/O File

The structure diagram of the command provided by Agile+ Relief, tools and the input and output files are shown as follows.



The summaries of various definition files are described as follows. For Details, please refer to "[4. Description of Various Definition Files](#)".

And for details of arqm command, please refer to "[5. How to use arqm command](#)".

#### Project Structure Definition

File Name	Description
Project Structure Definition File	The definition is located at the top folder of the objective asset for check. The folder under the designated folder is recognized as a project. In addition, the analysis related options are also defined.
Project Structure Definition Listing File	Define multiple project structure definition files. For the execution of project definition files with different folders as objective assets for check, and the output of quality info.

#### Definition of Check Base

File Name	Description
Check indication definition File or Check rule definition File	Define the rules for check. "Check indication definition File" is name for Agile+ Relief C/C++ while "Check rule definition File" is for Agile+ Relief J.
Dangerous /Attention Field Definition File	Define the scope of dangerous field/field for attention when performing the judgment of source program, such as nest count, complexity.
Diagnosis Indicator Definition File	Define the indices of quality status diagnose according to the definition of rules for check, dangerous field/field for attention.
Indication feature definition file	Define the value of quantitative judgment for the number of indications calculated for each feature.

#### Definition of Output Content

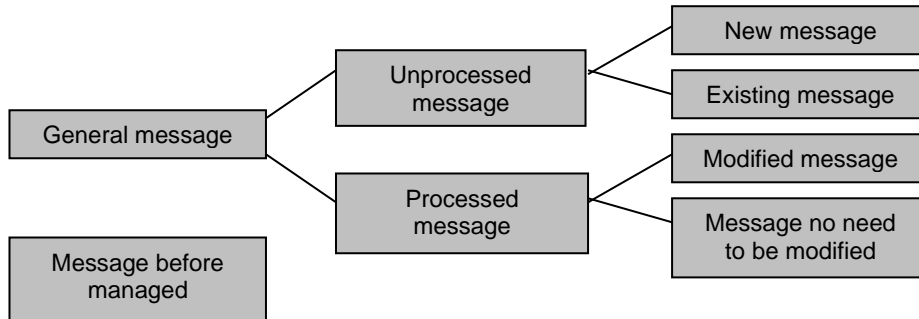
File Name	Description
HTML Output Template File	Define the format (Title, Layout, etc.) of HTML file for output.
Definition File of Output Content	Define the items of various lists for output, and the icon file names displayed in diagnose result.

Among the output files, in addition to the quality info results that are to be referenced by users, project analysis result information for creating quality info results is also included. The execution results of batch execution command (arqm command) are accumulated in project analysis result information. Please refer to "[5.4 Project Analysis Result Information](#)" for details.

## 2. How to use Agile+ Relief

### 2.1. Classification of Messages

Messages detected in Agile+ Relief can be classified into the following types.



#### General message

Indicates all detected messages except the messages before managed.

#### Unprocessed message

Indicates messages require processing.

#### New message

Indicates newly detected messages.

#### Existing message

Indicates previously detected messages.

#### Processed message

Indicates messages that have been processed.

#### Modified message

Indicates previous messages that has modified source program.

#### Message no need to be modified

Indicates previous messages of suppressed source program. This includes the messages that have been processed in suppression command.

For details of suppression command, please refer to "[2.6 Suppress comment](#)".

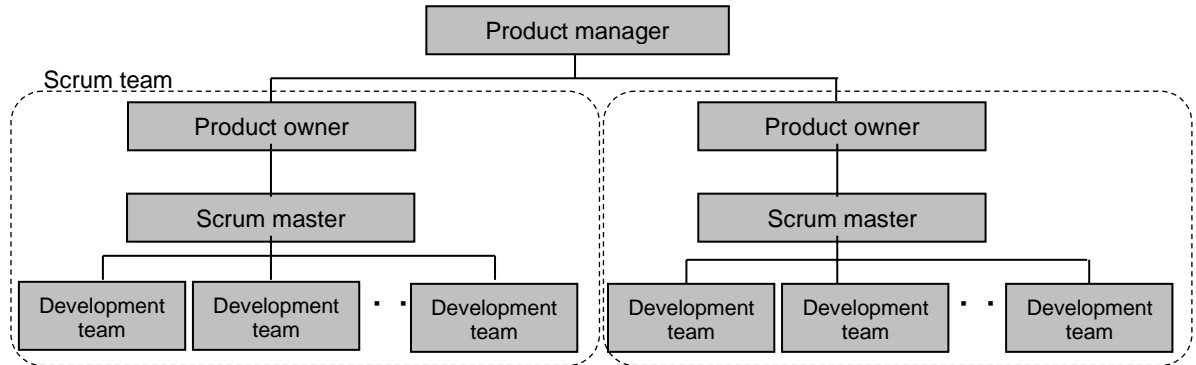
#### Message before managed

Indicates the messages of existing sources program that are set to not require processing by batch execution command (arqm command).

## 2.2. Organization Mechanism

About the user or organization for Agile+ Relief usage, the preliminary mechanism is provided in this document.

About the procedures applied in this mechanism, the illustration will be made after "[2.3 Procedures](#)".



### Product manager (QMO)

Person for quality management of multiple scrum teams.

### Product owner/Scrum master

Person for quality management of due scrum team.

### Development team

Program development team.

## 2.3. Procedures

There are two processing modes of Agile+ Relief.

### source analysis mode

Input the source program of C/C++, Java to perform analysis and consolidation of analysis result.

### analysis result consolidation mode

Input the analysis result of analysis command (pgr5, pgrjava etc.) and perform only the consolidation of analysis result.

The procedures for the Agile+ Relief, such as settings for various definition file, execution, source program correction is described as follows according to different output mode. The illustration will be provided in the order of person responsible for operation.

### 2.3.1. Procedures under source analysis mode

Perform the following preparations when using under the source analysis mode.

Order	Operation	Pre-operator	Summary
1	Setting Check Base	Product manager (QMO)	Define the base of Agile+ Relief use.
2	Setting Output Content	Product manager (QMO)	Define the content of HTML and CSV to output.
3	Setting Objective for Check	Product owner	Define the source program of objective for check.
4	Setting message before managed	Scrum master Product owner	Execute batch execution command (arqm command) and set the message to "Before Managed".

Perform the following operations repeatedly after preparations.

Order	Operation	Pre-operator	Summary
5	Output quality info results	Scrum master Product owner	Execute batch execution command (arqm command), and output quality info results.
6	Confirm quality info results	Development team Scrum master Product owner Product manager (QMO)	Confirm the quality info results that are output.
7	Process messages	Development team	Judge if the detected messages needs to be modified.
8	Combine the quality info results of multiple organizations	Product manager (QMO)	Combine the quality info results that are output separately into 1 result.

#### 2.3.1.1. Setting Check Base

Product manager (QMO) defines the guarding base for Agile+ Relief use in organization. Create following four types definition files according to the status of organizations (Create or

improve, whether the coding manners are applied).

#### Definition File of Indication Messages for Check, Definition File of Rules for Check

Define the rules for organization.

About how to define, please refer to "[4.2.1 Check indication definition File and Check rule definition File](#)".

#### Definition File of Dangerous Field/Field for Attention

Define the scope of dangerous field and field for attention when performing the judgment of metrics value (nest count, complexity, etc.).

About how to define, please refer to "[4.2.2 Dangerous /Attention Field Definition file](#)".

#### Definition File of Diagnose Indices

Define the indices of comprehensive diagnose according to the results of definition file of indication messages for check, definition file of rules for check and dangerous field/field for attention.

About how to define, please refer to "[4.2.3 Diagnosis Indicator Definition File](#)".

#### Indication feature definition file

Define the value of quantitative judgment for the number of indications calculated for each feature.

About how to define, please refer to "[4.2.4 Indication feature definition file](#)".

### 2.3.1.2. Setting Output Content

After the confirmation of product manager (QMO) for the contents of HTML and CSV to output, the following output related files should be updated.

#### HTML Output Template File

For the files output in HTML, the layout and description should be updated for the organization.

About how to define, please refer to "[3.5 Template File Update](#)".

#### Definition File of Output Content

Define the following content for output: Items of list for HTML files and CSV files to output, icon file names corresponding to the diagnose results output in HTML files.

About how to define, please refer to "[3.6 Description of Definition File for Output Content](#)".

### 2.3.1.3. Setting Objective for Check

Product owner judges which source program requires check using Agile+ Relief C/C++ or Agile+ Relief J

among the source programs saved on the server and create the following files:

#### Project Structure Definition File

Select the top folder, in which the objective source program file for check is saved. The folder under the designated folder will be recognized as one project. Even in the case of increase and

deduction of source program files in project, no modification upon settings is required.  
In addition, please set the analysis related options.  
About how to define, please refer to "[4.1.1 Project Structure Definition File](#)".

#### 2.3.1.4. Setting message before managed

For the existing source program on which the running test has been done, and which is about to apply Agile+ Relief shortly etc., the messages of existing source program do not need to be processed sometimes. At this moment, set the status of these messages to "Before Managed". This operation is performed by the scrum master/project owner through executing the batch execution command (arqm command) in the following steps.

##### Execute Analysis

Analyzing the source program through batch execution command (arqm command) and detect messages.

Specify each type of definition file mentioned before and the output target folder for the arqm command.

For batch execution command (arqm command), please refer to "[5. How to use arqm command](#)".

##### Set "Before Managed"

Through batch execution command (arqm command), set the message of existing source program to "Before Managed".

#### 2.3.1.5. Output quality info results

The source program is updated, arqm command is executed by the scrum master/product owner, and quality info results are output. Specify each type of definition file mentioned before and the output target folder for the arqm command.

For arqm command, please refer to "[5. How to use arqm command](#)".

#### 2.3.1.6. Confirm quality info results

Confirm the quality info results that are output. Please refer to "[3. Contents of Quality info Result](#)" for output details.

Product manager (QMO) checks the quality info of due multiple scrum teams for exception.

Product owner and scrum manager check the quality info of due scrum team for exception.

Development team will confirm whether the source program that violates the check base exists and process the message if it exists.

#### 2.3.1.7. Process messages

For the detected messages, if the development team judges it as requiring processing, then modify the source program.

User can confirm and record whether the indications are to be processing or not by using the indication message viewer when analyzing C/C++ with Agile+ Relief C/C++.

For indication message viewer, please refer to "[8. How to Use the Indication Message](#)".



[Viewer](#)" and HELP.

### 2.3.1.8. Combine the quality info results of multiple organizations

The scrum master /product owner need to combine and evaluate the quality status of due multiple development teams, and the product manager need to combine and evaluate the quality status of due multiple scrum teams. After the creation of project structure definition listing file, which is used for combined output of multiple quality info results, the arqm command will be executed.

#### Project Structure Definition Listing File

Set the name of a project structure definition file after combination of multiple project structure definition files. The project structure definition files of C/C++ and Java can be mixed.

About the description, please refer to "[4.1.2 Project Structure Definition Listing File](#)".

### 2.3.2. Procedures under analysis result consolidation mode

Perform the following preparations when using under the analysis result consolidation mode.

Order	Operation	Pre-operator	Summary
1	Setting Check Base	Product manager (QMO)	Define the base of Agile+ Relief use.
2	Setting Output Content	Product manager (QMO)	Define the content of HTML and CSV to output.
3	Extracting Consolidation Object	Product owner	Extract the file to be consolidated.
4	Setting Consolidation Object	Product owner	Define the asset to be consolidated.
5	Setting message before managed	Scrum master Product owner	Execute analysis command (pgr5, pgrjava etc.) and batch execution command (arqm) of Agile+ Relief and set the message to "Before Managed".

Perform the following operations repeatedly after preparations.

Order	Operation	Pre-operator	Summary
6	Output quality info results	Scrum master Product owner	Execute analysis command (pgr5, pgrjava etc.) and batch execution command (arqm) of Agile+ Relief, and output quality info results.
7	Confirm quality info results	Development team Scrum master Product owner Product manager (QMO)	Confirm the quality info results that are output.
8	Process messages	Development team	Judge if the detected messages needs to be modified.
9	Combine the quality info results of multiple organizations	Scrum master Product manager (QMO)	Combine the quality info results that are output separately into 1 result.

#### 2.3.2.1. Setting Check Base

Product manager (QMO) defines the guarding base for Agile+ Relief use in organization.

About settings, please refer to ["2.3.1.1 Setting Check Base"](#).

#### 2.3.2.2. Setting Output Content

After the confirmation of product manager (QMO) for the contents of HTML and CSV to output, the output related files will be updated.

About settings, please refer to ["2.3.1.2 Setting Output Content"](#).

### 2.3.2.3. Extracting Consolidation Object

Product owner judges which source program requires check using Agile+ Relief C/C++ or Agile+ Relief J among the source programs saved on the server and extract the analysis result as consolidation object asset.

The files extracted as consolidated asset are as follows.

#### File metrics file and function/method metrics file [Required]

This is the file that saves the metrics information of check object source program.

The extension of file metrics file is ".mfile" and the extension of function/method metrics file is ".mfunc".

#### Indication message file [Required]

This is the file that saves the check result of check object source program (indication message).

The indication message file does not have fixed extension. When getting the result, please add any extension.

#### Path file

This is the file that saves the relevant information accompanied with the exploring path of cross file indication message. It is output when checking the C source program with Agile+ Relief C/C++ Wide-ranging Detective option. The extension of path file is ".srinf".

#### MISRA check result file and SEC check result file

This is the file that saves the MISRA violation and IPA/SEC-C/C++ violation of C/C++ source program.

It will be output when using Agile+ Relief C/C++ MISRA option or IPA/SEC-C/C++ check function to check C/C++ source program. The extensions of MISRA check result file and SEC check result file are not fixed. When getting the result, please add any extension.

#### **[Reference]**

The following describes the method of getting consolidation object asset.

- Use the pgr5 + pgrmetrics + pgrmisra/pgrsec command to get consolidation object asset (Target at C/C++ source program)

- 1) Execute pgr5 command to get "Indication message file".

Example: pgr5 --qm sample1.c > sample1.c.message  
(\*1) (\*2)

(\*1) The --qm option can output result with required format according to the analysis consolidation mode. Please make sure to specify.

(\*2) The pgr5 command outputs analysis result according to standard output. When using the analysis result consolidation mode, please make sure to redirect the analysis result to file.

\* If the Agile+ Relief C/C++ Wide-ranging Detective option has been imported, when executing the pgr5 command, the "path file" can be obtained at the same time. Since "path file" is recorded together with the indication related to "indication message file", so the Agile+ Relief can identify it automatically.

(When the indication of Wide-ranging Detective does not exist, "path file" will not be output)

- 2) Execute the pgrmetrics command and get "File metrics file" and "Function metrics file".

Example: pgrmetrics -o sample1.c sample1.c.ao

- 3) Execute MISRA violation and IPA/SEC-C/C++ violation check as needed and get "MISRA check result file" and "SEC check result file".

Example: pgrmisra --qm -T1 sample1.c.message > sample1.c.misra  
(\*1) (\*2)

or

pgrsec --qm -T1 sample1.c.message > sample1.c.sec  
(\*1) (\*2)

(\*1) The --qm option can output result with required format according to the analysis consolidation mode. Please make sure to specify.

(\*2) The pgrmisra / pgrsec command outputs analysis result according to standard output. When using the analysis result consolidation mode, please make sure to redirect the analysis result to file.

After executing the above command, the following consolidation object asset can be obtained.

File metrics file	: sample1.c.mfile
Function metrics file	: sample1.c.mfunc
Indication message file	: sample1.c.message
MISRA check result file	: sample1.c.misra
SEC check result file	: sample1.c.sec

- Use the pgrfake + pgrmetrics command to get consolidation object asset  
(Target at C/C++ source program)

- 1) Get "indication message file" through replacing the calling command of compilers such as make file or batch file with the pgrfake.

Example: Replace the command in the make file with the pgrfake command

```
:  
pgrfake /D WIN sample1.c  
pgrfake /D WIN sample2.c  
:
```

(\*) Before executing the pgrfake command, it is needed to confirm whether the file generated through "command parameter conversion definition file" has been set as not to delete. If it is not set, when the file to be input in pgrmetrics will be deleted, and the expected analysis result cannot be obtained.

Example of setting command parameter conversion definition file

```
[PGRCOMMAND]  
:  
DELETEANAFILE = false  
:
```

- 2) Execute the pgrmetrics command and get "File metrics file" and "Function metrics file".

Example: pgrmetrics -o sample \*.ao

After executing the above command, the following consolidation object asset can be obtained.

```
File metrics file      : sample.mfile
Function metrics file  : sample.mfunc
Indication message file : sample1.c.message, sample2.c.message, ...
```

- Use the pgrjava command to get consolidation object asset  
(Target at java source program)

- 1) Execute the analysis command of Agile+ Relief J (pgrjava) and get "indication message file", "file metrics file" and "method metrics file".

Example: pgrjava.bat -qm C:prj1¥src C:¥prj1¥bin C:¥prj1¥prj1.message  
(\*1)

(\*1)-qm option can output result with required format according to the analysis consolidation mode. Please make sure to specify.

After executing the above command, the following consolidation object asset can be obtained.

```
File metrics file      : prj1.message.mfile
Function metrics file  : prj1.message.mfunc
Indication message file : prj1.message
```

<Notes during command execution>

- Before executing the analysis command, please make sure to specify -qm option or --qm option.

Command name	Option to specify
pgr5	--qm
pgrmisra	--qm
pgrsec	--qm
pgrjava	-qm

- For the output code of consolidation object asset, in case of Windows, please use SJIS/MS932 code. In case of Red Hat Enterprise Linux 5/6, please use UTF-8 command. In case of others, please use EUC code.
- When outputting "indication message file", "MISRA check result file", "SEC check result file" through the analysis command, please output after specifying extensions individually.
- In the folder that saves the consolidation object asset that is specified through TARGET key of the project structure definition file, please do not mix the analysis result of C/C++ with the analysis result of java.
- In the folder that saves the consolidation object asset that is specified through TARGET key of the project structure definition file, please do not mix the analysis result output in Japanese with the analysis result output in English.

For illustration of each command, please refer to the following manuals.

pgr5 command	: [Agile+ Relief C/C++ Command Manual]
pgrmetrics command	: [Agile+ Relief C/C++ Command Manual]
pgrmisra command	: [Agile+ Relief MISRA Option Manual]
pgrsec command	: [Agile+ Relief IPA/SEC-C/C++ Check]
pgrfake command	: [Agile+ Relief C/C++ Command Manual]
pgrjava command	: [Agile+ Relief J Command Manual]

#### 2.3.2.4. Setting Consolidation Object

Product owner sets the following file as consolidation object, such as where was the consolidation object asset extracted before are saved on the server etc.

##### Project Structure Definition File

Set the top folder, in which the consolidation object assets are saved. The folder under the designated folder will be recognized as one project.

About the description, please refer to "[4.1.1 Project Structure Definition File](#)".

#### 2.3.2.5. Setting message before managed

For the existing source program on which the running test has been done, and which is about to apply Agile+ Relief shortly etc., the messages of existing source program do not need to be processed sometimes. At this moment, set the status of these messages to "Before Managed". This operation is performed by the scrum master through executing the batch execution command (arqm command) in the following steps.

##### Execute Analysis

Analyzing the source program through analysis command of Agile+ Relief C/C++ or Agile+ Relief J (pgr5, pgrjava etc.), and detect messages.

##### Execute Consolidation

Through batch execution command (arqm command), consolidate the analysis result of analysis command.

Specify each type of definition file mentioned before and the output target folder for the arqm command.

For batch execution command (arqm command), please refer to "[5. How to use arqm command](#)".

##### Set "Before Managed"

Through batch execution command (arqm command), set the message of existing source program to "Before Managed".

#### 2.3.2.6. Output quality info results

The source program is updated, analysis command (pgr5, pgrjava etc.) and arqm command of Agile+ Relief are executed by the scrum master/product owner, and quality info results are output. Specify each type of definition file mentioned before and the output target folder for the

arqm command.

For arqm command, please refer to "[5. How to use arqm command](#)".

### 2.3.2.7. Confirm quality info results

The product manager (QMO), product owner, scrum master and development team confirm the quality info result.

For confirmation content, please refer to "[2.3.1.6 Confirm quality info results](#)".

### 2.3.2.8. Process messages

The development team processing the message that has been output.

For processing content, please refer to "[2.3.1.7 Process messages](#)".

### 2.3.2.9. Combine the quality info results of multiple organizations

The scrum master /product owner need to combine the quality status of due multiple development teams, and the product manager need to combine the quality status of due multiple scrum teams. For combined content, please refer to "[2.3.1.8 Combine the quality info results of multiple organizations](#)".

## 2.4. Execute Sample Project

The following sample project is used to illustrate how to use Agile+ Relief.

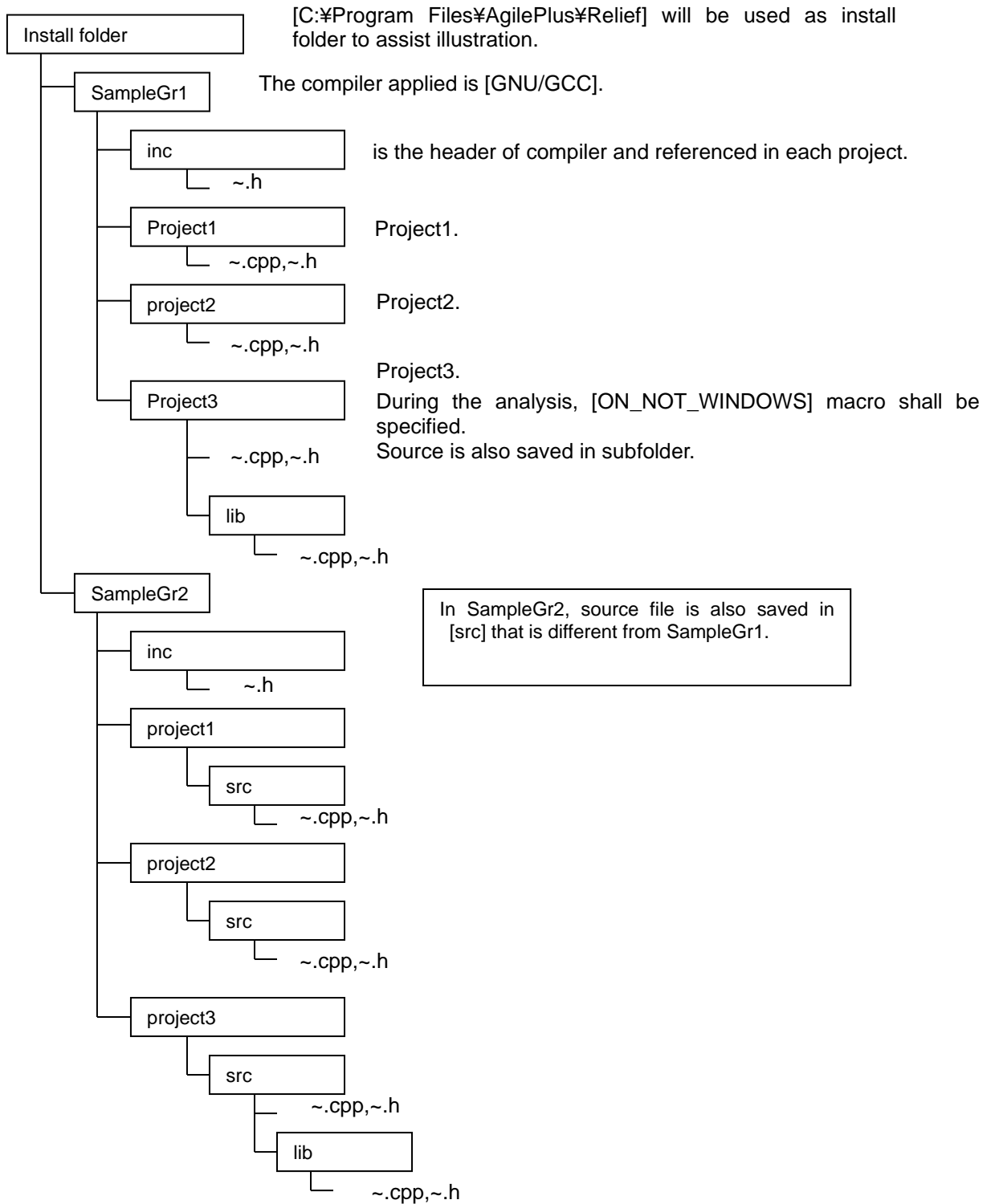
The following sections store the files in 'C: ¥ARQMDEMO' for execution confirmation. And the various files created will be saved in the [ARQMDEMO] folder which is in the install folder of Agile+ Relief.

### 2.4.1. Source analysis mode - Execute C/C++ Sample Project

The use of Agile+ Relief is described by using the sample project written in C/C++.

### 2.4.1.1. Structure of sample project

Sample project is saved under the installation directory of Agile+ Relief as follows.





### 2.4.1.2. Create check base

Create check indication definition file, dangerous /attention field definition file, diagnosis indicator definition file and indication feature definition file as the standard to be complied with by organizations when using Agile+ Relief.

#### Check indication definition File

Define the indication messages for check as the definition file of a group message, pgr0637(b group), and pgr0638(b group), and create in the folder of [C:¥ARQMDEMO¥check] as the file name of [PGReliefID.def].

The template is saved in the name of [set\_messageindication.def] in the [definition\_template] folder under the install folder of Agile+ Relief.

```
;Rule=PGRelief
a
0637
0638
```

#### Dangerous /Attention Field Definition file

Modify the initial value of [Times for Execution] of metrics value for check. Create the definition file of dangerous field/field for attention with the dangerous field of [Times for Execution] to be [200], while [Field for Attention] to be [100], in the file name of [Metrics.def], and save into the folder of [C:¥ARQMDEMO¥check].

Other items are left with initial values. The initial value of [Times for Execution] is [400] for dangerous field, and [200] for field for attention.

The template is saved in the name of [set\_cautionval.def] in the [definition\_template] folder under the install folder of Agile+ Relief. The initial values are already described in the template.

```
;line, Japanese label, English label, Invalid or Valid flag, Value for Attention, Value of
Dangerous Field
3,Total Lines, Number of total lines, TRUE, 400,600
4,Run Lines, Number of execution lines, TRUE, 100,200
5,Nest Count, Number of nesting levels,TRUE,10,20
6,Complexity 1,Complexity1,TRUE,20,50
7,Complexity 2,Complexity2,FALSE,20,50
8,goto,goto,TRUE,1,2
9,return,return,FALSE,10,20
10,break,break,FALSE,10,20
11,continue,continue,FALSE,10,20
12,if1,if1,TRUE,10,20
13,while1,while1,TRUE,10,20
14,dowhile1,dowhile1,TRUE,10,20
15,for1,for1,TRUE,10,20
16,switch1,switch1,TRUE,10,20
17,if2,if2,FALSE,10,20
18,while2,while2,FALSE,10,20
19,dowhile2,dowhile2,FALSE,10,20
20,for2,for2,FALSE,10,20
21,switch2,switch2,FALSE,10,20
```

#### Diagnosis indicator definition file

The definition file of diagnostic indices, in which NOGOOD is specified for [Indicated More Than 10 Times], or[Dangerous Alert More Than 5 Times], PASSBLE is specified for [Indicated More

Than Once], or [Dangerous Alert More Than Once], is saved in the name of [Diagnose.def] to the folder of [C:¥ARQMDEMO¥check].

The template is saved in the name of [set\_diagnosisindicat.def] in the [definition\_template] folder under the install folder of Agile+ Relief.

```
NOGOOD = INDICATION_COUNT >= 10, DANGER_COUNT >= 5
PASSABLE = INDICATION_COUNT >= 1, DANGER_COUNT >= 1
```

Indication feature definition file

Modify the indicator value of [Credibility] of [Quality feature] to [0.8] for dangerous range, [0.5] for cautious field.

The template is saved in [characteristic\_c] of [qualityinfo] under the install folder of Agile+ Relief. Copy the template to [qualityinfo], under the installation folder of Agile+ Relief, and update the Indication feature definition file [quality.pcs] in correspondence to [Quality Feature].

[CHARACTERISTIC-ID]				
;Characteristic-ID,	Item Name(english),	Item Name(japanese),	Danger Value,	Warning Value
;----- ----- ----- ----- -----				
Reliability	Reliability,	Reliability,	<b>0.8,</b>	<b>0.5</b>
Maintainability,	Maintainability,	Maintainability,	10,	5
Portability,	Portability,	Portability,	10,	5
Efficiency,	Efficiency,	Efficiency,	10,	5

### 2.4.1.3. Create project structure definition file

The project structure definition file of sample [SampleGr1] is saved in the name of [SampleGr1.pgs] to the folder of [C:¥ARQMDEMO¥Sample] as follows: The content is as follows.

The template is saved in the name of [project.pgs] in the [definition\_template] folder under the install folder of Agile+ Relief.

[SOURCES]	<- For the section auto recognized in project	
TARGET =	C:¥Program Files¥AgilePlus¥Relief¥SampleGr1	1
BINDPROJECT =	Project3	2
CPLUSPLUSEXT =	.cpp	3
[OPERATECOMMONPROJECT]	<-common section in project	
OPTION =	-KGNU/GCC -Y"C:¥Program Files¥AgilePlus¥Relief¥SampleGr1¥inc"	4
RULEFILE	= C:¥ARQMDEMO¥check¥PGReliefID.def	5
METRICSFILE	= C:¥ARQMDEMO¥check¥Metrics.def	6
INDICATORFILE	= C:¥ARQMDEMO¥check¥Diagnose.def	7
[OPERATEPROJECT]	<- unique section in project	
PROJECTTARGET =	Project3	8
APPENDOPTION	= -DON_NOT_WINDOWS	9

1. Specify the top directory with the asset to be checked being saved.
2. The source file to be saved in [lib] folder is designated to be included in project [Project3]. Otherwise, create project [lib].
3. Specify [cpp] as the extension of C++ source file to be checked.
4. Specify this analysis option. In this sample, specify the compiler type (-K) and standard include (-Y).
5. Specify the check indication definition file.
6. Specify the dangerous/attention field definition file.
7. Specify the diagnosis indicator definition file.
8. Because [ON\_NOT\_WINDOWS] macro is required during the analysis of project [Project 3], thus specify: the information needs to be designated in OPREATEPROJECT is project [Project3].
9. Specify the unique analysis option of project [Project 3].

### 2.4.1.4. Execute Batch Execution Command

Execute batch execution command (arqm command) according to the following format:

Note: Line feeder is found in the following description, while do not change the line order for actual command execution.

"C:¥Program Files¥AgilePlus¥Relief¥arqm"	1
-o "C:¥ARQMDEMO¥HTML"	2
"C:¥ARQMDEMO¥Sample¥SampleGr1.pgs"	3

1. Specify a command.
2. Specify the output directory of quality info.
3. Specify the project structure definition file name.

#### 2.4.1.5. Execute Sample of Different Folder Structure

Another project structure definition file of sample [SampleGr2] is saved on the file name of [SampleGr2.pgs] to the folder of [C:¥ARQMDEMO¥Sample].

All source files of project in sample [SampleGr2] are saved in the name of folder [Src]. Thus when the contents of project structure definition file is still [SampleGr1.pgs], all project names will be changed to [Src]. Then, the key [PROJECTLEVEL] will be applied.

The content is as follows.

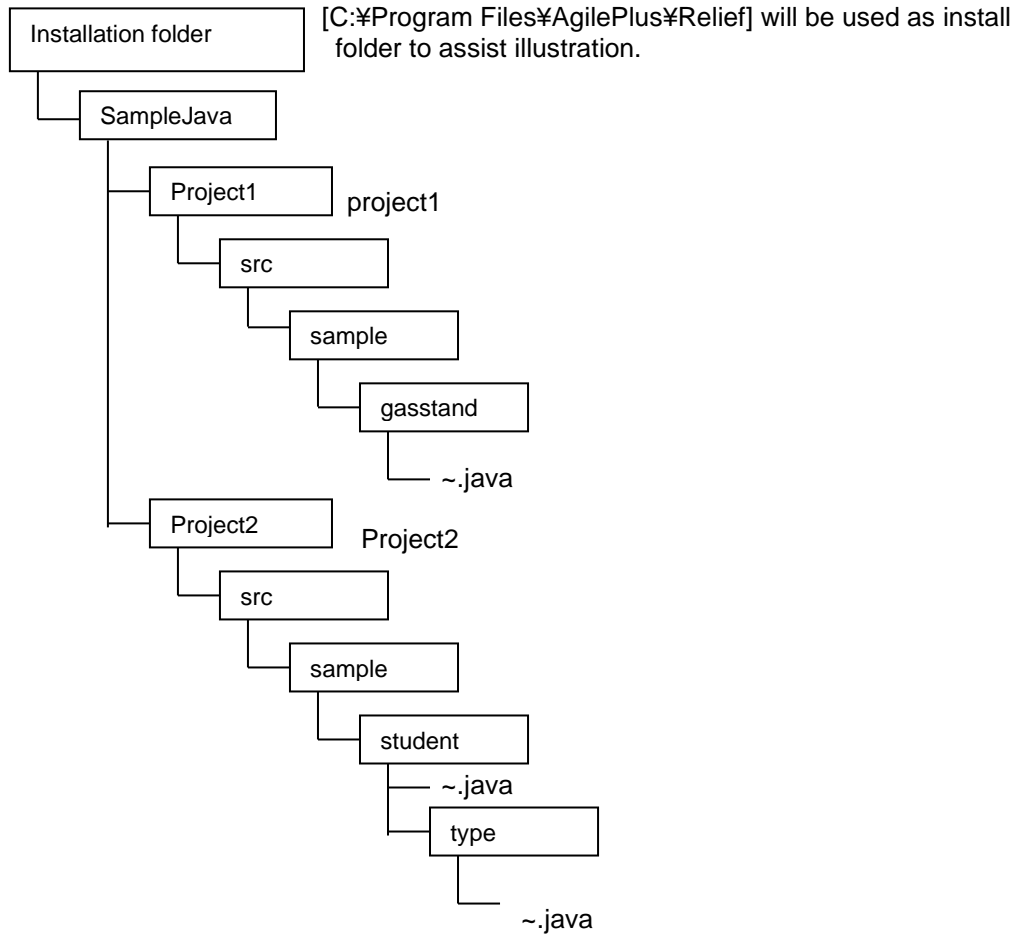
```
[SOURCES]
TARGET = C:¥Program Files¥AgilePlus¥Relief¥SampleGr2
PROJECTLEVEL = 1
CPLUSPLUSEXT = .cpp
[OPERATECOMMONPROJECT]
OPTION = -KGNU/GCC -Y"C:¥Program Files¥AgilePlus¥Relief¥SampleGr2¥inc"
RULEFILE      = C:¥ARQMDEMO¥check¥PGReliefID.def
METRICSFILE   = C:¥ARQMDEMO¥check¥Metrics.def
INDICATORFILE = C:¥ARQMDEMO¥check¥Diagnose.def
[OPERATEPROJECT]
PROJECTTARGET = Project3
APPENDOPTION  = -DON_NOT_WINDOWS
```

## 2.4.2. Source analysis mode - Execute Java Sample Project

The use of Agile+ Relief is described by using the sample project written in Java. The sample of Agile+ Relief J is used in sample project.

### 2.4.2.1. Structure of sample project

Sample project is saved under the installation directory of Agile+ Relief as follows.



### 2.4.2.2. Create Check Base

Create check rule definition file, dangerous /attention field definition file, diagnosis indicator definition file and indication feature definition file as the standard to be complied with by organizations when using Agile+ Relief.

#### Check rule definition File

Customize using the rule of Agile+ Relief J and create check rule definition file with file name [rule.pgrj] and save directory [C:¥ARQMDEMO¥check].

The template is saved in [definition\_template], under the install folder of Agile+ Relief, with file name [default\_rule.pgrj].

In addition, for the use of rule customization, please refer to [Agile+ Relief J Customizer Operation Guide].

#### Dangerous /Attention Field Definition file

The creation method is same as that of C/C++. Please refer to the description in [Dangerous /Attention field definition file] of "[2.4.1.2 Create check base](#)".

#### Diagnosis indicator definition file

The creation method is same as that of C/C++. Please refer to the description in [Diagnosis indicator definition file] of "[2.4.1.2 Create check base](#)".

#### Indication feature definition file

Modify the indicator value of [Credibility] of [Quality feature] to [0.8] for dangerous range, [0.5] for cautious range.

The template is stored in 'characteristic \_ java' under the 'qualityinfo' folder in the Agile+ Relief installation folder. Copy the template into the Agile+ Relief installation folder 'qualityinfo' and update the Point-Of-Life definition file 'quality characteristic' corresponding to 'quality.pcs'.

[CHARACTERISTIC-ID]					
;	Characteristic-ID,	Item Name(english),	Item Name(japanese),	Danger Value,	Warning Value
;	-----	-----	-----	-----	-----
	Reliability,	Reliability,	Reliability,	<b>0.8,</b>	<b>0.5</b>
	Maintainability,	Maintainability,	Maintainability,	10,	5
	Portability,	Portability,	Portability,	10,	5
	Efficiency,	Efficiency,	Efficiency,	10,	5
	Functionality,	Functionality,	Functionality,	10,	5

### 2.4.2.3. Create project structure definition file

Create project structure definition file of sample [Sample] with file name [SampleJava.pgs] and save in [C:¥ARQMDEMO¥Sample] as the following. The content is as follows.

The template is saved in [definition\_template], under the install folder of Agile+ Relief, with file name [projectjava.pgs].

[SOURCES]	<-For the section auto recognized in project	
TARGET = C:¥Program Files¥AgilePlus¥Relief¥SampleJava		1
PROJECTLEVEL = 1		2
[OPERATECOMMONPROJECT]	<-common section in project	
LANG = Java		3
RULEFILE = C:¥ARQMDEMO¥check¥rule.pgrj		4
METRICSFILE = C:¥ARQMDEMO¥check¥Metrics.def		5
INDICATORFILE = C:¥ARQMDEMO¥check¥Diagnose.def		6

1. Specify the top directory with the asset to be checked being saved.
2. Specify the level number [1] that was obtained from the designated directory in TARGET1 to the directory that is recognized as project.  
Without this specification, it will be recognized as containing all directories of source file, and [gasstand], [student] and [type] will be recognized as project.
3. Specify Java as the language of check object asset.
4. Specify the check rule definition file.
5. Specify the dangerous/attention field definition file.
6. Specify the diagnosis indicator definition file.

#### 2.4.2.4. Execute Batch Execution Command

Execute batch execution command (arqm command) according to the following format:

"C:¥Program Files¥AgilePlus¥Relief¥arqm"	1
-o "C:¥ARQMDEMO¥HTML"	2
"C:¥ARQMDEMO¥Sample¥SampleJava.pgs"	3

1. Specify a command.
2. Specify the output directory of quality info.
3. Specify the project structure definition file name.

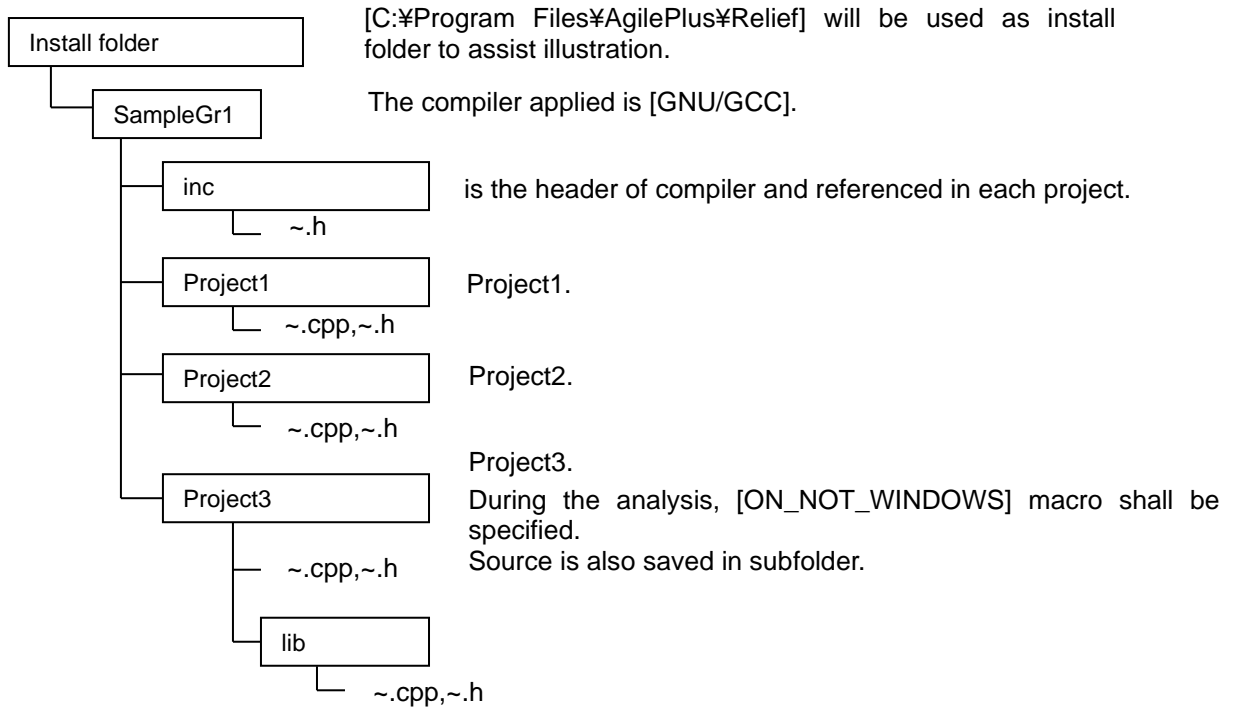
Note: Line feeder is found in the following description, while do not change the line order for actual command execution.

### 2.4.3. Analysis result consolidation mode - Execute Sample Project

The use of Agile+ Relief is described by using the sample project of analysis result consolidation mode. (The sample is described in C/C++)

#### 2.4.3.1. Structure of sample project

The sample project will be saved in install folder of Agile+ Relief as follows:



\* In this sample, SampleGr2 is not used.

#### 2.4.3.2. Create check base

Create check indication definition file, dangerous /attention field definition file, diagnosis indicator definition file and indication feature definition file as the standard to be complied with by organizations when using Agile+ Relief.

For information on how to create check base, please refer to "[2.4.1.2 Create check base](#)".



### 2.4.3.3. Analyze Source Program

Use the analysis command to analyze the asset under sample [SampleGr1] and save the analysis result under the [C:¥ARQMDEMO¥Sample¥AnaResult] folder as consolidation object asset.

In the sample, the batch file for executing the analysis command is created with the file name [SampleAna.bat], the folder for saving the file is [C:¥ARQMDEMO¥Sample], and the command is executed.

```
@echo off
setlocal

rem -----
rem  Set Demo environment
rem -----
set AR_INSTDIR=C:¥Program Files¥AgilePlus
set AR_DEMODIR=C:¥ARQMDEMO

rem -----
rem  Make directories
rem -----
if not exist "%AR_DEMODIR%¥Sample¥AnaResult"
    mkdir "%AR_DEMODIR%¥Sample¥AnaResult"
    :

rem -----
rem  Clear old files
rem -----
del /Q "%AR_DEMODIR%¥Sample¥AnaResult¥Project1¥*"
    :

rem -----
rem  Analyze
rem -----
rem Project1
"%AR_INSTDIR%¥PGRelief¥Analyze¥EPOM¥pgr5.exe" --qm -c -KGNU/GCC      1
-Y"%AR_INSTDIR%¥Relief¥SampleGr1¥inc"
-Z"%AR_DEMODIR%¥check¥PGReliefID.def"
--temp-file"%AR_DEMODIR%¥Sample¥AnaResult¥Project1¥sample1.c"
"%AR_INSTDIR%¥Relief ¥SampleGr1¥Project1¥sample1.c"
> "%AR_DEMODIR%¥Sample¥AnaResult¥Project1¥sample1.c.message"
    :
"%AR_INSTDIR%¥PGRelief¥Analyze¥EPOM¥pgrmetrics.exe"                    2
-o"%AR_DEMODIR%¥Sample¥AnaResult¥Project1¥prj1"
"%AR_DEMODIR%¥Sample¥AnaResult¥Project1¥*.ao"
    :
endlocal
@echo on
```

Note: Line feeder is found in the preceding description, while do not change the line order for actual command execution.

1. Set the pgr5 command.
  - Specify the --qm option, which can output according to the required format under analysis result consolidation mode.

- Specify the check indication definition file created in "[2.4.3.2 Create check base](#)" through the -Z option.
- Specify to output "indication message file" to [C:¥ARQMDEMO¥Sample¥AnaResult¥Project1¥sample1.c.message].

2. Set the pgrmetrics command.

- Specify to output "file metrics file" and "function metrics file" to [C:¥ARQMDEMO¥Sample¥AnaResult¥Project1¥prj1.mfile] and [C:¥ARQMDEMO¥Sample¥AnaResult¥Project1¥prj1.mfunc].

#### 2.4.3.4. Consolidate Analysis Result

Create project structure definition file to consolidate the analysis result obtained through "[2.4.3.3 Analyze Source Program](#)". Its file name is [SampleResultMerge.pgs] and the folder that saves the file is [C:¥ARQMDEMO¥sample]. The content is as follows.

The template is saved in the name of [project.pgs] in the [definition\_template] folder under the install folder of Agile+ Relief.

[SOURCES]	<-For the section auto recognized in project	
MODE	= RESULTMERGE	1
TARGET	= C:¥ARQMDEMO¥Sample¥AnaResult	2
[OPERATECOMMONPROJECT]		
METRICSFILE	= C:¥ARQMDEMO¥check¥Metrics.def	3
INDICATORFILE	= C:¥ARQMDEMO¥check¥Diagnose.def	4

1. Specify "analysis result consolidation mode" as the operation mode.
2. Specify the top folder, in which the consolidation object assets are saved.
3. Specify the dangerous/attention field definition file.
4. Specify the diagnosis indicator definition file.

After the project structure definition file has been created, execute batch execution command (arqm command) according to the following format.

"C:¥Program Files¥AgilePlus¥Relief¥arqm"	1
-o "C:¥ARQMDEMO¥HTML"	2
"C:¥ARQMDEMO¥Sample¥SampleResultMerge.pgs"	3

1. Specify a command.
2. Specify the folder to output quality info.
3. Specify the project structure definition file name.

Note: Line feeder is found in the following description, while do not change the line order for actual command execution.

## 2.5. Notes When Using

[Notes for source analysis mode/analysis result consolidation mode]

- Language (Japanese or English) can be selected during installation. If a language that is different from the last time is selected to execute batch execution command (arqm command), it cannot run normally. Please be sure to use the same language to execute batch execution command (arqm command) every time.
- Specify the operation mode (source analysis mode/analysis result consolidation mode) in project structure definition file, but if an operation mode that is different from the last time is selected to execute the batch execution command (arqm command), it cannot run normally. Please be sure to use the same operation mode to execute the batch execution command (arqm command) every time.

[Notes for analysis result consolidation mode]

- Please execute the analysis command for getting analysis result (pgr5, pgrjava etc.) and the batch execution command for consolidating analysis result (arqm command) on the same server.

## 2.6. Suppress comment

Some of the Agile+ Relief C/C++ or Agile+ Relief J output indication messages do not require modification of source files. If a suppress comment has been added, the indication message will be treated as a checked indication message and can be distinguished from the indication messages that need to modify source files. (The following example is in C / C ++.)

### ■ Format

Suppress comments should be described in the following format in the message line of the indication message that does not require modification of source files.

```
/* pgrXXXXX any string */
```

- Only blank spaces and tab characters are allowed between "/\*" and an indication message ID.
- Any string (comment, etc.) is allowed to follow an indication message ID.
- Only one indication message ID is allowed per /\* \*/ comment.
- When multiple suppress comments are required in a line, write them either consecutively or separated by blank space or tab character.
- Suppress comment written in // comment are invalid.

```
    :  
    return x++; // output /* pgr0017 */    <- invalid  
    :  
    return x++; /* pgr0017 */ // output    <- valid  
    :
```

### ■ Sample

pgr0010 : return may have been misspelled as @1.  
pgr0017 : the increment/decrement operation performed in the return value @1 is meaningless.

The following is an example for two suppress comments of indication message.

```
:  
retrun x++; /* pgr0010 */ /* pgr0017 */  
:
```

### 3. Contents of Quality Info Result

#### 3.1. Types of Output HTML Files

HTML file for quality info result is consisted of following types:

Types	Person for Reference	Description
Product Quality Info	Scrum master Product owner Product manager (QMO)	Report the overall quality info across scrum teams.
Product Quality Transition	Scrum master Product owner Product manager (QMO)	
Program Quality Info	Development team Scrum master Product owner	Report the quality info of individual programs within the scrum team (across development teams).
Program Quality Transition	Development team Scrum master Product owner	
Source Information with Indication Message	Development team	Display the embedded indication message in source file.
Source Information with MISRA Violation (*)	Development team	Display the embedded MISRA violation in source file.
Source Information with IPA/SEC-C/C++ Violation (*)	Development team	Display the embedded IPA/SEC-C/C++ violation in source file.
Metrics List	Development team	Display the metrics information for the method types within objective file to analyze.

\*It is able to output in case of C/C++.

When the project structure definition file is applied, the input of arqm command, such as the product quality info and the product quality transition will be output.

### 3.2. Example of HTML File Output

- Product Quality Info

The overall quality info across scrum teams will be collected and then output.

**SampleJava Quality Info**

[SampleJava]

[Details of Quality Info](#) [Indication feature distribution](#) [Download Quality Info](#)

[Project Group Info]

Project Group Name	SampleJava
Date of Statistic	2020/04/29 15:37:32
Status	

[Details of Quality Info]

Quality transition	Project Group Name/Project Name	Quality Status	Unresolved Indications	Resolved Indications	Suppressed Indications	Dangerous Fields	Fields for Attention	All Step Lines	Executed Step Lines	Files
[PPP]	Project1		0	8	16	0	0	267	199	9
[PPP]	Project2		9	6	5	0	0	443	335	9
[PPP]	total( SampleJava )		9	14	21	0	0	710	534	18

[Indication feature distribution]

<Indication feature distribution - Number of indications per 1KStep> \*Red cell: Out of dangerous range; Yellow cell: Out of cautious range

Quality characteristic

Project Group Name/Project Name	Executed Step Size	Number of all message	Reliability	Maintainability	Portability	Efficiency	Functionality	Non-correspondence
Project1	199	0	0.00	0.00	0.00	0.00	0.00	0.00
Project2	335	9	14.93	5.97	0.00	0.00	5.97	0.00
total	534	9	9.36	3.75	0.00	0.00	3.75	0.00

Grammar characteristic

■ Product Quality Transition

The overall quality transition across scrum teams will be collected and then output.

SampleJava Change Info of Quality

[SampleJava]

[Change Info of Quality]

Name	Item	2020/04/23	2020/04/24	2020/04/25	2020/04/26	2020/04/27	2020/04/28	2020/04/29
Project1	Quality Status	-						
	Unresolved Indications	-	24	24	5	5	24	0
	Resolved Indications	-	0	0	3	3	3	8
	Suppressed Indications	-	0	0	16	16	16	16
	Dangerous Fields	-	0	0	0	0	0	0
	Fields for Attention	-	0	0	0	0	0	0
	All Step Lines	-	253	253	254	254	254	267
	Executed Step Lines	-	195	195	196	196	196	199
Files	-	9	9	9	9	9	9	

Name	Item	2020/04/23	2020/04/24	2020/04/25	2020/04/26	2020/04/27	2020/04/28	2020/04/29
Project2	Quality Status	-						
	Unresolved Indications	-	20	20	20	20	20	9
	Resolved Indications	-	0	0	0	0	0	6
	Suppressed Indications	-	0	0	0	0	0	5
	Dangerous Fields	-	0	0	0	0	0	0
	Fields for Attention	-	0	0	0	0	0	0
	All Step Lines	-	439	439	439	439	439	443
	Executed Step Lines	-	335	335	335	335	335	335
Files	-	9	9	9	9	9	9	

Name	Item	2020/04/23	2020/04/24	2020/04/25	2020/04/26	2020/04/27	2020/04/28	2020/04/29
[Unlabeled Project]	Quality Status	-						
	Unresolved Indications	-	44	44	25	25	25	9
	Resolved Indications	-	0	0	3	3	3	14
	Suppressed Indications	-	0	0	16	16	16	21

■ Program Quality Info

The quality info of objective file for analysis within scrum team (across development teams) will be collected and then output.

The screenshot displays a web application interface for 'Project1 Quality Info'. On the left, a 'Project Structure' sidebar shows a tree view with '[SampleJava]' containing '[Project1]' and '[Project2]', and an 'Analysis Log' link. The main content area is titled 'Project1 Quality Info' and includes a sub-section '[SampleJava] - [Project1]' with links for 'Details of Quality Info', 'Indication feature distribution', and 'Download Quality Info'. Below this is a '[Project Info]' section with a table showing 'Project Name' as 'Project1', 'Date of Statistic' as '2020/04/29 15:37:32', and a 'Status' of a sun icon. The main section is '[Details of Quality Info]' containing a table titled '[Object to be checked]' with the following data:

Quality transition	File Name	Unresolved Indications	Resolved Indications	Suppressed Indications	Dangerous Fields	Fields for Attention	All Step Lines	Executed Step Lines
▶▶▶	Project1\src\sample\gasstand\AutoMobil.java	0	0	0	0	0	21	14
▶▶▶	Project1\src\sample\gasstand\DiscountGasStand.java	0	0	0	0	0	26	18
▶▶▶	Project1\src\sample\gasstand\Fuel.java	0	0	0	0	0	8	4
▶▶▶	Project1\src\sample\gasstand\GasStand.java	0	4	0	0	0	42	32
▶▶▶	Project1\src\sample\gasstand\GasStandSimulation.java	0	4	16	0	0	120	101
▶▶▶	Project1\src\sample\gasstand\Portable.java	0	0	0	0	0	8	4
▶▶▶	Project1\src\sample\gasstand\Refuelable.java	0	0	0	0	0	10	5
▶▶▶	Project1\src\sample\gasstand\Shop.java	0	0	0	0	0	10	6



■ Program Quality Transition

The quality transition of objective file for analysis within scrum team (across development teams) will be collected and then output.

Project2 Change Info of Quality

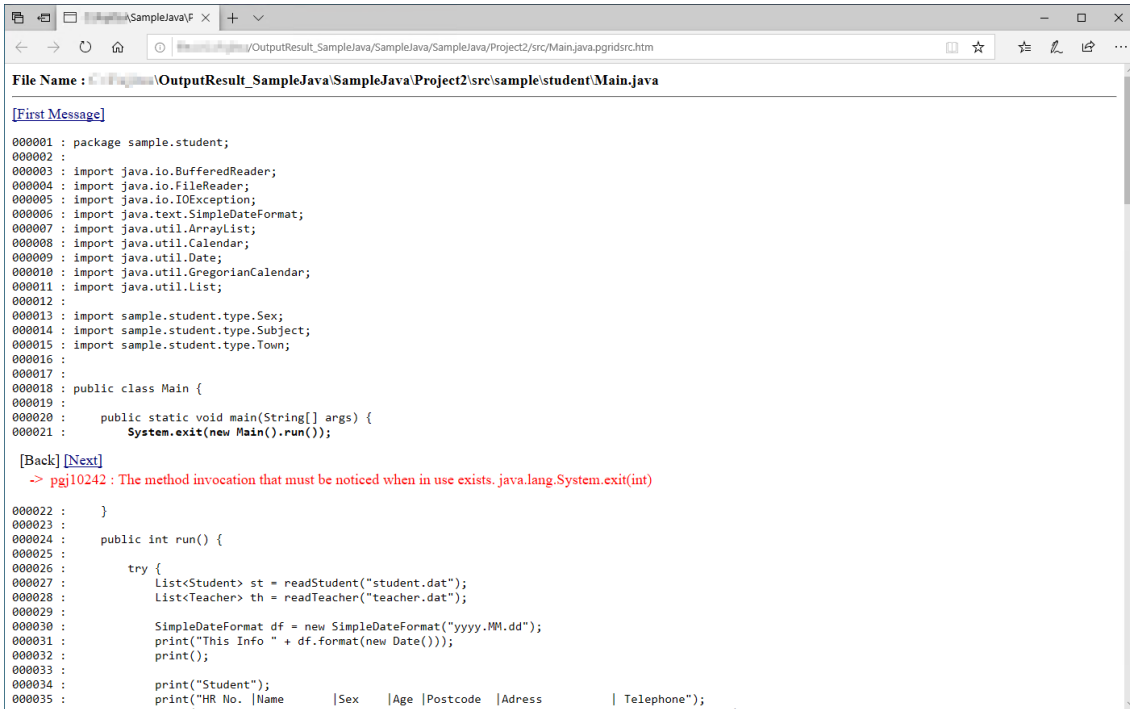
[SampleJava] - [Project2]

[Change Info of Quality]

Name	Item	2020/04/23	2020/04/24	2020/04/25	2020/04/26	2020/04/27	2020/04/28	2020/04/29
Project2\src\sample\student\Main.java	Unresolved Indications	-	17	17	17	17	17	9
	Resolved Indications	-	0	0	0	0	0	6
	Suppressed Indications	-	0	0	0	0	0	2
	Dangerous Fields	-	0	0	0	0	0	0
	Fields for Attention	-	0	0	0	0	0	0
	All Step Lines	-	142	142	142	142	142	146
	Executed Step Lines	-	113	113	113	113	113	113
Project2\src\sample\student\Person.java	Unresolved Indications	-	3	3	3	3	3	0
	Resolved Indications	-	0	0	0	0	0	0
	Suppressed Indications	-	0	0	0	0	0	3
	Dangerous Fields	-	0	0	0	0	0	0
	Fields for Attention	-	0	0	0	0	0	0
	All Step Lines	-	99	99	99	99	99	99
	Executed Step Lines	-	80	80	80	80	80	80
Project2\src\sample\student\Student.java	Unresolved Indications	-	0	0	0	0	0	0
	Resolved Indications	-	0	0	0	0	0	0
	Suppressed Indications	-	0	0	0	0	0	0
	Dangerous Fields	-	0	0	0	0	0	0
	Fields for Attention	-	0	0	0	0	0	0
	All Step Lines	-	29	29	29	29	29	29
	Executed Step Lines	-	20	20	20	20	20	20
Project2\src\sample\student\Student.java	Unresolved Indications	-	0	0	0	0	0	0
	Resolved Indications	-	0	0	0	0	0	0

## ■ Source Information with Indication Message

The source file with the indication message embedded will be output.

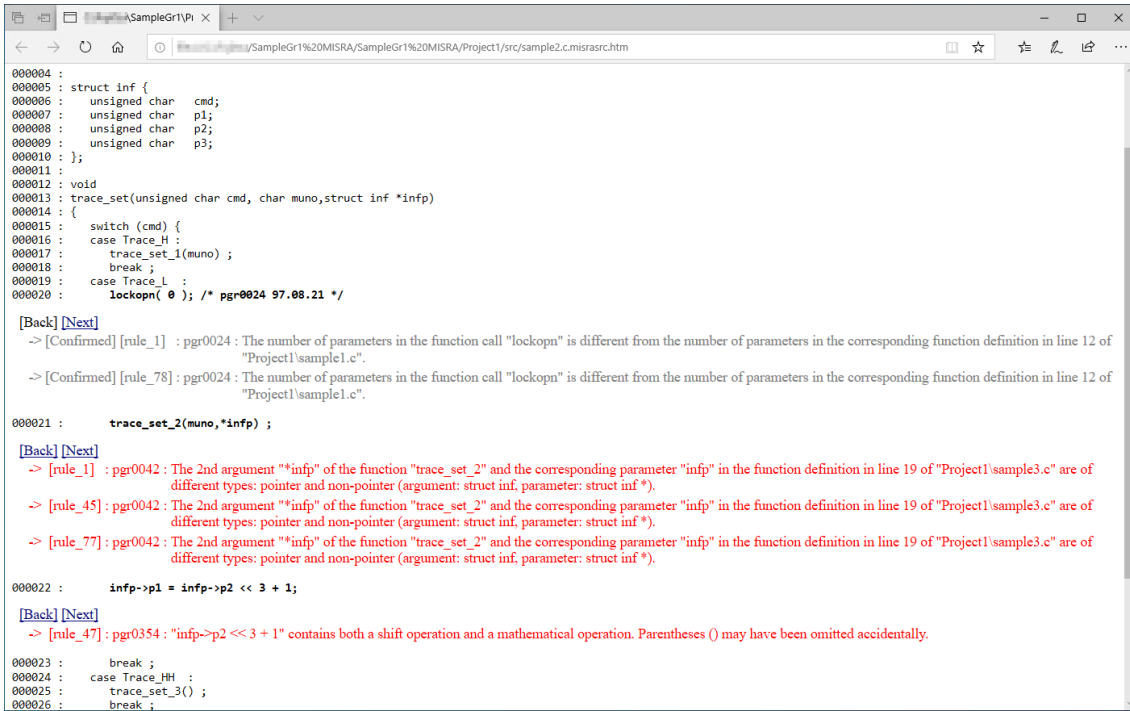


The screenshot shows a web browser window displaying the source code of a Java file. The browser's address bar shows the file path: `file:///C:/Users/pjg/Desktop/OutputResult_SampleJava/SampleJava/Project2/src/sample/student/Main.java`. The file name is `Main.java`. The code is displayed with line numbers from 000001 to 000035. A message is embedded in the code at line 000021, which is highlighted in red. The message reads: `-> pgj10242: The method invocation that must be noticed when in use exists. java.lang.System.exit(int)`. The code includes package declarations, imports for various Java classes, and a `Main` class with a `main` method and a `run` method. The `run` method contains logic to read data from files, format dates, and print student information.

```
000001 : package sample.student;
000002 :
000003 : import java.io.BufferedReader;
000004 : import java.io.FileReader;
000005 : import java.io.IOException;
000006 : import java.text.SimpleDateFormat;
000007 : import java.util.ArrayList;
000008 : import java.util.Calendar;
000009 : import java.util.Date;
000010 : import java.util.GregorianCalendar;
000011 : import java.util.List;
000012 :
000013 : import sample.student.type.Sex;
000014 : import sample.student.type.Subject;
000015 : import sample.student.type.Town;
000016 :
000017 :
000018 : public class Main {
000019 :
000020 :     public static void main(String[] args) {
000021 :         System.exit(new Main().run());
000022 :     }
000023 :
000024 :     public int run() {
000025 :         try {
000026 :             List<Student> st = readStudent("student.dat");
000027 :             List<Teacher> th = readTeacher("teacher.dat");
000028 :
000029 :             SimpleDateFormat df = new SimpleDateFormat("yyyy.MM.dd");
000030 :             print("This Info " + df.format(new Date()));
000031 :             print();
000032 :             print();
000033 :             print("Student");
000034 :             print("HR No. |Name      |Sex   |Age |Postcode |Adress      | Telephone");
000035 :
```

## ■ Source Information with MISRA Violation

The source file with MISRA violation embedded will be output.



```
000004 :
000005 : struct inf {
000006 :     unsigned char  cmd;
000007 :     unsigned char  p1;
000008 :     unsigned char  p2;
000009 :     unsigned char  p3;
000010 : };
000011 :
000012 : void
000013 : trace_set(unsigned char cmd, char muno,struct inf *infp)
000014 : {
000015 :     switch (cmd) {
000016 :     case Trace_H :
000017 :         trace_set_1(muno) ;
000018 :         break ;
000019 :     case Trace_L :
000020 :         lockopn( 0 ); /* pgr0024 97.08.21 */
000021 :
000021 :         trace_set_2(muno,*infp) ;
000022 :
000022 :         infp->p1 = infp->p2 << 3 + 1;
000023 :         break ;
000024 :     case Trace_HH :
000025 :         trace_set_3() ;
000026 :         break ;
```

[Back] [Next]

-> [Confirmed] [rule\_1] : pgr0024 : The number of parameters in the function call "lockopn" is different from the number of parameters in the corresponding function definition in line 12 of "Project1\sample1.c".

-> [Confirmed] [rule\_78] : pgr0024 : The number of parameters in the function call "lockopn" is different from the number of parameters in the corresponding function definition in line 12 of "Project1\sample1.c".

[Back] [Next]

-> [rule\_1] : pgr0042 : The 2nd argument "\*infp" of the function "trace\_set\_2" and the corresponding parameter "infp" in the function definition in line 19 of "Project1\sample3.c" are of different types: pointer and non-pointer (argument: struct inf, parameter: struct inf \*).

-> [rule\_45] : pgr0042 : The 2nd argument "\*infp" of the function "trace\_set\_2" and the corresponding parameter "infp" in the function definition in line 19 of "Project1\sample3.c" are of different types: pointer and non-pointer (argument: struct inf, parameter: struct inf \*).

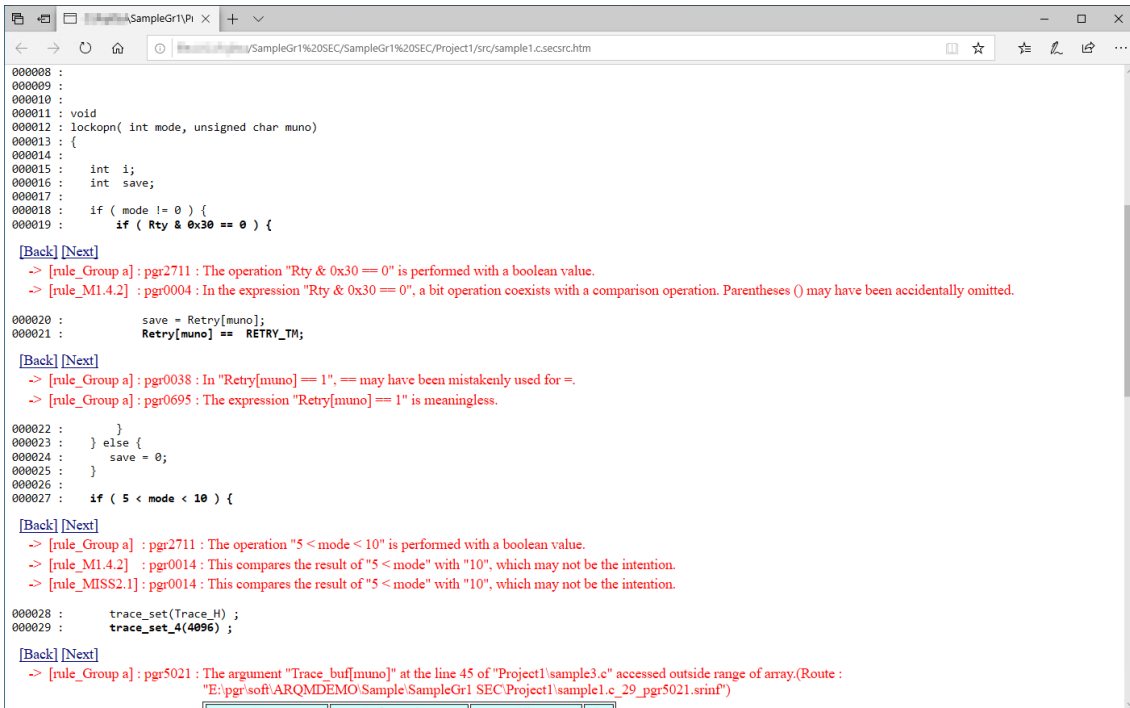
-> [rule\_77] : pgr0042 : The 2nd argument "\*infp" of the function "trace\_set\_2" and the corresponding parameter "infp" in the function definition in line 19 of "Project1\sample3.c" are of different types: pointer and non-pointer (argument: struct inf, parameter: struct inf \*).

[Back] [Next]

-> [rule\_47] : pgr0354 : "infp->p2 << 3 + 1" contains both a shift operation and a mathematical operation. Parentheses () may have been omitted accidentally.

## ■ Source Information with IPA/SEC-C/C++ Violation

The source file with IPA/SEC-C/C++ violation embedded will be output.



```
000008 :
000009 :
000010 :
000011 : void
000012 : lockopn( int mode, unsigned char muno)
000013 : {
000014 :
000015 :     int i;
000016 :     int save;
000017 :
000018 :     if ( mode != 0 ) {
000019 :         if ( Rty & 0x30 == 0 ) {
000020 :             save = Retry[muno];
000021 :             Retry[muno] == RETRY_TM;
000022 :         }
000023 :     } else {
000024 :         save = 0;
000025 :     }
000026 :
000027 :     if ( 5 < mode < 10 ) {
000028 :         trace_set(Trace_H) ;
000029 :         trace_set_4(4096) ;
000030 :     }
000031 : }
```

[Back] [Next]

- > [rule\_Group a] : pgr2711 : The operation "Rty & 0x30 == 0" is performed with a boolean value.
- > [rule\_M1.4.2] : pgr0004 : In the expression "Rty & 0x30 == 0", a bit operation coexists with a comparison operation. Parentheses () may have been accidentally omitted.

[Back] [Next]

- > [rule\_Group a] : pgr0038 : In "Retry[muno] == 1", == may have been mistakenly used for =.
- > [rule\_Group a] : pgr0695 : The expression "Retry[muno] == 1" is meaningless.

[Back] [Next]

- > [rule\_Group a] : pgr2711 : The operation "5 < mode < 10" is performed with a boolean value.
- > [rule\_M1.4.2] : pgr0014 : This compares the result of "5 < mode" with "10", which may not be the intention.
- > [rule\_MISS2.1] : pgr0014 : This compares the result of "5 < mode" with "10", which may not be the intention.

[Back] [Next]

- > [rule\_Group a] : pgr5021 : The argument "Trace\_buff[muno]" at the line 45 of "Project1/sample3.c" accessed outside range of array.(Route : "E:\pgr\soft\ARQMDEMO\Sample\SampleGr1 SEC\Project1\sample1.c\_29\_pgr5021.srinf")

## ■ Metrics List

The metrics information will be output for the functions within the objective file for analysis.

The screenshot shows a web browser window with the following content:

OutputResult\_SampleJava\SampleJava\Project2\src\sample\student\Main.java Metrics Info

[Metrics]

File name	All Step Size	Executed Step Size	Number of functions
Main.java	146	113	7

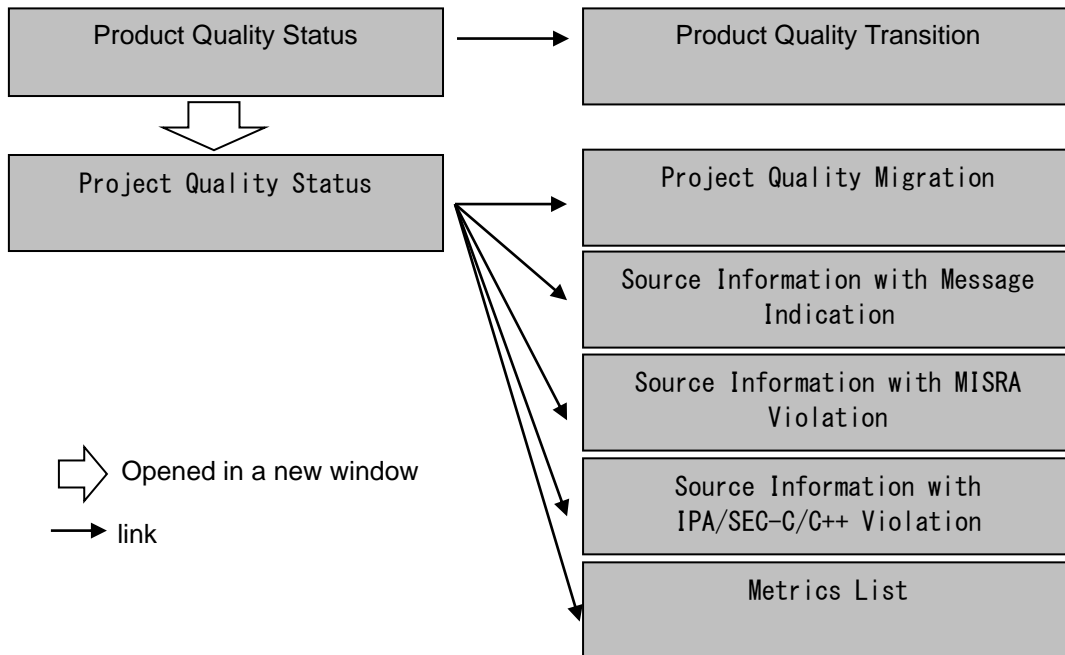
  

Function name	Line number	All Step Size	Executed Step Size	Nesting Levels	Complexity 1	Complexity 2	goto	return	break	continue	if1	while1	dowhile1	for1	switch1	if2	while2	dowhile2
sample.student.Main#main	20	3	3	0	1	1	-	0	0	0	0	0	0	0	0	0	0	0
sample.student.Main#run	24	36	29	2	3	3	-	3	0	0	0	0	0	2	0	0	0	0
sample.student.Main#readStudent	61	34	27	3	3	3	-	1	0	1	1	1	0	0	0	1	1	0
sample.student.Main#readTeacher	96	35	28	3	3	3	-	1	0	1	1	1	0	0	0	1	1	0
sample.student.Main#toCal	132	5	5	0	1	1	-	1	0	0	0	0	0	0	0	0	0	0
sample.student.Main#print	138	3	3	0	1	1	-	0	0	0	0	0	0	0	0	0	0	0
sample.student.Main#print	142	3	3	0	1	1	-	0	0	0	0	0	0	0	0	0	0	0

Red : Exceeds the value defined in Dangerous Field; Green : Exceeds the value defined in Field for Attention

### 3.3. Relation Among Output HTML Files

Various HTML files are interlinked as follows:



### 3.4. Types of Output CSV Files

As the data for further processing, the following CSV files are output:

Types	Description
Product Quality Info List	Output the quality info. The contents are different for types.
Program Quality Info List	About the items to output, please refer to " <a href="#">3.6 Description of Definition File for Output Content</a> ".
Product Quality Transition List	Output the quality transition. The contents are different for types.
Program Quality Transition List	About the items to output, please refer to " <a href="#">3.6 Description of Definition File for Output Content</a> ".
List of product indication feature distribution	Output the indication feature distribution. The contents are different for types.
List of program indication feature distribution	About the items to output, please refer to " <a href="#">3.6 Description of Definition File for Output Content</a> ".
Indication Message List	The message will be output for all objective files for analysis.
MISRA Violation List	MISRA violation will be output for all objective files for analysis within the project.
IPA/SEC-C/C++ Violation List	IPA/SEC-C/C++ violation will be output for all objective files for analysis within the project.
Metrics List	Metrics information will be output for all objective files for analysis.

### 3.5. Template File Update

HTML file will be output for the contents of content definition file and HTML template file. If want to change the output format, the compilation of HTML file is required; if want to change the items of list, the edition of definition file for output contents is required.

Furthermore, the template saved in the [HTML\_TEMPLATES] folder under the install folder of Agile+ Relief will be referenced.

The relation between the various HTML files and HTML template files are described in the following table:

HTML Template File name	Types of HTML File for Reference
index.htm	frame of [Product Quality Info] frame of [Program Quality Info]
project_tree.htm	Project Structure of [Product Quality Info] Project Structure of [Program Quality Info]
prjgrp_detail.htm	Project Quality Info of [Product Quality Info]
project_detail.htm	Project Quality Info of [Program Quality Info]
prjgrp_transition.htm	[Product Quality Transition]
prject_transition.htm	[Program Quality Transition]
source.htm	[Source Information with Indication Message] [Source Information with MISRA Violation] [Source Information with IPA/SEC-C/C++ Violation]
metrics.htm	[Metrics List]

The embedded keyword and its contents described in HTML template file are displayed as follows:

Classification	Description	Embedded Keyword
Frame	Display the tree page of project structure.	%%REF_PRJTREE_PAGE%%
	Display the quality info of product or scrum team.	%%REF_PRJGRP_PAGE%%
Analysis Log	Display the icon of analysis log file.	%%ICON_LOG%%
	Display the link pointing to the analysis log file.	%%LINK_LOG_ANALYZE%%
Product Information	Display the product name.	%%PRJGRP_NAME%%
	Display the statistic date and time for product.	%%PRJGRP_REPORT_TIME%%
	Display the quality info of product.	%%PRJGRP_STATUSES%%
Scrum Information team	Display the name of scrum team.	%%PROJECT_NAME%%
	Display the statistic date and time of scrum team.	%%PROJECT_REPORT_TIME%%
	Display the quality info of scrum team.	%%PROJECT_STATUSES%%

Quality Info	<p>The quality info of scrum team affiliated to product or development team affiliated to scrum team will be displayed in the list.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">3.6 Description of Definition File for Output Content</a>".</p>	%%PRJGRP_DETAIL_LIST%%
	<p>The quality info of source file affiliated to the development team will be displayed in the list.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">3.6 Description of Definition File for Output Content</a>".</p>	%%PROJECT_DETAIL_LIST%%
Quality Transition	<p>The change info of quality of scrum team affiliated to product or development team affiliated to scrum team will be displayed in the list.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">3.6 Description of Definition File for Output Content</a>".</p>	%%PRJGRP_TRANSITION_LIST%%
	<p>The change info of quality of source file affiliated to the development team will be displayed in the list.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">3.6 Description of Definition File for Output Content</a>".</p>	%%PROJECT_TRANSITION_LIST%%
Indication feature distribution	<p>Calculate the indication messages of scrum team affiliated to product or development team affiliated to scrum team according to its feature and display in forms of a list.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">4.2.4 Indication feature definition file</a>".</p>	%%PRJGRP_MESSAGE_COUNT_LIST%%
	<p>Calculate the indication messages of scrum team affiliated to product or development team affiliated to scrum team according to its feature and display the value per Kstep in forms of a list.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">4.2.4 Indication feature definition file</a>".</p>	%%PRJGRP_MESSAGE_K_COUNT_LIST%%
	<p>Calculate the indication messages of the source file that belongs to scrum team (development team) according to its feature and display in forms of a list.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">4.2.4 Indication feature definition file</a>".</p>	%%PROJECT_MESSAGE_COUNT_LIST%



	<p>Calculate the indication messages of the source file that belongs to scrum team (development team) according to its feature and display the value per Kstep in forms of a list.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">4.2.4 Indication feature definition file</a>".</p>	%%PROJECT_MESS AGE_K_COUNT_LIS T%%
Source Information	The indication message will be displayed in the form of embedded line in source.	%%SOURCE_WITH_ WARNING%%
Metrics Information	<p>In the list, the metrics information will be displayed separately for the method defined in source file.</p> <p>About the items to be displayed in the list, please refer to "<a href="#">4.2.2 Dangerous /Attention Field Definition file</a>".</p>	%%METRICS_LIST% %
download	Display the icon of download file.	%%ICON_DOWNLOA D%%
	Display the download link pointing to the CSV output result of quality info of product or scrum team.	%%LINK_CSV_PRJG RP_DETAIL_LIST%%
	Display the download link pointing to the CSV output result of quality transition of product or scrum team.	%%LINK_CSV_PRJG RP_TRANSITION_LI ST%%
	Display the download link pointing to the CSV output result of quality info of product or scrum team.	%%LINK_CSV_PROJ ECT_DETAIL_LIST% %
	Display the download link pointing to the CSV output result of quality transition of product or scrum team.	%%LINK_CSV_PROJ ECT_TRANSITION_LI ST%%
	Display the download link of CSV output result of product indication feature distribution list (Calculate according to the number of indication messages).	%%LINK_CSV_PRJG RP_MESSAGE_COU NT_LIST%%
	Display the download link of CSV output result of product indication feature distribution list (Calculate according to the number of indication messages per Kstep).	%%LINK_CSV_PRJG RP_MESSAGE_K_C OUNT_LIST%%
	Display the download link of CSV output result of program indication feature distribution list (Calculate according to the number of indication messages).	%%LINK_CSV_PROJ ECT_MESSAGE_CO UNT_LIST%%
	Display the download link of CSV output result of program indication feature distribution list (Calculate according to the number of indication messages per Kstep).	%%LINK_CSV_PROJ ECT_MESSAGE_K_C OUNT_LIST%%
	Display the download link pointing to the CSV output result of indication message list.	%%LINK_CSV_PGRI D_LIST%%
	Display the download link pointing to the CSV output result of MISRA violation.	%%LINK_CSV_MISR A_LIST%%
	Display the download link pointing to the CSV output result of IPA/SEC-C/C++ violation.	%%LINK_CSV_SEC_ LIST%%
	Display the download link of metrics list.	%%LINK_CSV_METR ICS_LIST%%

	Display the download link pointing to the file linking with the indication message viewer.	%%LINK_VIEWER_FILE%%
Others	Display the relation between product and scrum team in the form of tree structure.	%%PROJECT_TREE%%
	Display the sole name of product or scrum team.	%%PROJECT_POSITION%%
	Display the source file name.	%%FILE_NAME%%

The embedded keywords which can be recorded in HTML template file are described as follows:

HTML Template File Name		index.htm	project_tree.htm	prjgrp_detail.htm	project_detail.htm	prjgrp_transition.htm	project_transition.htm	source.htm	metrics.htm
Embedded Keyword									
Frame	%%REF_PRJTREE_PAGE%%	OK	-	-	-	-	-	-	-
	%%REF_PRJGRP_PAGE%%	OK	-	-	-	-	-	-	-
Analysis Log	%%ICON_LOG%%	-	OK	-	-	-	-	-	-
	%%LINK_LOG_ANALYZE%%	-	OK	-	-	-	-	-	-
Project Group Information	%%PRJGRP_NAME%%	-	-	OK	-	OK	-	-	-
	%%PRJGRP_REPORT_TIME%%	-	-	OK	-	OK	-	-	-
	%%PRJGRP_STATUS%%	-	-	OK	-	OK	-	-	-
Project Information	%%PROJECT_NAME%%	-	-	-	OK	-	OK	-	-
	%%PROJECT_REPORT_TIME%%	-	-	-	OK	-	OK	-	-
	%%PROJECT_STATUS%%	-	-	-	OK	-	OK	-	-
Quality Info	%%PRJGRP_DETAIL_LIST%%	-	-	OK	-	-	-	-	-
	%%PROJECT_DETAIL_LIST%%	-	-	-	OK	-	-	-	-
Quality Transition	%%PRJGRP_TRANSITION_LIST%%	-	-	-	-	OK	-	-	-
	%%PROJECT_TRANSITION_LIST%%	-	-	-	-	-	OK	-	-
Indication feature distribution (*)	%%PRJGRP_MESSAGE_COUNT_LIST%%	-	-	OK	-	-	-	-	-
	%%PRJGRP_MESSAGE_K_COUNT_LIST%%	-	-	OK	-	-	-	-	-
	%%PROJECT_MESSAGE_COUNT_LIST%%	-	-	-	OK	-	-	-	-
	%%PROJECT_MESSAGE_K_COUNT_LIST%%	-	-	-	OK	-	-	-	-
Source Information	%%SOURCE_WITH_WARNING%%	-	-	-	-	-	-	OK	-
Metrics Information	%%METRICS_LIST%%	-	-	-	-	-	-	-	OK
Download	%%ICON_DOWNLOAD%%	-	-	OK	OK	OK	OK	-	-
	%%LINK_CSV_PRJGRP_DETAIL_LIST%%	-	-	OK	-	OK	-	-	-
	%%LINK_CSV_PRJGRP_TRANSITION_LIST%%	-	-	OK	-	OK	-	-	-
	%%LINK_CSV_PROJECT_DETAIL_LIST%%	-	-	-	OK	-	OK	-	-
	%%LINK_CSV_PROJECT_TRANSITION_LIST%%	-	-	-	OK	-	OK	-	-
	%%LINK_CSV_PRJGRP_MESSAGE_COUNT_LIST%%	-	-	OK	-	-	-	-	-
	%%LINK_CSV_PRJGRP_MESSAGE_K_COUNT_LIST%%	-	-	OK	-	-	-	-	-
	%%LINK_CSV_PROJECT_MESSAGE_COUNT_LIST%%	-	-	-	OK	-	-	-	-
	%%LINK_CSV_PROJECT_MESSAGE_K_COUNT_LIST%%	-	-	-	OK	-	-	-	-
	%%LINK_CSV_PGRID_LIST%%	-	-	-	OK	-	OK	-	-
	%%LINK_CSV_MISRA_LIST%%	-	-	-	OK	-	OK	-	-

	%%LINK_CSV_SEC_LIST%%	-	-	-	OK	-	OK	-	-
	%%LINK_CSV_METRICS_LIST%%	-	-	-	OK	-	OK	-	-
	%%LINK_VIEWER_FILE%%	-	-	-	OK	-	-	-	-
Others	%%PROJECT_TREE%%	-	OK	-	-	-	-	-	-
	%%PROJECT_POSITION%%	-	-	OK	OK	OK	OK	-	-
	%%FILE_NAME%%	-	-	-	-	-	-	OK	OK

\*The HTML template file contains the embedded keyword for displaying indication feature distribution. When indication feature distribution is not displayed, please delete the embedded keyword for displaying indication feature distribution from the HTML template file.

### 3.6. Description of Definition File for Output Content

Within the output content definition file, the definition is made for the items to be displayed in the various lists, and the icon file names to be displayed in diagnose results.

- File Name

Created in the file name of [arqm.ini] in the [HTML\_TEMPLATE] folder under the install folder of Agile+ Relief.

- Format

The format is described as follows: When ";" (Semicolon) is described at the beginning of a line, this line will be invalid.

```
[section]
key = Value
key = Value
:
:
[section]
key = Value
:
```

- Define the Items to be displayed in List

Define the items to be displayed in the list of product quality info, and program quality info. Furthermore, the settings are also applied to the items to be displayed in the quality transition (including CSV file).

Sections corresponding to various lists are displayed as follows:

Types of Lists	Section Name
Product Quality Info	DETAIL_LIST_MANAGER
Scrum Team Quality Info	DETAIL_LIST_LEADER
Development Team Quality Info	DETAIL_LIST

The items displayed in the list are corresponding to the item names and titles defined by the format under the section directory. The characters for title shall be within 256 bytes.

```
[DETAIL_LIST_MANAGER]
Identifier Name for Item = Strings of Title are displayed.
:
[DETAIL_LIST_LEADER]
Identifier Name for Item = Strings of Title are displayed.
:
[DETAIL_LIST]
Identifier Name for Item = Strings of Title are displayed.
:
```

The items, relevant identifier names, and whether they can be recorded into the section are described as follows:

Identifier Name	Description	Recordable		
		DETAIL_LIST_MANAGER	DETAIL_LIST_LEADER	DETAIL_LIST
PRJGRP_NAME	Project Group Name	Yes	Yes	No
FILE_NAME	File name	No	No	Yes
QUALITY_STATUS	Quality status Icon	Yes	Yes	Yes*1
NEW_COUNT	New indication count	Yes	Yes	Yes
NEW_K_COUNT	New indication count in every 1 KS step Calculated according to "New Indication Count/Times for Execution"	Yes	Yes	Yes
NEW_GROUP_A_COUNT	New Group A indication count	Yes	Yes	Yes
EXIST_COUNT	Existing indication count	Yes	Yes	Yes
EXIST_K_COUNT	Existing indication count in every 1 KS step Calculated according to "Existing Indication Count/Times for Execution"	Yes	Yes	Yes
EXIST_GROUP_A_COUNT	Existing Group A indication count	Yes	Yes	Yes
INDICATION_COUNT	Unprocessed indication count	Yes	Yes	Yes
INDICATION_K_COUNT	Indication count in every 1 KS step Calculated according to "Unprocessed Indication Count/Times for Execution"	Yes	Yes	Yes
INDICATION_GROUP_A_COUNT	Unprocessed Group A indication count	Yes	Yes	Yes
INDICATION_FINISHED_COUNT	Processed indication count	Yes	Yes	Yes
MODIFIED_COUNT	Modified indication count	Yes	Yes	Yes
SUPPRESS_COUNT	Suppressed indication count	Yes	Yes	Yes
BEFORE_MANAGED_COUNT	Indication count before managed	Yes	Yes	Yes
NEW_MISRA_COUNT	New MISRA violation count	Yes	Yes	Yes
NEW_MISRA_K_COUNT	New MISRA violation count in every 1 KS step Calculated according to "New MISRA Violation Count/Times for Execution"	Yes	Yes	Yes
EXIST_MISRA_COUNT	Existing MISRA violation count	Yes	Yes	Yes

EXIST_MISRA_K_COUNT	Existing MISRA violation count in every 1 KS step Calculated according to "Existing MISRA Violation Count/Times for Execution"	Yes	Yes	Yes
MISRA_COUNT	Unprocessed MISRA violation count	Yes	Yes	Yes
MISRA_K_COUNT	Unprocessed MISRA violation count in every 1 KS step Calculated according to "Unprocessed MISRA Violation Count/Times for Execution"	Yes	Yes	Yes
MISRA_FINISHED_COUNT	Processed MISRA violation count	Yes	Yes	Yes
MODIFIED_MISRA_COUNT	Modified MISRA violation count	Yes	Yes	Yes
SUPPRESS_MISRA_COUNT	Suppressed MISRA violation count	Yes	Yes	Yes
BEFORE_MANAGED_MISRA_COUNT	MISRA violation count before managed	Yes	Yes	Yes
NEW_SEC_COUNT	New IPA/SEC-C/C++ violation count	Yes	Yes	Yes
NEW_SEC_K_COUNT	New IPA/SEC-C/C++ violation count in every 1 KS step Calculated according to "New IPA/SEC-C/C++ Violation Count/Times for Execution"	Yes	Yes	Yes
EXIST_SEC_COUNT	Existing IPA/SEC-C/C++ violation count	Yes	Yes	Yes
EXIST_SEC_K_COUNT	Existing IPA/SEC-C/C++ violation count in every 1 KS step Calculated according to "Existing IPA/SEC-C/C++ Violation Count/Times for Execution"	Yes	Yes	Yes
SEC_COUNT	Unprocessed IPA/SEC-C/C++ violation count	Yes	Yes	Yes
SEC_K_COUNT	Unprocessed IPA/SEC-C/C++ violation count in every 1 KS step Calculated according to "Unprocessed IPA/SEC-C/C++ Violation Count/Times for Execution"	Yes	Yes	Yes
SEC_FINISHED_COUNT	Processed IPA/SEC-C/C++ violation count	Yes	Yes	Yes
MODIFIED_SEC_COUNT	Modified IPA/SEC-C/C++ violation count	Yes	Yes	Yes
SUPPRESS_SEC_COUNT	Suppressed IPA/SEC-C/C++ violation count	Yes	Yes	Yes
BEFORE_MANAGED_SEC_COUNT	IPA/SEC-C/C++ violation count before managed	Yes	Yes	Yes

DANGER_COUNT	Danger Field Excess	Yes	Yes	Yes
MARKED_COUNT	Field for Attention Excess	Yes	Yes	Yes
ALL_LINE_COUNT	total lines	Yes	Yes	Yes
RUN_LINE_COUNT	Run Lines	Yes	Yes	Yes
FILE_COUNT	File Count	Yes	Yes	Yes*1
UPDATE_DATE	Date for File Update	No	No	Yes
CHECK_DATA	Date for File Check	No	No	Yes

\*1 Only Total Column of project quality transition is valid.

Sample is described as follows:

[DETAIL_LIST_LEADER]	
PRJGRP_NAME	= daily business process development
QUALITY_STATUS	= quality stat
INDICATION_COUNT	= remaining indication count
MODIFIED_COUNT	= modified indication count
SUPRESS_COUNT	= suppressed indication count
DANGER_COUNT	= danger field excess
MARKED_COUNT	= field for attention excess
ALL_LINE_COUNT	= total lines
RUN_LINE_COUNT	= run lines
FILE_COUNT	= file count

■ Settings of Displayed Icon

The settings of displayed icons in HTML file are described in [ICON] section.

Specify the icon file name saved in the same folder of [HTML\_TEMPLATE%image] with that of batch execution command (arqm command).

Various key settings are displayed as follows:

name	Description
ICON_GOOD	Specify the icon file name, which represents [Diagnose Result = GOOD] to be displayed in project structure page.
ICON_PASSABLE	Specify the icon file name, which represents [Diagnose Result = PASSBLE] to be displayed in quality info page.
ICON_NOGOOD	Specify the icon file name, which represents [Diagnose Result = NOGOOD] to be displayed in quality info page.
ICON_ANALYZEERR	Specify the icon file name, which represents [Diagnose Result = ANALYZEERR] to be displayed in quality info page.
ICON_LOG	Specify the icon file name, which represents [Log File] to be displayed in project structure page.
ICON_DOWNLOAD	Specify the icon file name, which represents [Download] to be displayed in quality info page.
ICON_REFERENCE	Specify the icon file name, which represents [Browse] to be displayed in quality info page.

The sample is displayed as follows:



[ICON]	
ICON_GOOD	= clear.png
ICON_PASSABLE	= cloudy.png
ICON_NOGOOD	= rain.png
ICON_LOG	= log.png
ICON_DOWNLOAD	= download.png
ICON_REFERENCE	= ref.png

■ Other Display Relevant Settings

Other display settings are described in [ENVIRONMENT] section.

Various key settings are displayed as follows:

name	Description
DAYSTARTTIME	Specify the start time to collect data in one day within the quality transition list display. Format: hh:mm:ss Specify with reference to 24 hours.
CONVERTKANA	Specify the characters recorded in source file and designated by project name, when half-width katakana is converted into full-width katakana to be displayed in HTML file. 0:Output in half-width katakana. 1:Output after half-width katakana is converted into full-width katakana.
TABNUM	When TAB character displayed in HTML file is replaced with space character, specify the space within the scope of 1~10. 4 by default when omitted.

The sample is described as follows:

[ENVIRONMENT]	
DAYSTARTTIME	= 23:00:00
CONVERTKANA	= 0

## 4. Description of Various Definition Files

### 4.1. Project Structure Definition

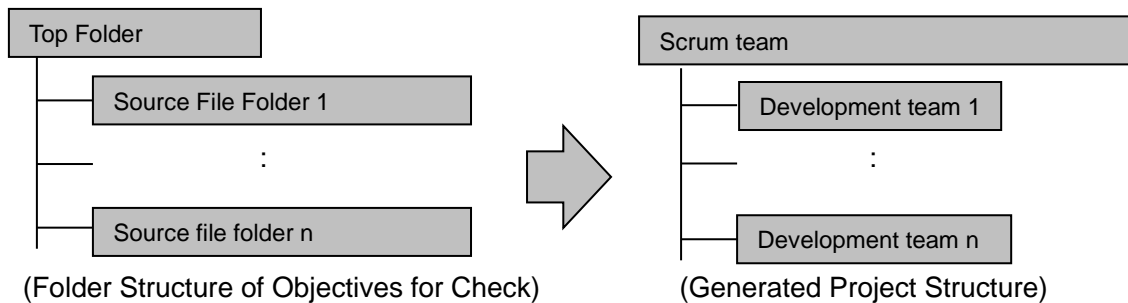
The project structure definition is described in [Project Structure Definition File] and [Project Structure Definition Listing File].

#### ■ Project Structure Definition File

The definition file is the minimum input unit for arqm command.

Define the top folder of objective asset for check, and the folders beneath are regarded as development team, the source files under the development team directory are recognized as the objectives for check.

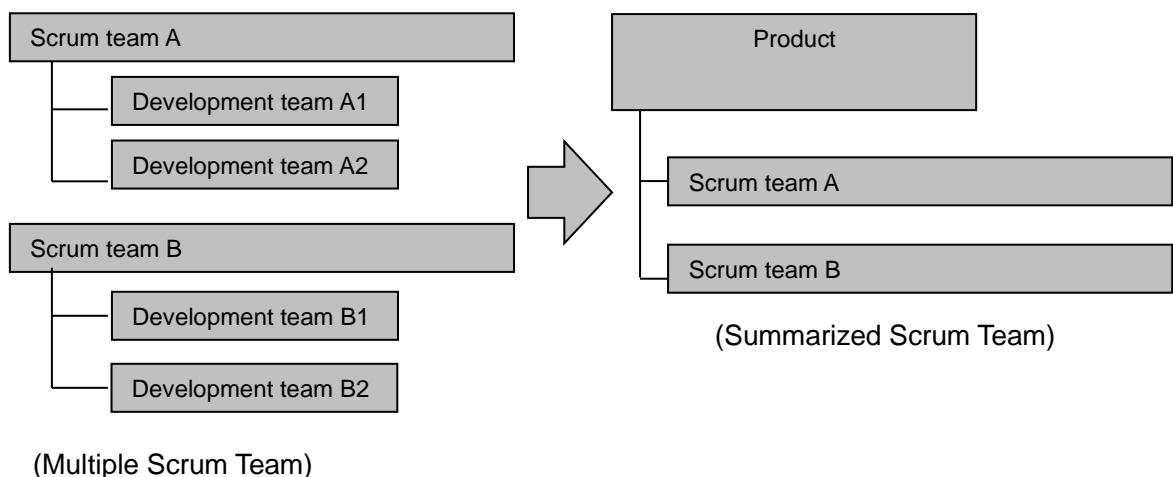
A scrum team is created by defining project structure definition file, and multiple development teams will be generated based on the folder structure affiliated to the directory.



#### ■ Project Structure Definition Listing File

Combine the multiple project groups into one group. In this way, the units for execution or output of arqm command can be collected.

Within the project structure definition listing file, multiple [Project Structure Definition File] or [Project Structure Definition Listing File] are defined.



### 4.1.1. Project Structure Definition File

■ File Name

The format of file name for project structure definition file is described as follows:

any.pgs

■ Creation

The template is saved in the file name of [project.pgs] to the same folder [definition\_template] with that of batch execution command (arqm command). The necessary settings for template file for reference are described as follows:

■ Format

The format is described as follows: When ";" (Semicolon) is described at the beginning of a line, this line will be invalid.

```
[section]
key = Value
key = Value
:
:
[section]
key = Value
:
```

■ Section and Key (common)

The necessary settings for section and key are described as follows:

Also, when extracting objective asset for check from the version management tools, in addition to the following section and KEY, please specify other sections and KEY shown in the table of "Section and Key (for version management tools)".

Section and Key	Yes No			Description
	source analysis		analysis result consolidation	
	C / C + +	J a v a		
[SOURCES]	Yes	Yes	Yes	Set the section of objective asset for check.
MODE			Yes	Specify the operation mode of Agile+ Relief. ANALYZE : source analysis mode. Input the source file of C/C++ and Java, analyze and consolidate the analysis result. RESULTMERGE : analysis result consolidation mode. Input the analysis result of analysis command and consolidate only the analysis result. Default is ANALYZE.

LANG		Yes		Specify the language of asset to be checked. C: C/C++ Java: Java Default is C/C++.
TARGET	Yes	Yes		Specify the top folder in which the objectives for checking and consolidating are saved. [Under source analysis mode] Search the folder directory specified by TARGET, the folder containing C/C++ source files will be recognized as project. When using the version management tools, asset can be recovered to the folder directory designated by TARGET. [Under analysis result consolidation mode] Search the folder directory specified by TARGET, the folder containing the analysis result of analysis command will be recognized as project. About the sample for project creation, please refer to " <a href="#">4.1.3 Example to Designate Target Folder for Project Creation</a> ".
PROJECTLE VEL			No	Specify when the folders recognized as project is positioned to the corresponding layers(1~20) designated by distance TARGET. The upper folders upon the specified layer are different for projects. About the sample for illustration, please refer to " <a href="#">2.4.1.5 Execute Sample of Different Folder Structure</a> ".
CODE				Specify the code of objectives assets for check. [In case of C/C++] SJIS : SJIS code EUC : EUC-JP code GB2312 : GB2312 code UTF8 : UTF-8 code The code when it is omitted is as follows. ● Chinese OS GB2312 code ● OS in other languages SJIS code in Windows, UTF-8 code in Red Hat Enterprise Linux 5/6, and EUC code in others [In case of Java] SJIS : Cp1252 code. UTF8 : UTF-8 code EUC : EUC-JP code The default is Cp1252 code.

CPLUSPLUS EXT		No	No	<p>[In case of C/C++]</p> <p>Specify the extension of C++ source file of objectives for check.</p> <p>When multiple objective extensions are found, multiple keys need to be specified.</p>
JAVAEXT	No		No	<p>[In case of Java]</p> <p>Specify the extension of file of objectives for check.</p> <p>When multiple objective extensions are found, multiple keys need to be specified.</p> <p>About the file of objectives for check, please refer to "1.2.2 File of target for Code check" of [Agile+ Relief J Code check Operation Guide].</p>
MSGFILEEXT	No	No		<p>Specify an extension for the consolidated indication message file.</p> <p>In default, ".message" is set as the extension of message file.</p>
MISRAFILEEXT	No	No		<p>Specify an extension for the consolidated MISRA check result file.</p> <p>In default, ". misra" is set as the extension of MISRA check result file.</p>
SECFILEEXT	No	No		<p>Specify an extension for the consolidated SEC check result file.</p> <p>In default, ". sec" is set as the extension of SEC check result file.</p>
BINDPROJECT			No	<p>Specify when the different layers within the folder by TARGET need to be treated as same project.</p> <p>In the folder will become the object, specify the path except for the folder specified by TARGET.</p> <p>When multiple designations are required, multiple keys need to be specified.</p> <p>About the sample of project creation, please refer to "<a href="#">4.1.3 Example to Designate Target Folder for Project Creation</a>".</p>
EXCEPTDIR		No	No	<p>[In case of C/C++]</p> <p>Specify the folder to save the source files dropped from the objectives for check in the folder by TARGET.</p> <p>The one that passing the folder specified with TARGET was removed is specified among folders that become objects.</p> <p>When multiple designations are required, multiple keys need to be specified.</p>

	EXCEPTFILE		No	No	<p>[In case of C/C++]</p> <p>Specify the source files to be removed from the folder by TARGET.</p> <p>Specify the contents under the path other than the folder by TARGET in the objective folder.</p> <p>When multiple designations are required, multiple keys need to be specified.</p>
--	------------	--	----	----	--

	[OPERATECOMMONPROJECT]				Set the section for project common action.
	MESSAGE_COUNT				<p>Specify whether to calculate indication features.</p> <p>0 :Do not calculate indication feature</p> <p>1 :Calculate indication feature (Default value)</p> <p>When calculating indication feature, quote the indication feature definition file under the same folder as batch execution command (arqm command). For indication feature definition file, please refer to "<a href="#">4.2.4 Indication feature definition file</a>".</p>
	OPTION		No	No	<p>[In case of C/C++]</p> <p>Specify the pgr5 command option for the analysis of include and compiler. While the following options are invalid even being specified.</p> <p>-V, -E, --output_code, --project, -csv, -c, -G, -Z, --cplusplus</p> <p>If you wish to analyze the input file with extension .c as C++ source file, please specify .c for CPLUSPLUSEX KEY.</p> <p>Furthermore, about pgr5 command, please refer to [Agile+ Relief C/C++ Command Manual].</p>
	CLASSPATH	No		No	<p>[In case of Java]</p> <p>Specify the value of -classpath or environment variable CLASSPATH during javac command execution.</p> <p>Please describe this key by multiple lines when specifying multiple values.</p>
	SOURCEPATH	No		No	<p>[In case of Java]</p> <p>Specify the value of -sourcepath during javac command execution.</p> <p>Please describe this key by multiple lines when specifying multiple</p>
	JAVAC	No		No	<p>[In case of Java]</p> <p>Specify the option of javac command.</p>

PGRJAVA	No		No	<p>[In case of Java]</p> <p>Specify the option of pgrjava command. While the following options are invalid even being specified.</p> <p>-af, -r, -e</p> <p>Furthermore, about pgrjava command, please refer to [Agile+ Relief J Command Manual].</p>
RULEFILE			No	<p>Specify the definition file of message indication for check and definition file of rules for check for actual use.</p> <p>Multiple specification is allowed in case of C/C++. At that time, specify multiple keys.</p> <p>If it is not specified, then the default value will be set as the object to be checked. For the default value, please refer to [Default value] of "<a href="#">4.2.1 Check indication definition File and Check rule definition File</a>".</p>
METRICSFILE				<p>Specify the definition file for dangerous field/field for attention.</p> <p>When not specified, the definition file for dangerous field/field for attention saved in install folder of Agile+ Relief will be applied.</p>
INDICATORFILE				<p>Specify the definition file for diagnose indices.</p> <p>When not specified, the definition file for diagnose indices saved in install folder of Agile+ Relief will be applied.</p>
GROUPFILE		No	No	<p>[In case of C/C++]</p> <p>Specify the group change file.</p> <p>About group change file, please refer to [Agile+ Relief C/C++ Command Manual].</p>

MISRA-CVERSION		No		<p>[In case of C/C++]</p> <p>In source analysis mode, specify the version of MISRA to be checked. In analysis result consolidation mode, specify the version of MISRA specified when the pgrmisra command is executed.</p> <p>0 : Not Check for MISRA (Default)  1 : Check for MISRA-C V1  2 : Check for MISRA-C V2  3 : Check for MISRA-C V3  P1 : Check for MISRA-C++ V1</p> <p>MISRA-CVERSION and SEC-CVERSION cannot be specified at the same time.</p> <p>When specifying the RULEFILE key, please confirm the Check indication definition File along the specified rule. About Check indication definition File, please refer to "<a href="#">4.2.1 Check indication definition File and Check rule definition File</a>".</p>
SEC-CVERSION		No		<p>[In case of C/C++]</p> <p>In source analysis mode, specify the version of IPA/SEC-C/C++ to be checked. In analysis result consolidation mode, specify the version of IPA/SEC-C/C++ specified when the pgrsec command is executed.</p> <p>0 : Not Check for IPA/SEC-C/C++ (Default)  1 : Check for IPA/SEC-C V1  2 : Check for IPA/SEC-C V2  3 : Check for IPA/SEC-C V3  P1 : Check for IPA/SEC-C++ V1  P2 : Check for IPA/SEC-C++ V2</p> <p>MISRA-CVERSION and SEC-CVERSION cannot be specified at the same time.</p> <p>When specifying the RULEFILE key, please confirm the Check indication definition File along the specified rule. About Check indication definition File, please refer to "<a href="#">4.2.1 Check indication definition File and Check rule definition File</a>".</p>
ANALYZELINK		No	No	<p>[In case of C/C++]</p> <p>Specify whether multiple files analysis is applied.</p> <p>0 : Not Apply Multiple Files Analysis  1 : Apply Multiple Files Analysis (Default)</p>
[OPERATEPROJECT]			No	Set the section for project action.



PROJECTTARGET			No	Specify the project needs to be set separately. Specify the contents under the path other than the folder by TARGET in the objective folder. When there are two or more projects with individual settings, the required [OPERATEPROJECT] section shall be described.
PROJECTNAME			No	Specify the project name. Multibyte characters can be specified. Please specify within 256 bytes. When omitted, PROJECTTARGET will be used for the project name.
OPTION		No	No	About the contents for description, please refer to the illustration of key in [OPERATECOMMONPROJECT]. If the key is specified, the settings of [OPERATECOMMONPROJECT] will be invalid. [In case of C/C++] OPTION key and APPENDOPTION key cannot be specified at the same time. MISRA-CVERSION and SEC-CVERSION cannot be specified at the same time.
CLASSPATH	No		No	
SOURCEPATH	No		No	
JAVAC	No		No	
PGRJAVA	No		No	
RULEFILE			No	
METRICSFILE			No	
INDICATORFILE			No	
GROUPFILE		No	No	
MISRA-CVERSION		No	No	
SEC-CVERSION		No	No	
ANALYZELINK		No	No	
FILEOPTION		No	No	[In case of C/C++] If want to specify the analysis options separately in the unit of files beneath project directory, the format for specification shall be as follows: FILEOPTION = "objective file name", "option" The contents of objective file name are as follows: <ul style="list-style-type: none"> <li>● For the folder specified by PROJECTTARGET, -&gt;Specify the contents under the path other than the folder by TARGET in the objective folder.</li> <li>● For the folder specified by ADDDIR, ● Identical with the file name by ADDFILE -&gt;Specify the file name in full path.</li> </ul> The option to be described is the same with OPTION key of [OPERATECOMMONPROJECT]. When this key is specified, the settings of OPTION key of [OPERATECOMMONPROJECT] and that of [OPERATEPROJECT] will be invalid. FILEOPTION key and APPENDFILEOPTION key cannot be specified at the same time.

APPENDOPTION		No	No	<p>[In case of C/C++]</p> <p>Specify when the relevant options for special project analysis are appended.</p> <p>Both the contents specified by this key and that of OPTION key to [OPERATECOMMONPROJECT] are valid.</p> <p>OPTION key and APPENDOPTION key cannot be specified at the same time.</p>
APPENDFILEOPTION		No	No	<p>[In case of C/C++]</p> <p>Specify when the relevant options for special file analysis are appended. The format is identical with that of FILEOPTION key.</p> <p>Both the relevant analysis options specified by this key and those by project are valid.</p> <p>FILEOPTION key and APPENDFILEOPTION key cannot be specified at the same time.</p>
ADDDIR		No	No	<p>[In case of C/C++]</p> <p>Specify when the folders other than those under the folder by TARGET are regarded as the objectives for check.</p> <p>When multiple designations are required, multiple keys need to be specified.</p>
ADDFILE		No	No	<p>[In case of C/C++]</p> <p>Specify when the files other than those under the folder by TARGET are regarded as the objectives for check.</p> <p>When multiple designations are required, multiple keys need to be specified.</p>
[REPORTOPTION]				Set the section of relevant options for quality info result.
RANGE				<p>Specify the date, days and unit for the output of quality transition.</p> <p>Please specify as follows:</p> <p>Days for Output, Unit for Output</p> <p>Days for Output:2~14</p> <p>Unit for Output: Specify among D(Day), W(Week) and M(Month).</p> <p>Cannot be specified at the same time with SPOTDATE.</p>

	SPOTDATE			<p>Specify the specific date for output of quality transition.</p> <p>Please specify as follows:</p> <p>yyyy/mm/dd</p> <p>yyyy : Western Calendar (4bits Value)</p> <p>mm : Month (2bits Value. 0 starts for 1bit value)</p> <p>dd : Day (2bits Value. 0 starts for 1bit value)</p> <p>When multiple designations are required, multiple keys need to be specified. However, the maximum to specify is 14.</p> <p>In addition, it cannot be specified at the same time with RANGE.</p>
	[VIEWEROPTION]		No	For setting the section of related option linking with indication message viewer.
	RESULTSHARE DPATH		No	<p>If the drive name for saving project structure definition file is different, which are server and client respectively, specify a shared path name.</p> <p>Once this KEY is specified, the path needs not to be set in the "Path Replacement Preferences" dialog box when starting the indication message viewer.</p>

■ Section and Key (for version management tools)

When extracting objective asset for check from the version management tools, the section and KEY to be set are shown in the following table.

Section and Key	Y: Must be specified N: Cannot be specified					Description
	source analysis				analysis result consolidation	
	C V S	S V N	V S S	A D M		
[VERSIONSYSTEM_SOURCES]						This section is used for setting relevant options when linking with the version management tools
TOOLTYPE	Y	Y	Y	Y	N	Linked version management tools are all specified with lowercase letters. Linked with CVS : cvs Linked with Subversion : svn Linked with VisualSourceSafe: vss Linked with Interstage Application Development Cycle Manager: adm
BRANCH		N	N	N	N	Specify the branch name when extracting objective asset for check from the version management tools. Straight line when it is omitted.
ROOT	Y	N	N	N	N	Specify the repository of version management tools.
MODULENAME	Y	Y	Y	N	N	Specify the module name or path name that is extracted from version management tools as objective asset for check.
USERNAME	N		Y	N	N	Specify the registered user name for accessing the repository of version management tools. When it is omitted in Subversion, the repository can be access as no specified user name.
PASSWORD	N		Y	N	N	Specify the password of registered user name for accessing the repository of version management tools. When it is omitted in Subversion, the repository can be access as no specified password. In addition, since this KEY is used for recording the password, please pay much attention to the management of the project structure definition file.

	TOOLPATH					N	Specify the startup path of version management tools. When it is omitted, the version management tools will be started by each command name.
	COMMANDOPTION	N	N	N	Y	N	Specify the command option of the linked version management tool.

■ Notes

- The midway file is generated during the first stage of quality info result within the first layer of folder, in which the file is saved. For the necessary information will be saved during the output of project group/project quality transition, please do not delete. In addition, please save the file on the disk with sufficient free space.
- When the folder name or file name specified for the key other than OPTION, APPENDOPTION, FILEOPTION, APPENDFILEOPTION, TOOLPATH, please do not embrace with double quote (").
- When extracting objective asset for check from the version management tools, pay attention to the following items according to different needs of the tools being used.

[In common CVS / Subversion / VisualSourceSafe /

Interstage Application Development Cycle Manager]

- Asset under the folder directory designated by TARGET will be updated. Therefore, even the source program file is edited, the result of editing will be lost.

- When input request from the version management tool is generated by the input mistake etc. of the password while taking out the property, there might be no response. Please end the batch execution command (arqm command) and confirm each key to the [VERSIONSYSTEM\_SOURCES] section is set correctly.

[When using Subversion]

- When extracting the objective asset for check using other MODULENAME, the error on Subversion will be output sometimes, and the objective asset for check cannot be extracted normally. Please execute batch execution command (arqm command) after deleting the folder specified by TARGET manually based on needs.

[When using VisualSourceSafe]

- When the objective asset for check has been deleted, or the objective asset for check is extracted using other MODULENAME, the unnecessary files or folders under the folder directory designated by TARGET will be not deleted. Please execute batch execution command (arqm command) after deleting the folder specified by TARGET manually based on needs.

[When using Interstage Application Development Cycle Manager]

- Please specify the option of "escm export" command to the COMMANDOPTION key. While the following options are invalid even being specified.

```
-t <directory name>
```

- When the objective asset for check has been deleted, or the objective asset for check is extracted at other storage locations, the unnecessary files or folders under the folder directory designated by TARGET will be not deleted. Please execute batch execution command (arqm command) after deleting the folder specified by TARGET manually based on needs.

- When "Definition of remote mode access" should be used during the asset extracting, please set "Definition of remote mode access" (execute escm set server) first and execute batch execution command (arqm command). When the batch execution command (arqm command) is executed without the setting, the request for connection information input from Interstage Application Development Cycle Manager will occur and the batch execution command will not be executed.

#### ■ Sample

The followings are the record samples when the version management tools are not linked.

(In case of C/C++)

```
-----  
; Set the objective asset for check  
[SOURCES]  
TARGET          = C:¥SamplePrj¥src¥  
CODE            = SJIS  
CPLUSPLUSEXT   = .cpp  
CPLUSPLUSEXT   = .CC  
BINDPROJECT    = common  
EXCEPTDIR    = common¥worksrc // <-The information other than that under the  
                                     // directory of C:¥SamplePrj¥src¥common¥worksrc  
EXCEPTFILE   = View¥main¥main_old.c // <-The files other than that under the directory  
                                     // of C:¥SamplePrj¥src¥View¥main¥main_old.c  
-----  
; Set project common action  
[OPERATECOMMONPROJECT]  
OPTION          = -lc:¥include -DDEBUG  
RULEFILE        = C:¥arsetting¥comrule.txt  
RULEFILE        = C:¥arsetting¥grprule.txt  
METRICSFILE     = C:¥arsetting¥metrics.txt  
INDICATORFILE   = C:¥arsetting¥shihyo.txt  
GROUPFILE       = C:¥arsetting¥group.txt  
-----  
; Set project related action  
[OPERATEPROJECT]  
PROJECTTARGET  = Analyze  
PROJECTNAME    = Analyze  
OPTION          = -lc:¥include -DRELEASE  
FILEOPTION     = "common¥option.c", "-lc:¥include -lc:¥com_inc -DDEBUG"  
                                     //Set file option of C:¥SamplePrj¥src¥common¥option.c  
ADDDIR         = D:¥guisrc¥mainwnd  
ADDFILE        = D:¥guisrc¥dlg¥fileselect.c  
ADDFILE        = D:¥guisrc¥dlg¥folderselect.c  
-----  
; Settings during report output  
[REPORTOPTION]  
RANGE = 4,W
```

(In case of Java)

```
-----  
; Set the objective asset for check  
[SOURCES]  
TARGET          = C:¥SampleJava¥  
CODE             = SJIS  
LANG            = Java  
PROJECTLEVEL    = 1  
JAVAEXT         = .xml  
JAVAEXT         = .jsp  
-----  
; Set project common action  
[OPERATECOMMONPROJECT]  
RULEFILE        = C:¥arsetting¥rule.pgrj  
METRICSFILE     = C:¥arsetting¥metrics.txt  
INDICATORFILE   = C:¥arsetting¥shihyo.txt  
-----  
; Set project related action  
[OPERATEPROJECT]  
PROJECTTARGET   = Project1  
CLASSPATH       = C:¥Common¥lib <- Classpath is set refers by project 'Project1'  
-----  
; Settings during report output  
[REPORTOPTION]  
RANGE = 4,W
```

(In case of analysis result consolidation mode)

```
-----  
; Set the objective asset for consolidation  
[SOURCES]  
MODE            = RESULTMERGE <- Set analysis result consolidation mode  
TARGET          = C:¥SamplePrj¥result¥  
LANG            = C  
CODE            = SJIS  
MSGFILEEXT      = .message  
MISRAFILEEXT    = .misra  
-----  
; Set project common action  
[OPERATECOMMONPROJECT]  
METRICSFILE     = C:¥arsetting¥metrics.txt  
INDICATORFILE   = C:¥arsetting¥shihyo.txt  
MISRA-CVERSION  = 2  
-----  
; Settings during report output  
[REPORTOPTION]  
RANGE = 4,W
```

Followings are the record samples when linking with the version management tool "CVS".  
 Except for the [VERSIONSYSTEM\_SOURCES] section, all are the same as the sample when the version management tools are not linked. Following is the example which sets the source program of C/C++ as the object.

```

;-----
; Set the objective asset for check
[SOURCES]
TARGET          = C:¥SamplePrj¥src¥
CODE            = SJIS
CPLUSPLUSEXT   = .cpp
CPLUSPLUSEXT   = .CC
BINDPROJECT     = common
EXCEPTDIR     = common¥worksrc // <-The information other than that under the
                                     // directory of C:¥SamplePrj¥src¥common¥worksrc
EXCEPTFILE    = View¥main¥main_old.c // <-The files other than that under the directory
                                     // of C:¥SamplePrj¥src¥View¥main¥main_old.c
;-----
; Set project common action
[OPERATECOMMONPROJECT]
OPTION          = -lc:¥include -DDEBUG
RULEFILE        = C:¥arsetting¥comrule.txt
RULEFILE        = C:¥arsetting¥grprule.txt
METRICSFILE     = C:¥arsetting¥metrics.txt
INDICATORFILE   = C:¥arsetting¥shihyo.txt
GROUPFILE       = C:¥arsetting¥group.txt
;-----
; Set project related action
[OPERATEPROJECT]
PROJECTTARGET   = Analyze
PROJECTNAME     = Analyze
OPTION          = -lc:¥include -DRELEASE
FILEOPTION      = "common¥option.c", "-lc:¥include -lc:¥com_inc -DDEBUG"
                                     //Set file option of C:¥SamplePrj¥src¥common¥option.c
ADDDIR          = D:¥guisrc¥mainwnd
ADDFILE         = D:¥guisrc¥dlg¥fileselect.c
ADDFILE         = D:¥guisrc¥dlg¥folderselect.c
;-----
; Settings during report output
[REPORTOPTION]
RANGE = 4,W
;-----
; Set the version management tool "CVS"
[VERSIONSYSTEM_SOURCES]
TOOLTYPE        = cvs
ROOT            = :local:e:¥sample¥qmsample_CVSROOT
MODULENAME      = SampleSrc
BRANCH         = Branch01
TOOLPATH        = "C:¥Program Files¥cvs"

```



Followings are the record samples when linking with version management tool "Subversion".  
 Except for the [VERSIONSYSTEM\_SOURCES] section, all are the same as the sample when the version management tools are not linked. Following is the example which sets the source program of C/C++ as the object.

```

;-----
; Set the objective asset for check
[SOURCES]
TARGET          = C:¥SamplePrj¥src¥
CODE            = SJIS
CPLUSPLUSEXT   = .cpp
CPLUSPLUSEXT   = .CC
BINDPROJECT     = common
EXCEPTDIR     = common¥worksrc // <-The information other than that under the
                                     // directory of C:¥SamplePrj¥src¥common¥worksrc
EXCEPTFILE    = View¥main¥main_old.c // <-The files other than that under the directory
                                     // of C:¥SamplePrj¥src¥View¥main¥main_old.c
;-----
; Set project common action
[OPERATECOMMONPROJECT]
OPTION          = -lc:¥include -DDEBUG
RULEFILE       = C:¥arsetting¥comrule.txt
RULEFILE       = C:¥arsetting¥grprule.txt
METRICSFILE    = C:¥arsetting¥metrics.txt
INDICATORFILE  = C:¥arsetting¥shihyo.txt
GROUPFILE      = C:¥arsetting¥group.txt
;-----
; Set project related action
[OPERATEPROJECT]
PROJECTTARGET  = Analyze
PROJECTNAME    = Analyze
OPTION         = -lc:¥include -DRELEASE
FILEOPTION     = "common¥option.c", "-lc:¥include -lc:¥com_inc -DDEBUG"
                                     //Set file option of C:¥SamplePrj¥src¥common¥option.c
ADDDIR        = D:¥guisrc¥mainwnd
ADDFILE       = D:¥guisrc¥dlg¥fileselect.c
ADDFILE       = D:¥guisrc¥dlg¥folderselect.c
;-----
; Settings during report output
[REPORTOPTION]
RANGE = 4,W
;-----
; Set the version management tool "Subversion"
[VERSIONSYSTEM_SOURCES]
TOOLTYPE       = svn
MODULENAME     = file:///e:/sample/qmsample_SVNROOT/SampleSrc
TOOLPATH       = "C:¥Program Files¥svn"

```

Followings are the record samples when linking with version management tool "VisualSourceSafe".

Except for the [VERSIONSYSTEM\_SOURCES] section, all are the same as the sample when the version management tools are not linked. Following is the example which sets the source program of C/C++ as the object.

```
-----  
; Set the objective asset for check  
[SOURCES]  
TARGET      = C:¥SamplePrj¥src¥  
CODE        = SJIS  
CPLUSPLUSEXT = .cpp  
CPLUSPLUSEXT = .CC  
BINDPROJECT = common  
EXCEPTDIR  = common¥worksrc // <-The information other than that under the  
                                     // directory of C:¥SamplePrj¥src¥common¥worksrc  
EXCEPTFILE = View¥main¥main_old.c // <-The files other than that under the directory  
                                     // of C:¥SamplePrj¥src¥View¥main¥main_old.c  
-----  
; Set project common action  
[OPERATECOMMONPROJECT]  
OPTION      = -lc:¥include -DDEBUG  
RULEFILE    = C:¥arsetting¥comrule.txt  
RULEFILE    = C:¥arsetting¥grprule.txt  
METRICSFILE = C:¥arsetting¥metrics.txt  
INDICATORFILE = C:¥arsetting¥shihyo.txt  
GROUPFILE   = C:¥arsetting¥group.txt  
-----  
; Set project related action  
[OPERATEPROJECT]  
PROJECTTARGET = Analyze  
PROJECTNAME   = Analyze  
OPTION        = -lc:¥include -DRELEASE  
FILEOPTION    = "common¥option.c", "-lc:¥include -lc:¥com_inc -DDEBUG"  
                                     //Set file option of C:¥SamplePrj¥src¥common¥option.c  
ADDDIR        = D:¥guisrc¥mainwnd  
ADDFILE       = D:¥guisrc¥dlg¥fileselect.c  
ADDFILE       = D:¥guisrc¥dlg¥folderselect.c  
-----  
; Settings during report output  
[REPORTOPTION]  
RANGE = 4,W  
-----  
; Set the version management tool "VisualSourceSafe"  
[VERSIONSYSTEM_SOURCES]  
TOOLTYPE      = vss  
MODULENAME    = $/SampleSrc  
USERNAME      = user1  
PASSWORD      = pw1  
TOOLPATH      = "C:¥Program Files¥vss"
```

Followings are the record samples when linking with version management tool "Interstage Application Development Cycle Manager".

Apart from the [VERSIONSYSTEM\_SOURCES] sections, all are different from the sample when the version management tools are not linked. Following is the example which sets the source program of C/C++ as the object.

```

;-----
; Set the objective asset for check
[SOURCES]
TARGET          = C:¥SamplePrj¥src¥
CODE            = SJIS
CPLUSPLUSEXT   = .cpp
CPLUSPLUSEXT   = .CC
BINDPROJECT    = common
EXCEPTDIR     = common¥worksrc // <-The information other than that under the
                                     // directory of C:¥SamplePrj¥src¥common¥worksrc
EXCEPTFILE    = View¥main¥main_old.c // <-The files other than that under the directory
                                     // of C:¥SamplePrj¥src¥View¥main¥main_old.c
;-----
; Set project common action
[OPERATECOMMONPROJECT]
OPTION          = -lc:¥include -DDEBUG
RULEFILE       = C:¥arsetting¥comrule.txt
RULEFILE       = C:¥arsetting¥grprule.txt
METRICSFILE    = C:¥arsetting¥metrics.txt
INDICATORFILE  = C:¥arsetting¥shihyo.txt
GROUPFILE      = C:¥arsetting¥group.txt
;-----
; Set project related action
[OPERATEPROJECT]
PROJECTTARGET  = Analyze
PROJECTNAME    = Analyze
OPTION         = -lc:¥include -DRELEASE
FILEOPTION     = "common¥option.c", "-lc:¥include -lc:¥com_inc -DDEBUG"
               //Set file option of C:¥SamplePrj¥src¥common¥option.c
ADDDIR         = D:¥guisrc¥mainwnd
ADDFILE        = D:¥guisrc¥dlg¥fileselect.c
ADDFILE        = D:¥guisrc¥dlg¥folderselect.c
;-----
; Settings during report output
[REPORTOPTION]
RANGE = 4,W
;-----
; Set the version management tool "Interstage Application Development Cycle Manager"
[VERSIONSYSTEM_SOURCES]
TOOLTYPE      = adm
TOOLPATH      = "C:¥Fujitsu¥InterstageASM¥commandline"
COMMANDOPTION = -eu admsrv¥user1 -ep pw1 -o -s admsrv -d qmsample_ADMROOT
               -p SampleSrc -r 1.0 -c "MAIN"

```

### 4.1.2. Project Structure Definition Listing File

- File Name

The format of project structure definition listing file is described as follows:

any.pgl

- Creation

The template is saved in the file name of [projectgroup.pgl] to the same folder [definition\_template] with that of batch execution command (arqm command). The necessary settings for template file for reference are described as follows:

- Format

The format is described as follows: When ";" (Semicolon) is described at the beginning of a line, this line will be invalid.

```
[section]
key = Value
key = Value
:
:
[section]
key = Value
:
```

- Section and Key

The necessary settings for section and key are described as follows:

Section and Key	Required	Description
[PROJECTLIST]	Yes	Set the section for the combined project group list.
PROJECTFILE	Yes	Specify the combined project structure definition file and project structure definition listing file. When multiple designations are required, multiple keys need to be specified.
[REPORTOPTION]		Set the section of relevant options for quality info result.
RANGE		Please refer to the illustration of project structure definition file.
SPOTDATE		

- Notes

When specifying the file name, please do not embrace with double quote (").

■ Sample

The sample is described as follows:

```
-----  
; Set the project structure definition file for application  
[PROJECTLIST]  
PROJECTFILE = C:¥ArSetting¥ModelPrj.pgs  
PROJECTFILE = E:¥arprojects¥ViewPrj.pgs  
PROJECTFILE = E:¥arprojects¥ControlPrj.pgs  
-----  
; Settings during report output  
[REPORTOPTION]  
SPOTDATE = 2018/08/15  
SPOTDATE = 2018/09/15  
SPOTDATE = 2018/10/15  
SPOTDATE = 2018/11/15  
SPOTDATE = 2018/12/15  
SPOTDATE = 2019/01/15  
SPOTDATE = 2019/02/15
```

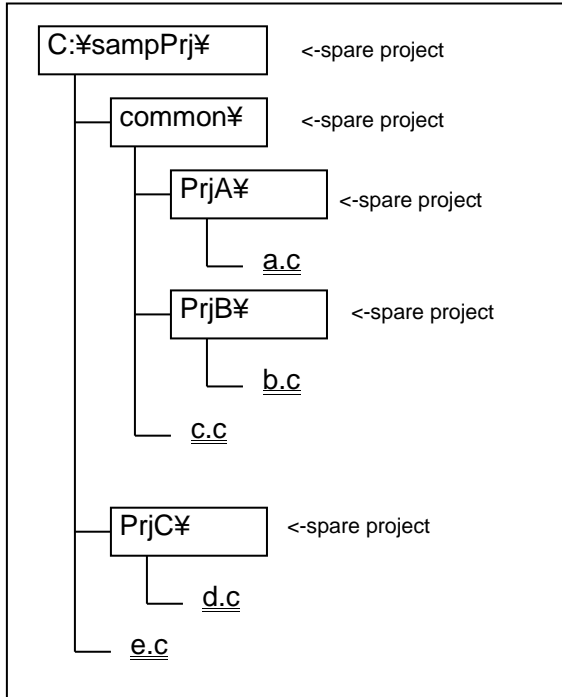
### 4.1.3. Example to Designate Target Folder for Project Creation

#### 4.1.3.1. Example to Designate Target Folder under source analysis mode

The example explains the relationship between project and source file in C/C++, and it is the same for Java.

#### 1. Standard Pattern (Auto Recognized Project)

##### ■ Asset Folder Structure



##### ■ Specify Project Structure Definition File (SampleProject.pgs)

```
[SOURCES]
TARGET = C:\$sampPrj¥
```

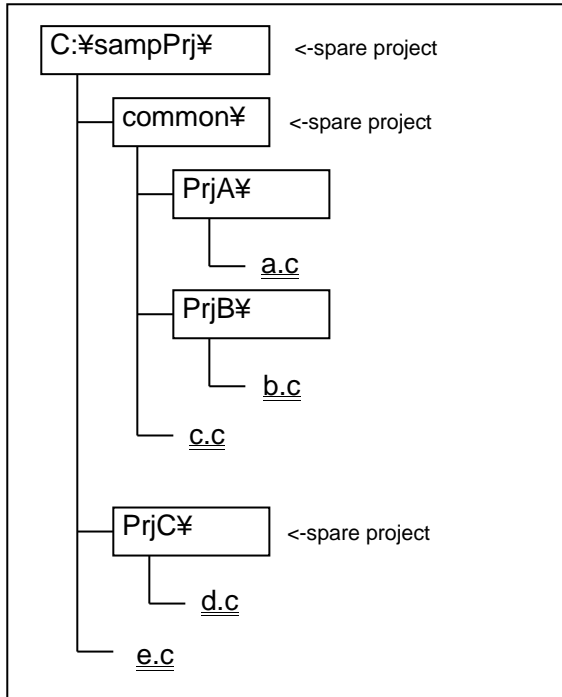
##### ■ Generated Project

Project	Objective File for Check
SampleProject	C:\\$sampPrj¥e.c
common	C:\\$sampPrj¥common¥c.c
PrjA	C:\\$sampPrj¥common¥PrjA¥a.c
PrjB	C:\\$sampPrj¥common¥PrjB¥b.c
PrjC	C:\\$sampPrj¥PrjC¥d.c

The project (project of [SampleProject]) of project structure definition file name will be generated for the source file (e.c) of the first layer by TARGET.

## 2. Exceptional Pattern (Explicit Project)

### ■ Asset Folder Structure



### ■ Specify Project Structure Definition File (SampleProject.pgs)

```

[SOURCES]
TARGET      = C:\$sampPrj\
BINDPROJECT = common
  
```

### ■ Generated Project

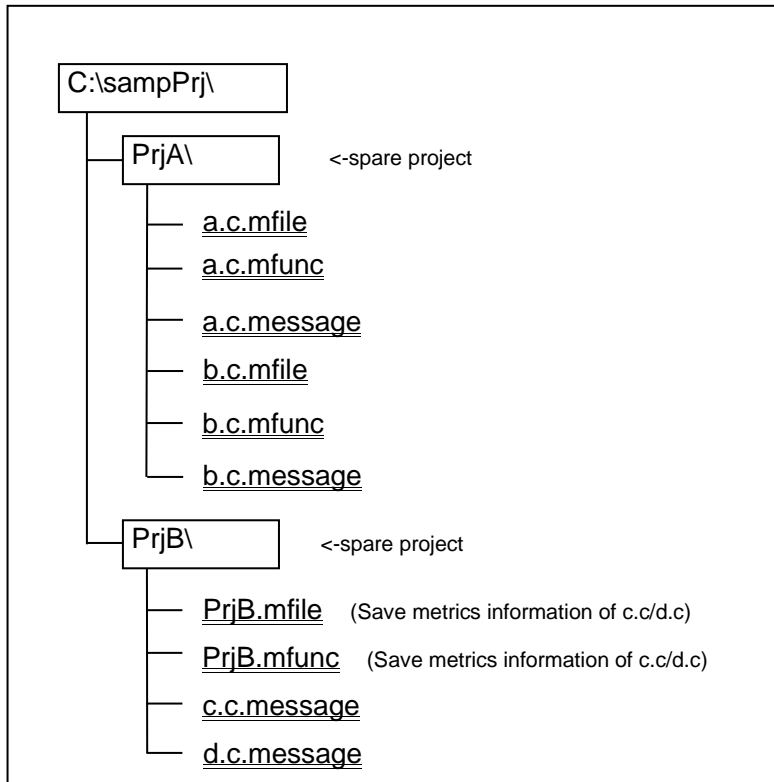
Project	Objective File for Check
SampleProject	C:\\$sampPrj\e.c
common	C:\\$sampPrj\common\PrjA\ a.c C:\\$sampPrj\common\PrjB\ b.c C:\\$sampPrj\common\ c.c
PrjC	C:\\$sampPrj\PrjC\ d.c

#### 4.1.3.2. Example to Designate Target Folder under analysis result consolidation mode

The example explains the relationship between project and source file in C/C++, and it is the same for Java.

##### 1. Standard Pattern

###### ■ Asset Folder Structure



###### ■ Specify Project Structure Definition File (SampleProject.pgs)

```

[SOURCES]
MODE = RESULTMERGE
TARGET = C:¥sampPrj¥
  
```

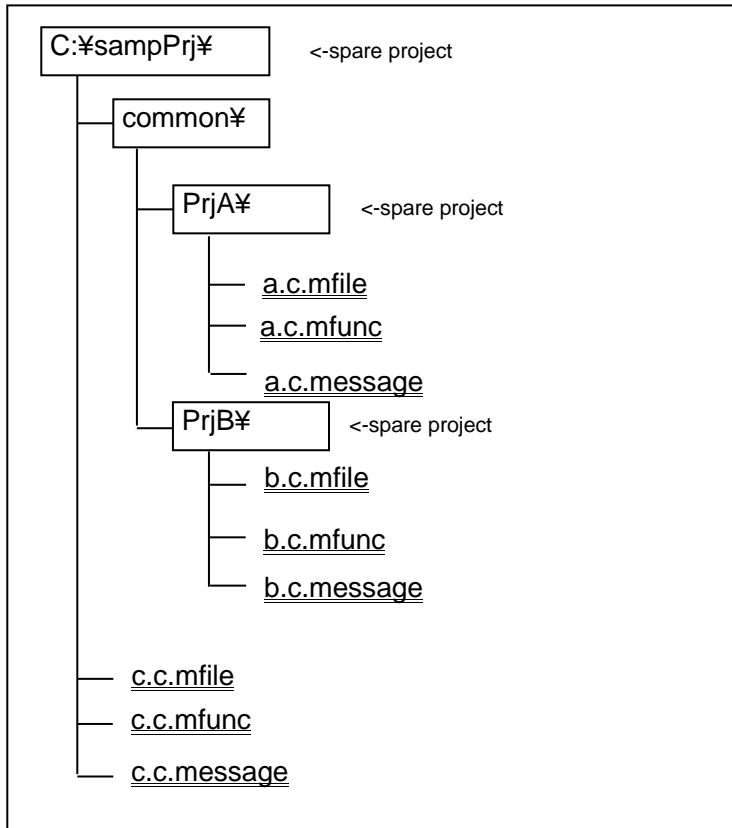
###### ■ Generated Project

Project	Consolidation Object Asset
PrjA	C:¥sampPrj¥PrjA¥a.c.mfile C:¥sampPrj¥PrjA¥a.c.mfunc C:¥sampPrj¥PrjA¥a.c.message C:¥sampPrj¥PrjA¥b.c.mfile C:¥sampPrj¥PrjA¥b.c.mfunc C:¥sampPrj¥PrjA¥b.c.message
PrjB	C:¥sampPrj¥PrjB¥PrjB.mfile C:¥sampPrj¥PrjB¥PrjB.mfunc C:¥sampPrj¥PrjB¥c.c.message C:¥sampPrj¥PrjB¥d.c.message



## 2. The pattern with asset right under TARGET

### ■ Asset Folder Structure



### ■ Specify Project Structure Definition File (SampleProject.pgs)

```

[SOURCES]
MODE = RESULTMERGE
TARGET = C:¥sampPrj¥
  
```

### ■ Generated Project

Project	Consolidation Object Asset
SampleProject	C:¥sampPrj¥c.c.mfile C:¥sampPrj¥c.c.mfunc C:¥sampPrj¥c.c.message
PrjA	C:¥sampPrj¥common¥PrjA¥a.c.mfile C:¥sampPrj¥common¥PrjA¥a.c.mfunc C:¥sampPrj¥common¥PrjA¥a.c.message
PrjB	C:¥sampPrj¥common¥PrjB¥b.c.mfile C:¥sampPrj¥common¥PrjB¥b.c.mfunc C:¥sampPrj¥common¥PrjB¥b.c.message

\*The project (project of [SampleProject]) of project structure definition file name will be generated for the analysis result (c.c.xxx) under TARGET.

## 4.2. Definition of Check Base

Product manager (QMO) defines the guarding base for Agile+ Relief use in organization. According to the status of organization, four types of definition files, such as definition file of indication messages for check, definition file of rules for check, definition file of dangerous field/field for attention, definition file of diagnose indices shall be created.

### 4.2.1. Check indication definition File and Check rule definition File

Define the indication messages and rules abided by in the organization.

#### 4.2.1.1. The checking rule definition in case of C/C++

##### ■ File Name

The name of definition file of indication messages for check can be anything, also anywhere is available for the location to create.

##### ■ Creation

The template is saved in the same folder [definition\_template] as the batch execution command (arqm command) with the following file name.

[set\_messageindication.def] for Agile+ Relief C/C++ indication,  
[set\_messageindication\_misrav1.def] for MISRA-C V1 rule,  
[set\_messageindication\_misrav2.def] for MISRA-C V2 rule,  
[set\_messageindication\_misrav3.def] for MISRA-C V3 rule,  
[set\_messageindication\_misrapv1.def] for MISRA-C ++ V1 rule,  
[set\_messageindication\_secv1.def] for IPA/SEC-C V1 rule,  
[set\_messageindication\_secv2.def] for IPA/SEC-C V2 rule,  
[set\_messageindication\_secv3.def] for IPA/SEC-C V3 rule,  
[set\_messageindication\_secpv1.def] for IPA/SEC-C ++ V1 rule,  
[set\_messageindication\_secpv2.def] for IPA/SEC-C ++ V2 rule,

The necessary settings for template file for reference are described as follows:

##### ■ Format

The format is described as follows:

(for Agile+ Relief C/C++ Indication)

```
;Rule=PGRelief
Message ID or Message Group
:
:
```

(for MISRA-C Rule)

```
;Rule=MISRA-C [V1 or V2 or V3]
Rule ID or Agile+ Relief C/C++ Message Group
:
:
```

(for MISRA-C++ Rule)

```
;Rule=MISRA-C++ V1
Rule ID or Agile+ Relief C/C++ Message Group
:
:
```

(for IPA/SEC-C Rule)

```
;Rule=SEC-C [V1 or V2 or V3]
Rule ID or Agile+ Relief C/C++ Message Group
:
:
```

(for IPA/SEC-C++ Rule)

```
;Rule=SEC-C++ [V1 or V2]
Rule ID or Agile+ Relief C/C++ Message Group
:
:
```

#### ■ Notes

- The types shall be specified at the first line of file.
- When ";" (semicolon) is found at the beginning of a line, this line will be invalid.
- Please specify a 4bits value other than prefix "pgr" for message ID.
- When specifying the message group for the indication messages in Agile+ Relief C/C++, please use [a] or [b] such lowercase English letters.
- For rule ID of IPA/SEC-C/C++, when the standardization is selected, please record until the number with braces.
- When specifying the message group for MISRA rule or IPA/SEC-C/C++ rule in Agile+ Relief C/C++, please use [group-a] or [group-b], such lowercase English letters after [group-] to specify the message group of Agile+ Relief C/C++.

#### ■ Sample

The sample is described as follows:

(for Agile+ Relief C/C++ Indication Message)

```
;Rule=PGRelief
a          <- take the message ID affiliated to group a as the objective
0300
0301
;0302     <- comment line = Indication not for Check
0304
```

(for MISRA-C Rule)

```
;Rule=MISRA-C V1
1
2
;3      <- comment line =Rule not for Check
4
group-a
```

(for MISRA-C++ Rule)

```
;Rule=MISRA-C++ V1
0-1-1
0-1-2
;0-1-3  <-comment line=Rule not for Check
0-1-4
group-a
```

(for IPA/SEC-C Rule)

```
;Rule=SEC-C V1
R1.1.1
M1.2.1(1)
;M1.2.1(2) <-comment line=Rule not for Check
M1.7.1
group-a
```

(for IPA/SEC-C++ Rule)

```
;Rule=SEC-C++ V1
R1.1.1
M1.2.1(1)
;M1.2.1(2) <-comment line=Rule not for Check
M1.7.1
group-a
```

#### ■ Default Value

When the specification of this file is omitted, the definition file of indication messages for check saved in the same folder [qualityinfo] with that of batch execution command (arqm command) will be referenced.

In case of Agile+ Relief C/C++ indication, the indication group [a] is defined, and the file name is [msgindication.qdf].

For MISRA-C rule, when the required rule, mandatory rule(Only MISRA-C V3), and message group [a] are defined, the file name of MISRA-C V1 will be [msgindication\_misrav1.qdf], the file name of MISRA-C V2 will be [msgindication\_misrav2.qdf], the file name of MISRA-C V3 will be [msgindication\_misrav3.qdf], the file name of MISRA-C++ V1 will be [msgindication\_misrapv1.qdf].

For IPA/SEC-C/C++ rule, when the pointer[o] and [\*] for rules and message group [a] are defined, the file name of IPA/SEC-C V1 will be [msgindication\_secv1.qdf], the file name of IPA/SEC-C V2 will be [msgindication\_secv2.qdf], the file name of IPA/SEC-C++ V1 will be [msgindication\_secpv1.qdf], the file name of IPA/SEC-C++ V2 will be [msgindication\_secpv2.qdf].

#### 4.2.1.2. The checking rule definition in case of Java

##### ■ File Name

The name of definition file of rules for check can be anyone, also anywhere is available for the location to create.

##### ■ Creation

The template is saved with file name [rule.pgrj] in [definition\_template], the same folder as batch execution command (arqm command).

Create by using the rule customization of Agile+ Relief J.

In addition, for the use of rule customization, please refer to the additional manual [Agile+ Relief J Customizer Operation Guide].

##### ■ Default Value

When the specification of this file is omitted, the check rule definition file [rule\_def.pgrj] saving in the folder [qualityinfo], which is same as the batch execution command (arqm command) will be referenced.

#### 4.2.2. Dangerous /Attention Field Definition file

Define the scope of dangerous field/field for attention when performing the judgment of metrics value (nest count, complexity, etc.).

##### ■ File Name

The name of definition file of dangerous field/field for attention can be anyone, also anywhere is available for the location to create.

##### ■ Creation

The template is saved in the file name of [set\_cautionval.def] to the same folder [definition\_template] with that of batch execution command (arqm command). The necessary settings for template file for reference are described as follows:

##### ■ Format

The format is described as follows:

Line, Japanese label, English label, <b>valid or invalid flag, value for attention, value for dangerous field</b>
---

Furthermore, please do not edit the line, Japanese label and English label. Otherwise, the normal running cannot be guaranteed.

Item Name	Description
Line	Column location information of metrics. User cannot edit due to fixed value.
Japanese Label	Item name of metrics information (Japanese Name). For the value is fixed by the users, thus no edition is available.

English Label	Item name of metrics information (English Name). For the value is fixed by the users, thus no edition is available.
<b>Invalid or Valid Flag</b>	Set whether to check the metrics information displayed by Japanese label, English label. TRUE : Take as objectives for check. FALSE : Not take as objectives for check.
<b>Value for Attention</b>	Specify the value for attention displayed by Japanese label, English label with integer.
<b>Value for Dangerous Field</b>	Specify the value for dangerous field displayed by Japanese label, English label with integer.

About the following setting items, specify the invalid or valid flag, value for attention, value for dangerous field.

About the contents of each setting item, please refer to "3.1 Metrics Structure" of [Agile+ Relief C/C++ Command Manual].

Line	Japanese label	English label	Description for Value Settings
3	total lines	Number of total lines	Specify whether to take "Total Lines" as flag, value for attention or dangerous field of objectives for check.
4	Run Lines	Number of execution lines	Specify whether to take "Run Lines" as flag, value for attention or dangerous field of objectives for check.
5	Nest Count	Number of nesting levels	Specify whether to take "Nest Count" as flag, value for attention or dangerous field of objectives for check.
6	Complexity 1	Complexity1	Specify whether to take "Complexity 1" as flag, value for attention or dangerous field of objectives for check.
7	Complexity 2	Complexity2	Specify whether to take "Complexity 2" as flag, value for attention or dangerous field of objectives for check.
9	return	return	Specify whether to take "Return Count" as flag, value for attention or dangerous field of objectives for check.
10	break	break	Specify whether to take "Break Count" as flag, value for attention or dangerous field of objectives for check.
11	continue	continue	Specify whether to take "Continue Count" as flag, value for attention or dangerous field of objectives for check.
12	if1	if1	Specify whether to take "if1 Count" as flag, value for attention or dangerous field of objectives for check.
13	while1	while1	Specify whether to take "while1 Count" as flag, value for attention or dangerous field of objectives for check.
14	dowhile1	dowhile1	Specify whether to take "dowhile1 Count" as flag, value for attention or dangerous field of objectives for check.
15	for1	for1	Specify whether to take "for1 Count" as flag, value for attention or dangerous field of objectives for check.

16	switch1	switch1	Specify whether to take "switch1 Count" as flag, value for attention or dangerous field of objectives for check.
17	if2	if2	Specify whether to take "if2 Count" as flag, value for attention or dangerous field of objectives for check.
18	while2	while2	Specify whether to take "while2 Count" as flag, value for attention or dangerous field of objectives for check.
19	dowhile2	dowhile2	Specify whether to take "dowhile2 Count" as flag, value for attention or dangerous field of objectives for check.
20	for2	for2	Specify whether to take "for2 Count" as flag, value for attention or dangerous field of objectives for check.
21	switch2	switch2	Specify whether to take "switchi2 Count" as flag, value for attention or dangerous field of objectives for check.

■ Sample

The sample is described as follows:

```

;line, Japanese label,English label,Invalid or Valid flag,Value for Attention,Value of
Dangerous Field
3,Total Lines,Number of total lines,TRUE,400,600
4,Run Lines,Number of execution lines,TRUE,200,400
5,Nest Count,Number of nesting levels,TRUE,10,20
6,Complexity1,Complexity1,TRUE,20,50
7,Complexity2,Complexity2,FALSE,20,50
9,return,return,FALSE,10,20
10,break,break,FALSE,10,20
11,continue,continue,FALSE,10,20
12,if1,if1,TRUE,10,20
13,while1,while1,TRUE,10,20
14,dowhile1,dowhile1,TRUE,10,20
15,for1,for1,TRUE,10,20
16,switch1,switch1,TRUE,10,20
17,if2,if2,FALSE,10,20
18,while2,while2,FALSE,10,20
19,dowhile2,dowhile2,FALSE,10,20
20,for2,for2,FALSE,10,20
21,switch2,switch2,FALSE,10,20

```

■ Default Value

If this file is not specified in the project configuration definition file, the file name 'qualityinfo' under the folder name 'cautionval.cfg' in the folder where the batch execution command (arqm command) is stored is referred to.

The following values are specified in this dangerous /attention field definition file.

Item	Invalid or Valid Flag	Value for Attention	Value for Dangerous Field
total lines	TRUE	400	600
Run Lines	TRUE	200	400
Nest Count	TRUE	10	20
Complexity 1	TRUE	20	50
Complexity 2	FALSE	20	50
return	FALSE	10	20
break	FALSE	10	20
continue	FALSE	10	20
if1	TRUE	10	20
while1	TRUE	10	20
dowhile1	TRUE	10	20
for1	TRUE	10	20
switch1	TRUE	10	20
if2	FALSE	10	20
while2	FALSE	10	20
dowhile2	FALSE	10	20
for2	FALSE	10	20
switch2	FALSE	10	20



### 4.2.3. Diagnosis Indicator Definition File

Define the comprehensive diagnose indices according to the result of definition file for rules for check, definition file of dangerous field/field for attention.

The definition is applied to project. And the total quality status of objective source file for check is compared with the conditions defined by diagnoses indices.

In addition, for the quality status of top project group of projects, the worst value of project quality status within the directory will be taken as the project group quality info.

■ File Name

The name of definition file of diagnose indices can be anyone, also anywhere is available for the location to create.

■ Creation

The template is stored as the file name 'definition \_ template' under the folder name 'set \_ diagnosisindicat.def' in the folder where the batch execution command (arqm command) is placed. The necessary settings for template file for reference are described as follows:

■ Format

The format is described as follows:

When multiple conditions are recorded, they will be judged as "OR condition".

Type key of indices value = condition 1(, condition 2, ... condition n)

The types of indices value shall be defined to any key as follows:

Type Key of Indices Value	Description
PASSABLE	Type that specifies the condition where there is a problem in part of the diagnose result.
NOGOOD	Specify the types of NOGOOD conditions in the diagnose result.

The contents not identical with PASSABLE and NOGOOD will be judged to be good.

The format of conditions is described as follows:

EvaluationKey Operator Value

The items of various conditions are described as follows:

Item	Description
------	-------------

EvaluationKey	The following evaluation keys can be recorded. INDICATION_COUNT :Indication Message Count (Excluding Suppressed Indication Message Count) MISRA_COUNT :MISRA Violation Count SEC_COUNT :IPA/SEC-C/C++ Violation Count INDICATION_K_COUNT :Indication Message Count Every 1ks (Excluding Suppressed Indication Message Count) DANGER_COUNT :Dangerous Field Alert Count MISRA_K_COUNT :MISRA Violation Count Every 1ks SEC_K_COUNT :IPA/SEC-C/C++ Violation Count Every 1ks MARKED_COUNT :Field for Attention Alert Count
Operator	The following operators can be recorded. ==, <, >, <=, >=
value	The comparison of evaluation key by operators shall be specified with decimal.

In addition, the evaluation shall be made based upon the conditions recorded in front of the file. Therefore, it is necessary to describe it in order of severity level.

■ Sample

The sample is described as follows:

```

NOGOOD=INDICATION_COUNT>=10,MISRA_COUNT>=10,DANGER_COUNT>=10,MARKED_COUNT>=10
; NOGOOD is judged in case [Indicated More than 10 Times], or [MISRA Violation More than 10], or [Dangerous Field Alert More than 10 Times], or [Field for Attention Alert Count More than 10 Times]

PASSABLE=INDICATION_COUNT>=5,MISRA_COUNT>=5,DANGER_COUNT>=5,MARKED_COUNT>=5
; PASSABLE is judged when NOGOOD conditions are not qualified, and [Remaining Indication Message More than 5], or [Remaining MISRA Violation More than 5], or [Dangerous Field Alter Count More than 5], or [Field for Attention Alert More than 5].

```

■ Default Value

If this file is not specified in the project configuration definition file, the file name 'qualityinfo' under the folder name 'diagnosisindicat.qdf' in the folder where the bulk execution command (arqm command) is stored is referred to.

The diagnose definition: the judgment of [NOGOOD] will be judged when message indicated more than 1, or dangerous field alert more than 1; while [PASSABLE] will be judged when field for attention alert more than 1.

#### 4.2.4. Indication feature definition file

Define the value of quantitative judgment for the number of indications calculated for each feature.

■ File Name

The file name and creation location are fixed and stored under the 'qualityinfo' folder in the

Agile+ Relief installation folder with the following file name:

File type	File name
List of indication feature definition file	pcsindex.ini
The quality feature has been defined Indication feature definition file	quality.pcs
The problem feature has been defined Indication feature definition file	problem.pcs
The syntax feature has been defined Indication feature definition file	grammar.pcs

■ Creation

The template is saved in the following folder in the [qualityinfo] folder under the install folder of Agile+ Relief.

Project type	Storage folder name
C/C++ language	characteristic_c
Java language	characteristic_java
Mixture of C/C++ and Java	characteristic_c_java

Use the templates and update the dangerous range (3 of description example) and cautious range (4 of description example) according to the description example.

■ Sample

Following is the description example of indication feature definition file with defined [Quality feature].

This is the example of updating the file [quality.pcs] in [qualityinfo] folder under the install folder of Agile+ Relief.

Update the value of dangerous range (3) and cautious range per Kstep (4).

[CODE]				
SJIS <-1.file's encoding				
[CHRACTERISTIC-NAME]				
; Name(english),		Name(japanese)		
----- -----				
Quality Characteristic,		Quality Characteristic <- 2.quality characteristic's class		
[CHRACTERISTIC-ID]				
;Characteristic-ID,	Item Name(english),	Item Name(japanese),	Danger Value,	Warning Value
----- ----- ----- ----- -----				
Reliability,	Reliability,	Reliability,	30,	20
Maintainability,	Maintainability,	Maintainability,	30,	20
Portability,	Portability,	Portability,	30,	20
Efficiency,	Efficiency,	Efficiency,	30,	20
Functionality,	Functionality,	Functionality,	30,	20
None,	None,	None,	-,	-
			3.Danger value	4. Cautious value

1. Indicate the file code.
2. Indicate the type of quality feature.
3. Indicate the dangerous range per Kstep.
4. Indicate the cautious range per Kstep.  
The item without warning is indicated by hyphen (-).

■ Default Value

Indication feature definition file needs to be used when calculating indication feature.

When calculating indication feature, quote the Indication feature definition file defined in [pcsindex.ini] in the [qualityinfo] folder under the install folder of Agile+ Relief.

### 4.3. Definition File for Output Content

About the description to output template file in HTML, please refer to "[3.5 Template File Update](#)"; about how to describe the definition file for output content, please refer to "[3.6 Description of Definition File for Output Content](#)".

## 5. How to use arqm command

### 5.1. Command Format

Arqm command is used for source program analysis and quality info output. The format of command is described as follows:

arqm [option] project structure definition file or project structure definition listing file
--

### 5.2. Option

The options of arqm command are described as follows:

Option	Description
-A	Specify only for source program analysis. When this option is specified, the quality info will not be output.
-R	Specify only for quality info output. When this option is specified, the source program analysis will not be executed.
-basepoint	Set the messages till the last time to "Before Managed" status. In case if this option is specified, then the quality info will not be output and the source program analysis will not be executed.
-continueanalyze	Specify for the analysis processing continues even if errors occur during the analysis. For the project in which errors occurs, the quality of the previous resolution results will be output.
-o Output Target Folder	Specify the output target folder for quality info result (HTML, CSV). When half-width space is found within the output folder name, please embrace it with double quote ("). when -A option is not specified, this option will be required.
-reporttype { <u>tree</u> } { flat }	Specify the display format of project structure to be output. tree : Display in tree format. flat : Display in non-layer format.
-V	Display the version of arqm command. When this option is specified, the settings of other options will be invalid.

### 5.3. Before Execution

- The HTML template files and output definition files (arqm.ini) must be stored in the same folder 'HTML\_TEMPLATE' as the arqm command.
- When Java language has been specified for the asset language of checked object in the project structure definition file, it is required to set the path of javac command in the environment variable PATH.

## 5.4. Project Analysis Result Information

The analysis results and processing records are saved in project analysis result information of the source program. The quality info results of HTML and CSV will be generated based on project analysis result information.

The project analysis result information is generated under the folder directory for saving project structure definition files. When moving the project structure definition file to another folder, please move the project analysis result information under the folder directory together, if the project analysis result information is not moved, the quality transition status will not be displayed.

## 5.5. Set the message before managed

To put previous messages in the "Before Managed" state, please execute the command, please execute the command according to the following.

```
arqm -basepoint project structure definition file
      or
      Project structure definition listing file
```

## 5.6. Return Value

The return values of arqm command are described as follows:

Return Value	Description
0	Normally Ended
1	License Error, such as failed to obtain license
2	Running environment error, command error
4	Definition file error
8	Error for source program analysis not completely finished.
16	HTML or CSV output error
99	Forcibly Terminated (Forcibly interrupt with Ctrl + C)

## 5.7. Error Message

### **ARQM\_001 Project Structure Definition File/Project Structure Definition Listing File is not specified.**

Project structure definition file or project structure definition listing file is not specified. Please specify the project structure definition file or project structure definition listing file.

### **ARQM\_002 Invalid option @1.**

Invalid option @1. Please specify the correct option.

### **ARQM\_003 Missing argument to @1.**

Missing argument to @1. Please specify the parameter.

### **ARQM\_004 Invalid argument. option = @1, argument = @2.**

Invalid argument. option = @1, argument = @2. Please specify the correct parameter.

### **ARQM\_005 Input file @1 is not the Project Structure Definition File/Project Structure Definition Listing File.**

The @1 file for project structure definition file or project structure definition listing file is with the extension except for [.pgs] or [.pgl]. Please specify the correct file name.

**ARQM\_006 Cannot open the Project Structure Definition File/Project Structure Definition Listing File @1.**

Cannot open the Project Structure Definition File/Project Structure Definition Listing File @1. Please confirm the file name is correct and the access to the file.

**ARQM\_007 Options @1 and @2 cannot be specified concurrently.**

Options @1 and @2 cannot be specified concurrently. Please specify either one.

**ARQM\_008 Required option -o is not specified.**

Required option -o is not specified. Please specify the -o option.

**ARQM\_009 Options must precede Project Structure Definition File /Project Structure Definition Listing File.**

Options must precede Project Structure Definition File /Project Structure Definition Listing File. Please specify the option before project structure definition file or project structure definition listing file.

**ARQM\_010 Out of memory.**

Out of memory. Please check the system for error. And then re-execute with other processes have been ceased.

**ARQM\_011 License certification failed.@ 1**

Agile+ Relief license certification failed. @1 is the error ID of license management system. Please refer to install guide of Agile+ Relief.

**ARQM\_012 "MISRA Option" license certification failed.@ 1**

"MISRA Option" license certification failed. @1 is the error ID of license management system. Please refer to install guide of Agile+ Relief.

**ARQM\_013 "Agile+ Relief" license certification failed.@ 1**

License certification of Agile+ Relief failed. @1 is the error ID of license management system. Please refer to install guide of Agile+ Relief.

**ARQM\_014 License management module load failed. Please install Agile+ Relief C/C++, Agile+ Relief J corresponding to "Agile+ Relief".**

Agile+ Relief license certification failed. Please check the installed Agile+ Relief C/C++ or Agile+ Relief J, and then install the corresponding Agile+ Relief C/C++ or Agile+ Relief J with Agile+ Relief.

**ARQM\_015 Option @1 cannot be specified concurrently with other options.**

Option @1 cannot be specified concurrently with other options. Please specify @1 only when specifying the @1 option.

**ARQM\_050 Cannot open log file @1.**

Cannot open log file @1. Please be sure there are enough disk space and the access for log file @1.

**ARQM\_051 Cannot open file @1.**

Cannot open file @1. Please check if the file @1 exists and check its access limit.

**ARQM\_052 Unable to create folder @1.**

Unable to create folder @1. Please check there is enough disk space and the write access for folder @1.

**ARQM\_053 Unable to create file @1.**

Unable to create file @1. Please check there is enough disk space and the write access for file @1.

**ARQM\_054 Failed to read file @1.**

Failed to read file @1. Please check the access for file @1.

**ARQM\_055 Agile+ Relief C/C++, Agile+ Relief J is not installed. Please install Agile+ Relief C/C++, Agile+ Relief J corresponding to "Agile+ Relief".**

Agile+ Relief C/C++ or Agile+ Relief J not installed. For the use of Agile+ Relief, please installed the required Agile+ Relief C/C++ or Agile+ Relief J.

**ARQM\_056 The file @1 does not exist. Please install Agile+ Relief C/C++, Agile+ Relief J corresponding to "Agile+ Relief".**

The file @1 does not exist. Please check for the installed Agile+ Relief C/C++ or Agile+ Relief J, and then install the corresponding Agile+ Relief C/C++ or Agile+ Relief J to Agile+ Relief.

**ARQM\_057 Unable to delete file @1.**

Unable to delete file @1. Please check if the file @1 exists and check its access limit.

**ARQM\_058 Unable to delete folder @1.**

Unable to delete folder @1. Please check if the folder @1 exists and check its access limit.

**ARQM\_059 Unable to access folder @1.**

Unable to access folder @1. Please check if the folder @1 exists and check its access limit.

**ARQM\_100 Invalid section @1.**

Invalid section @1. Please specify with a correct section.

**ARQM\_101 Invalid value. section = @1, value = @2**

Incorrect value @2 is specified in @1 section. Please specify with a correct value.

**ARQM\_102 Inexistent folder. section = @1, key = @2, folder = @3**

Folder @3 specified by @2 key in section @1 does not exist. Please specify the correct folder name.

**ARQM\_103 Inexistent file. section = @1, key = @2, file = @3**

File @3 specified by @2 key in section @1 does not exist. Please specify the correct file name.

**ARQM\_104 Invalid value. section = @1, key = @2, value = @3**

Unsuitable value @3 is specified by @1 for @2 key within section. Please specify with a correct value.

**ARQM\_105 Invalid key in the section. section = @1, key = @2**

Unable to specify @2 key for @1 section. Please specify the correct key.

**ARQM\_106 Key is not specified. section = @1, key = @2**

@2 key is not specified for @1 section. Please specify @2 key for @1 section.

**ARQM\_107 Invalid file. section = [PROJECTLIST], key = "PROJECTFILE", file = @1**

Incorrect file @1 was specified by PROJECTFILE key of [PROJECTLIST] section with the extension other than [.pgs] and [.pgl]. Please specify the correct file name.

**ARQM\_109 Invalid key. section = [OPERATEPROJECT], PROJECTTARGET key = @1.**

No valid key is specified for [OPERATEPROJECT] section. Please record valid key in [OPERATEPROJECT] section or delete [OPERATEPROJECT] section.

**ARQM\_110 The value "." is specified for "BINDPROJECT" key of [SOURCES] section. Other specifications for BINDPROJECT key will be invalid.**

"." is specified for BINDPROJECT key of section, while other BINDPROJECT keys are



specified. When "." is specified for BINDPROJECT key, please do not specify the other BINDPROJECT keys.

**ARQM\_111 The value @1 is specified for "BINDPROJECT" key of [SOURCES] section. The value @2 is invalid.**

@2 folder beneath the folder specified by BINDPROJECT key @1 is specified for BINDPROJECT key. The designation of @2 folder is redundant, please delete.

**ARQM\_112 The value @1 is specified for "BINDPROJECT" key of [SOURCES] section. The value @2 specified for "EXCEPTDIR" key will be invalid.**

Folder @1 for BINDPROJECT key and folder @2 for EXCEPTDIR key are in the relation of include. Please delete either one.

**ARQM\_113 Required key is not specified. section = @1, key = @2**

@2 key is not specified for @1 section. For @2 key is required, please specify.

**ARQM\_114 Required folder was not found. section = @1, key = @2**

Folder specified for @2 key of @1 section does not exist. Please specify the existed folder.

**ARQM\_115 Required file was not found. section = @1, key = @2**

File specified for @2 key of @1 section does not exist. Please specify the existed file.

**ARQM\_116 The @1 section cannot be specified in the Project Structure Definition Listing File.**

The @1 section cannot be specified in the Project Structure Definition Listing File. Please specify with a correct section.

**ARQM\_117 The @1 section cannot be specified in the Project Structure Definition File.**

The @1 section cannot be specified in the Project Structure Definition File. Please specify with a correct section.

**ARQM\_118 The same file @1 is specified in the Project Structure Definition Listing File in an upper level.**

The same file @1 is specified in the Project Structure Definition Listing File in an upper level. Please delete the same designation of project structure definition listing file.

**ARQM\_119 The keys cannot be specified concurrently. section = @1, key1 = @2, key2 = @3**

@2 and @3 cannot be specified for @1 section at the same time. Please delete either one.

**ARQM\_120 The "SPOTDATE" key in [REPORTOPTION] section cannot be specified with above 14 days.**

The "SPOTDATE" key in [REPORTOPTION] section cannot be specified with above 14 days. Please specify within 14 days.

**ARQM\_121 The folder does not exist in the folder specified by the "TARGET" key. section = @1, key = @2, folder = @3, value of "TARGET" key = @4**

Folder @3 specified by @2 key of @1 section is not found in folder @4 by TARGET key. Please specify the correct folder name under folder @4 by TARGET key.

**ARQM\_122 The file does not exist in the folder specified by the "TARGET" key. section = @1, key = @2, file = @3, value of "TARGET" key = @4**

File @3 specified by @2 key of @1 section is not found in folder @4 by TARGET key. Please specify the correct file name under folder @4 by TARGET key.

**ARQM\_123 The value of "PROJECTLEVEL" key in [SOURCES] section is specified.**

**The value "." specified for "BINDPROJECT" key will be invalid.**

Both PROJECTLEVEL key and BINDPROJECT key are specified for [SOURCES] section. Please specify either one.

**ARQM\_124 The path specified for "BINDPROJECT" key in [SOURCES] section exceeds @1 levels specified by "PROJECTLEVEL" key.**

Path, which surpasses the layer count @1 by PROJECTLEVEL key of [SOURCES] section is specified by BINDPROJECT key. Please change the path layer by BINDPROJECT key, or the layers of PROJECTLEVEL.

**ARQM\_125 The length of the value exceeds 256 bytes. section = @1, key = @2**

The character length of value by @2 key of @1 section surpasses 256 bytes. Please specify the value of @2 key of @1 section within 256 bytes.

**ARQM\_126 The value is out of the range. section = @1, key = @2, range of value = @3 - @4**

The value of @2 key of @1 section surpasses the range of @3~@4 Is specified. Please specify the value of @2 key of @1 section within the range of @3~@4.

**ARQM\_127 Only one Coding Guideline can be specified. section = @1, MISRA-CVERSION = @2, SEC-CVERSION = @3**

Both MISRA violation check (MISRA-CVERSION) and IPA/SEC-C/C++ violation check (SEC-CVERSION) are specified in @1 section. For two of them cannot be specified at the same time, please delete or disable (=0) either one.

**ARQM\_128 Required key is not specified. (When "TOOLTYPE = @1" in [VERSIONSYSTEM\_SOURCES] section is specified, key @2 is required)**

In [VERSIONSYSTEM\_SOURCES] section, the @2 KEY is not specified. When @1 is specified for the TOOLTYPE KEY of [VERSIONSYSTEM\_SOURCES] section, the @2 KEY is mandatory, so please specify it.

**ARQM\_129 Invalid key. (When "TOOLTYPE = @1" in [VERSIONSYSTEM\_SOURCES] section is specified, key @2 is invalid)**

When @1 is specified for the TOOLTYPE KEY of the [VERSIONSYSTEM\_SOURCES] section, the @2 KEY cannot be specified.

**ARQM\_200 The file to be checked does not exist in the path @1 specified by "TARGET" in [SOURCE] section.**

Objective source file for check is not found under the folder by TARGET key of [SOURCES] section of project structure definition file. Please specify the folder with source files.

**ARQM\_201 Cannot find the project specified by "PROJECTTARGET" key. file = @1, section = [OPERATEPROJECT], key = "PROJECTTARGET", value of key = @2**

Cannot find the project specified by "PROJECTTARGET" key. file = @1, section = [OPERATEPROJECT], key = "PROJECTTARGET", value of key = @2. Please specify the path containing the objective source files for check and can be recognized as project.

**ARQM\_202 The file specified by "FILEOPTION/APPENDFILEOPTION" key is not the source file to be checked in the project. file = @1, section = [OPERATEPROJECT], key = "FILEOPTION" or "APPENDFILEOPTION", value of key = @2**

The file specified by "FILEOPTION" or "APPENDFILEOPTION" key is not the source file to be checked in the project. file = @1, section = [OPERATEPROJECT], key = "FILEOPTION" or "APPENDFILEOPTION", value of key = @2. Please specify the correct file name.

**ARQM\_203 The file specified by "FILEOPTION/APPENDFILEOPTION" key is invalid.**

**Please specify another path except for the path specified in "TARGET" key. file = @1, section = [OPERATEPROJECT], key = "FILEOPTION" or "APPENDFILEOPTION", value of key =@2.**

The file specified by "FILEOPTION" key is invalid. Please specify another path except for the path specified in "TARGET" key. file = @1, section = [OPERATEPROJECT], key = "FILEOPTION" or "APPENDFILEOPTION", value of key =@2.

**ARQM\_204 Check the setup of Coding Guideline. Different Coding Guidelines are specified in [OPERATECOMMONPROJECT] section and [OPERATEPROJECT] section. file = @1**

[OPERATECOMMONPROJECT] section and [OPERATEPROJECT] section of specified file @1 are different coding manners. If different coding manner check from that of [OPERATECOMMONPROJECT] section is specified in [OPERATEPROJECT] section, please disable(=0) the settings of [OPERATEPROJECT] section for the coding manner check of [OPERATECOMMONPROJECT] section.

**ARQM\_250 Checkout process encountered an error.**

Checkout process encountered an error. Please act according to the checkout error that occurs and execute again.

**ARQM\_300 Analysis process encountered an error.**

Analysis process encountered an error. Please deal with the analysis error, and then re-execute.

**ARQM\_401 An inappropriate keyword is specified in HTML template file @1.**

An inappropriate keyword is specified in HTML template file @1. Please delete the invalid keyword.

**ARQM\_402 Analysis has not been done. Information to be displayed does not exist.**

The page without project analysis information is output during HTML output. Please execute the analysis.

**ARQM\_501 The description of @1 section is incorrect. file = @2**

The description of @1 section of the file @2 is incorrect. Please specify with a correct format.

**ARQM\_502 Invalid section @1. file = @2**

An incorrect @1 section has been specified for file @2. Please specify with a correct section.

**ARQM\_503 Invalid value. section = @1, value = @2, file = @3**

An incorrect value @2 has been specified for @1 section of file @3. Please specify with a correct value.

**ARQM\_810 Definition file of diagnose indices @1 error.**

Definition file of diagnose indices @1 error. Please correct the error.

**ARQM\_811 Cannot recognize character @1.**

Cannot recognize character @1. Please correct the error.

**ARQM\_812 Format of indices value error.**

Format of indices value error. Please correct the error.

**ARQM\_813 Indices value error. type key = @1.**

Error occurred for type key @1 of diagnose indices value settings. Please correct the error.

**ARQM\_814 Indices value error. type key = @1, operator = @2.**

Error occurred for type key @1 operator @2 of diagnose indices value settings. Please

correct the error.

**ARQM\_815 Indices value error. type key = @1, evaluation key = @2.**

Error occurred for type key @1 evaluation key @2 of diagnose indices value settings. Please correct the error.

**ARQM\_816 Indices value is set again. type key = @1, evaluation key = @2**

Settings of type key @1 evaluation key @2 of diagnose indices value has been set repeatedly. Please delete either one.

**ARQM\_817 Indices value error. type key = @1, evaluation key = @2, operator = @3**

Error occurred for type key @1 evaluation key @2 of diagnose indices value settings. Please correct the error.

**ARQM\_818 Indices value error. type key = @1, evaluation key = @2, value = @3**

Error occurred for type key @1 evaluation key @2 value @3 of diagnose indices value settings. Please correct the error.

**ARQM\_819 Indices value error. type key = @1, error element = @2**

Error occurred for type key @1 of diagnose indices value settings. Please correct the error.

**ARQM\_820 Setting/format of indices value error.**

Error occurred for the settings or format of diagnose indices value. Please correct the error.

**ARQM\_821 The "type key" of indices value is set again. type key = @1**

Settings of type key @1 of diagnose indices value has been set repeatedly. Please delete either one.

**ARQM\_822 The data of indices value does not exist.**

The data of indices value does not exist. Please correct the error.

**ARQM\_831 Invalid key specified in the section of definition file for output content. section = @1, key = @2**

Error occurred to the @1 section of definition file for output contents. Please correct the error.

**ARQM\_832 Invalid time format in definition file for output content. @1**

Error occurred to the time format of definition file for output contents. Please correct the error.

**ARQM\_833 Invalid value in the section of definition file for output content. section = @1, value = @2**

Settings that cannot be processed are found in @1 section of definition file for output contents. Please correct the error.

**ARQM\_835 Can not read definition file for output content."@1"**

Can not read definition file @1 for output content. Please check the file contents and the access for the file.

**ARQM\_837 Invalid key in definition file for output content."@1"**

Key @1 that cannot be processed is found in definition file for output contents. Please correct the error.

**ARQM\_838 Format error in definition file for output content."@1"**

Error occurred to the definition file for output contents. Please correct the error.

**ARQM\_856 Unable to create an analysis result file."@1"**

Unable to create an analysis result file. @1. Please check for the disk space of project structure definition file or project structure definition listing file, and the access for the file.

**ARQM\_860 Failed to read the definition file of "dangerous field/field for attention" @1.**  
Failed to read the definition file of "dangerous field/field for attention" @1. Please check the access for the definition file of dangerous field / field for attention.

**ARQM\_861 The dangerous field is not specified.**  
The dangerous field is not specified. Please check the file contents.

**ARQM\_863 Invalid columns of dangerous field/field for attention.**  
Invalid columns of dangerous field/field for attention. Please correct the error.

**ARQM\_864 Invalid field specified.**  
Invalid field specified. Please do not use fixed value.

**ARQM\_865 The string can not be identified to be a BOOL value is specified in dangerous field/field for attention. @1**  
String other than BOOL value is specified for dangerous field/field for attention. Please specify to be TRUE of FALSE.

**ARQM\_866 Please set the value of field for attention less than dangerous field. field for attention = @1, dangerous field = @2**  
A value greater than that of dangerous field is specified for the field for attention. The value for field for attention shall be set less than that of dangerous field.

**ARQM\_867 Duplicate data is specified in the dangerous field/field for attention. row = @1**  
Row @1 is specified by multiple lines in the definition file settings of dangerous field/field for attention. Please delete either one.

## 6. To Agile+ Relief C/C++ MISRA Option User

### 6.1. Replacement of Template File

The files to be replaced, no matter MISR check will be performed or not, are saved in the same folder [HTML\_TEMPLATE\_MISRA / HTML\_TEMPLATE\_MISRAPLUS] and [HTML\_TEMPLATE\_NORMAL] with that of batch execution command.

Link to download the CSV output result of MISRA violation list is included in the template file (file name: project\_detail.htm) for MISRA.

MISRA violation count, MISRA violation count every 1ks and the suppressed MISRA violation count are included in the items displayed in quality info list within the definition file (file name: arqm.ini) for output contents for MISRA.

- When outputting the result of MISRA-C check

Replace the file in the folder [HTML\_TEMPLATE\_MISRA] with the file in the folder [HTML\_TEMPLATE] that is the same as the one of batch execution command.

- When outputting the result of MISRA-C++ check

Replace the file in the folder [HTML\_TEMPLATE\_MISRAPLUS] with the file in the folder [HTML\_TEMPLATE] that is the same as the one of batch execution command.

Furthermore, if the project that does not perform MISRA check is found in the project structure definition file or project structure definition listing file, the download link target file of CSV output result for MISRA violation list will not be generated. In addition, the MISRA violation count, MISRA violation count every 1ks, and the suppressed MISRA violation count will be displayed to be 0.

## 6.2. Description of Project Structure Definition File

MISRA violation check is recorded by [MISRA-CVERSION] key in the project structure definition file.

In addition, the analysis using the identifier file for MISRA are recorded by [-F] option of [OPTION] key in the project structure definition file. The identifier file for MISRA is saved in [Analyze¥EPOM¥MisraInfo] of install folder in Agile+ Relief C/C++.

For more details of the project structure definition file, please refer to "[4.1.1 Project Structure Definition File](#)".

The sample is described as follows:

```
-----  
; Set the objective asset for check  
[SOURCES]  
TARGET          = C:¥SamplePrj¥src¥  
CODE             = SJIS  
CPLUSPLUSEXT    = .cpp  
CPLUSPLUSEXT    = .CC  
BINDPROJECT     = common  
EXCEPTDIR     = common¥worksrc  
-----  
; Set project common action  
[OPERATECOMMONPROJECT]  
OPTION          = -lc:¥include -DDEBUG  
-F"C:¥Program  
Files¥AgilePlus¥PGRelief¥Analyze¥EPOM¥MisraInfo¥MISRA_C_V1.idt"  
RULEFILE        = C:¥arsetting¥comrule.txt  
RULEFILE        = C:¥arsetting¥grprule.txt  
METRICSFILE     = C:¥arsetting¥metrics.txt  
INDICATORFILE   = C:¥arsetting¥shihyo.txt  
GROUPFILE       = C:¥arsetting¥group.txt  
MISRA-CVERSION = 1  
-----  
; Settings during report output  
[REPORTOPTION]  
RANGE = 4,W
```

## 7. To IPA/SEC-C/C++ Coding Manner Check User

### 7.1. Replacement of Template File

The files to be replaced, no matter IPA/SEC-C/C++ check will be performed or not, are saved in the same folder [HTML\_TEMPLATE\_SEC] and [HTML\_TEMPLATE\_NORMAL] with that of batch execution command.

Link to download the CSV output result of IPA/SEC-C/C++ violation list is included in the template file (file name: project\_detail.htm) for IPA/SEC-C/C++.

IPA/SEC-C/C++ violation count, IPA/SEC-C/C++ violation count every 1ks and the suppressed IPA/SEC-C/C++ violation count are included in the items displayed in quality info list within the definition file (file name: arqm.ini) for output contents for IPA/SEC-C/C++.

Above mentioned files are replaced with those saved in the same folder [HTML\_TEMPLATE] of batch execution command when output the IPA/SEC-C/C++ check result.

Furthermore, if the project that does not perform IPA/SEC-C/C++ check is found in the project structure definition file or project structure definition listing file, the download link target file of CSV output result for IPA/SEC-C/C++ violation list will not be generated. In addition, the IPA/SEC-C/C++ violation count, IPA/SEC-C/C++ violation count every 1ks, and the suppressed IPA/SEC-C/C++ violation count will be displayed to be 0.



## 7.2. Description of Project Structure Definition File

IPA/SEC-C/C++ violation check is recorded by [SEC-CVERSION] key in the project structure definition file.

In addition, the analysis using the identifier file for IPA/SEC-C/C++ are recorded by [-F] option of [OPTION] key in the project structure definition file. The identifier file for IPA/SEC-C/C++ is saved in [Analyze¥EPOM¥SecInfo] of install folder in Agile+ Relief C/C++.

For more details of the project structure definition file, please refer to "[4.1.1 Project Structure Definition File](#)".

The sample is described as follows:

```
-----  
; Set the objective asset for check  
[SOURCES]  
TARGET          = C:¥SamplePrj¥src¥  
CODE            = SJIS  
CPLUSPLUSEXT   = .cpp  
CPLUSPLUSEXT   = .CC  
BINDPROJECT    = common  
EXCEPTDIR    = common¥worksrc  
-----  
; Set project common action  
[OPERATECOMMONPROJECT]  
OPTION          = -lc:¥include -DDEBUG  
-F"C:¥Program  
Files¥AgilePlus¥PGRelief¥Analyze¥EPOM¥SecInfo¥SEC C V1.idt"  
RULEFILE       = C:¥arsetting¥comrule.txt  
RULEFILE       = C:¥arsetting¥grprule.txt  
METRICSFILE    = C:¥arsetting¥metrics.txt  
INDICATORFILE  = C:¥arsetting¥shihyo.txt  
GROUPFILE      = C:¥arsetting¥group.txt  
SEC-CVERSION = 1  
-----  
; Settings during report output  
[REPORTOPTION]  
RANGE = 4 W
```

## 8. How to Use the Indication Message Viewer

The following explains how to use the indication message viewer. For the details of each operation, please refer to the HELP.

### 8.1. Download reference files of project analysis results

Confirm that the corresponding reference files of project analysis results of the project of the message can be downloaded from "Project Quality Info" of the output HTML file.

When the saving destination of project structure definition file is changed, please download again.

The screenshot displays the Agile+ Relief Analysis R application interface. On the left, the 'Project Structure' sidebar shows a tree view for '[SampleGr1\_MISRA]' containing sub-projects '[Project1]', '[Project2]', and '[Project3]'. The main area contains two summary tables and a download section.

**Project Summary Table:**

File Name	Executed Step Size	Number of all messages	Style	Types	Declarations and definitions	Expressions	Statements	Preprocessing directives	Environment/others	Encapsulation	Override	Overload	Inheritance	Exception handling
Project1 sample2.c	26	3	3	0	0	0	0	0	0	0	0	0	0	0
Project1 sample3.c	56	7	7	0	0	0	0	0	0	0	0	0	0	0
Project1 sample4.c	83	4	4	0	0	0	0	0	0	0	0	0	0	0
<b>total</b>	<b>213</b>	<b>35</b>	<b>28</b>	<b>4</b>	<b>0</b>	<b>4</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Grammar characteristic Table:**

File Name	Executed Step Size	Number of all messages	Style	Types	Declarations and definitions	Expressions	Statements	Preprocessing directives	Environment/others	Encapsulation	Override	Overload	Inheritance	Exception handling
Project1 sample.h	10	0	0	0	0	0	0	0	0	0	0	0	0	0
Project1 sample1.c	38	21	4	0	1	11	2	0	0	0	0	0	0	0
Project1 sample2.c	26	3	0	0	0	2	1	0	0	0	0	0	0	0
Project1 sample3.c	56	7	0	0	0	6	1	0	0	0	0	0	0	0
Project1 sample4.c	83	4	0	0	0	0	4	0	0	0	0	0	0	0
<b>total</b>	<b>213</b>	<b>35</b>	<b>4</b>	<b>0</b>	<b>1</b>	<b>19</b>	<b>8</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Problem characteristic Table:**

File Name	Executed Step Size	Number of all messages	Initialization gap	Incorrect Memory operation	Simple mistake	Lack of information	Performance degradation	Porting/environment problem	Security vulnerability	Low maintainability	Resource release miss	Resource wasting	Exception handling miss
Project1 sample.h	10	0	0	0	0	0	0	0	0	0	0	0	0
Project1 sample1.c	38	21	1	4	8	1	0	0	0	4	0	0	0
Project1 sample2.c	26	3	0	0	2	1	0	0	0	0	0	0	0
Project1 sample3.c	56	7	0	0	0	0	0	0	0	0	0	0	0
Project1 sample4.c	83	4	0	0	0	0	0	0	0	0	0	0	0
<b>total</b>	<b>213</b>	<b>35</b>	<b>1</b>	<b>4</b>	<b>10</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>4</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Download section:**

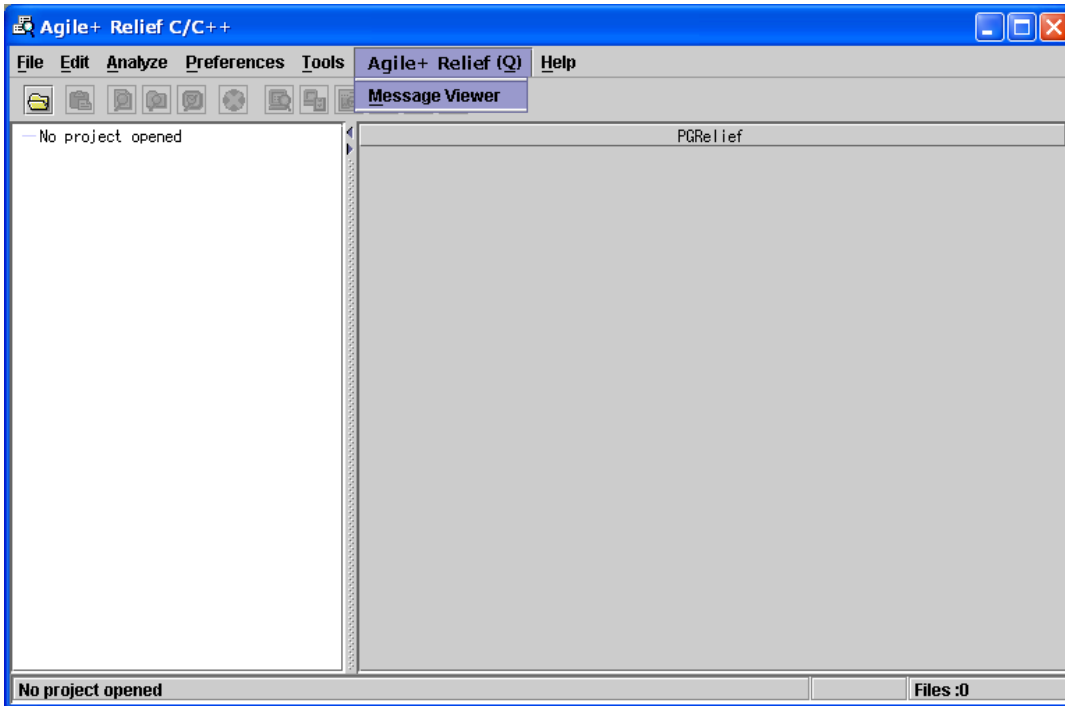
- [Project analysis result reference file\(pqqr\)](#)
- [MISRA-C violation list file\(csv format\)](#)
- [Message list file\(csv format\)](#)
- [Details of quality information\(csv format\)](#)
- [Change information of quality\(csv format\)](#)
- [Metrics information\(csv format\)](#)
- [Indication feature distribution \(Number of indications per 1KStep\) list \(csv format\)](#)
- [Indication feature distribution \(Number of indications\) list \(csv format\)](#)

A yellow callout box with the text "Download the project analysis result reference file." points to the first link in the 'Download' section.

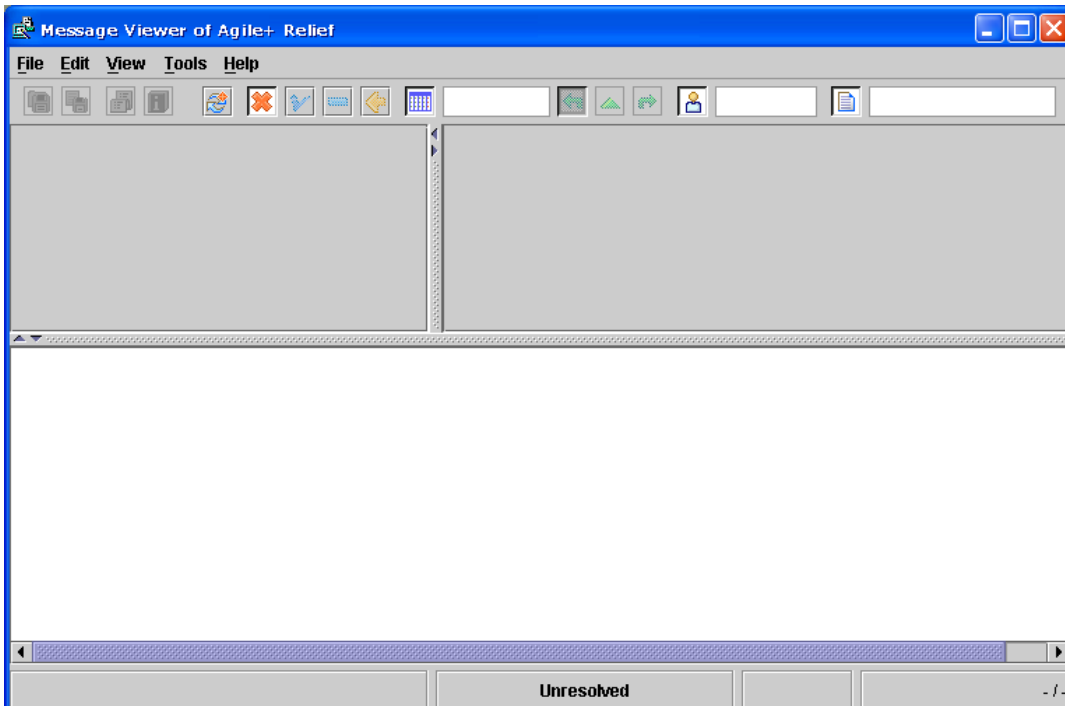
## 8.2. Startup

Select [Message Viewer] from the [Agile+ Relief] menu of the [main window].

When the installation of indication message viewer finished, the [Agile+ Relief] menu will be displayed.

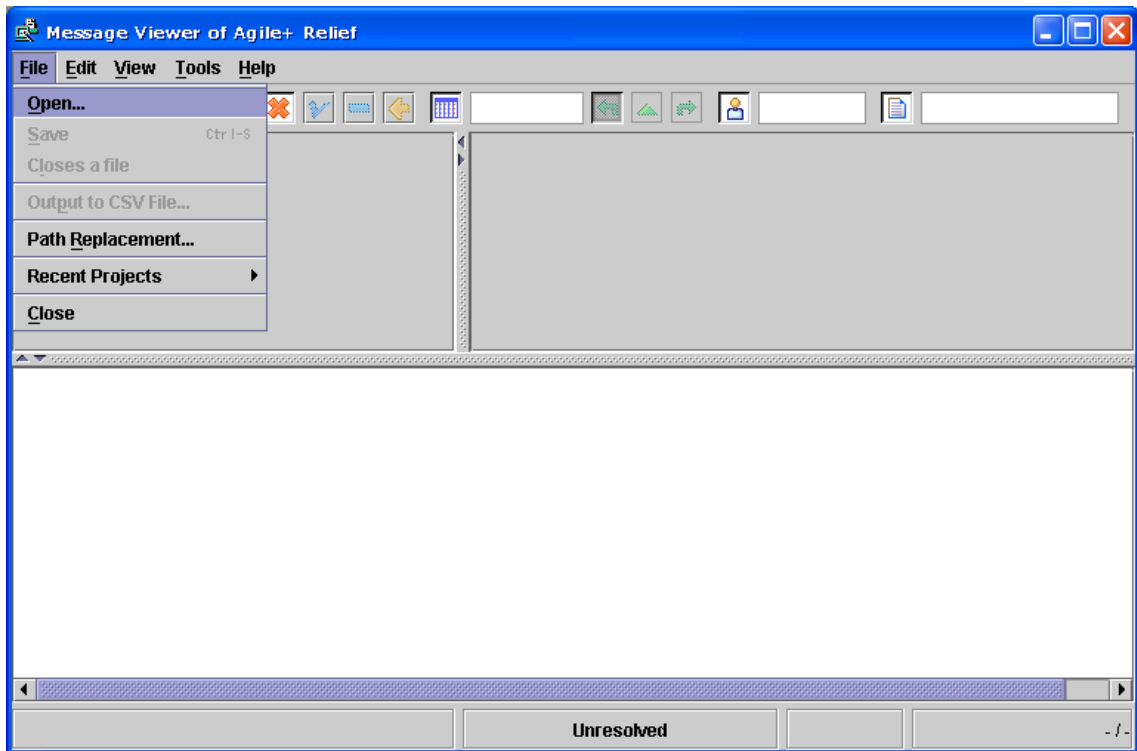


Start the Indication message viewer.

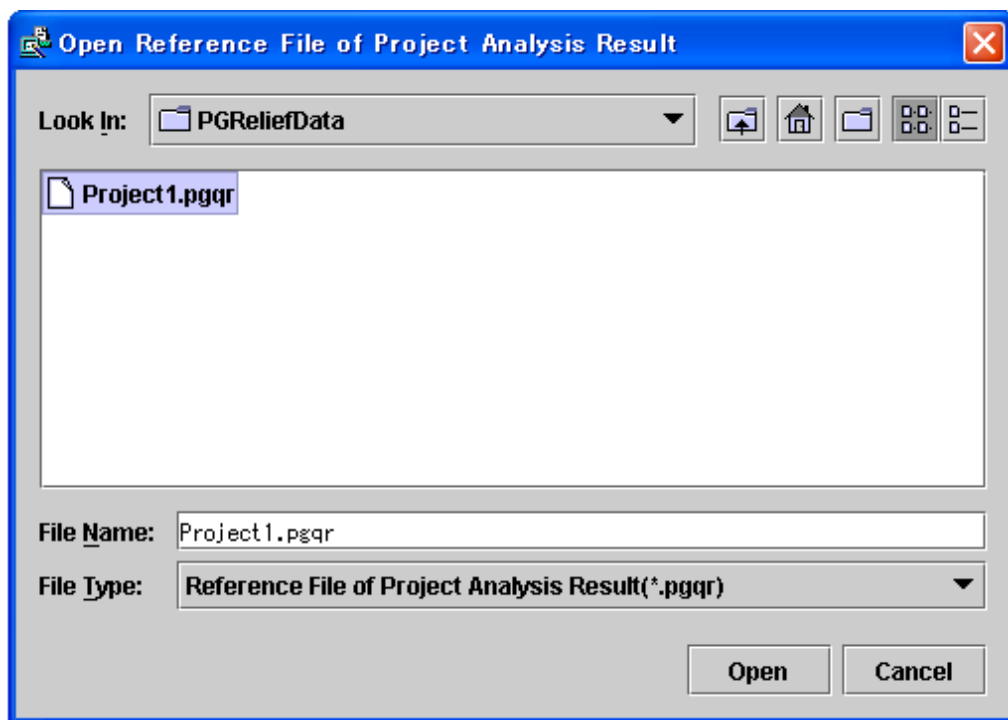


## 8.3. Open the project analysis result reference file

Select [Open...] of [File] menu.



Display the [Open Reference File of Project Analysis Result] dialog box.



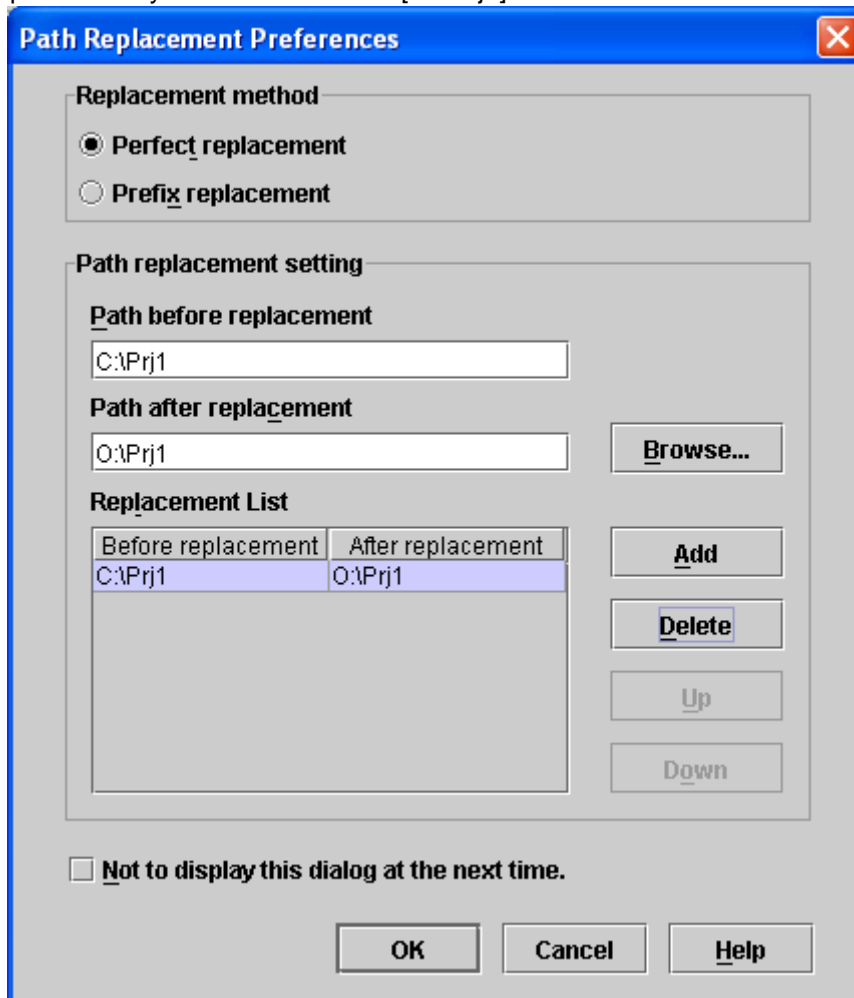
Select the project analysis result reference file that has just been downloaded and select [Open].

Then the [Path Replacement Preferences] dialog box will be displayed. However, for the environment where the paths of the client and server are the same, the setting of this dialog box

is not required.

The environment of the server that has executed batch execution command (arqm command) is recorded in the path saved in the project analysis result information. For the environment with different paths for client and server, if the project analysis result information is referenced in the indication message viewer of client, then the indication message or source program will not be displayed. In this case, set the correspondence for the paths of server and client in the [Path Replacement Preferences] dialog box.

The following sample is the situation when the path of analysis result of server is [C:¥Prj1], and the path of analysis result of client is [O:¥Prj1].

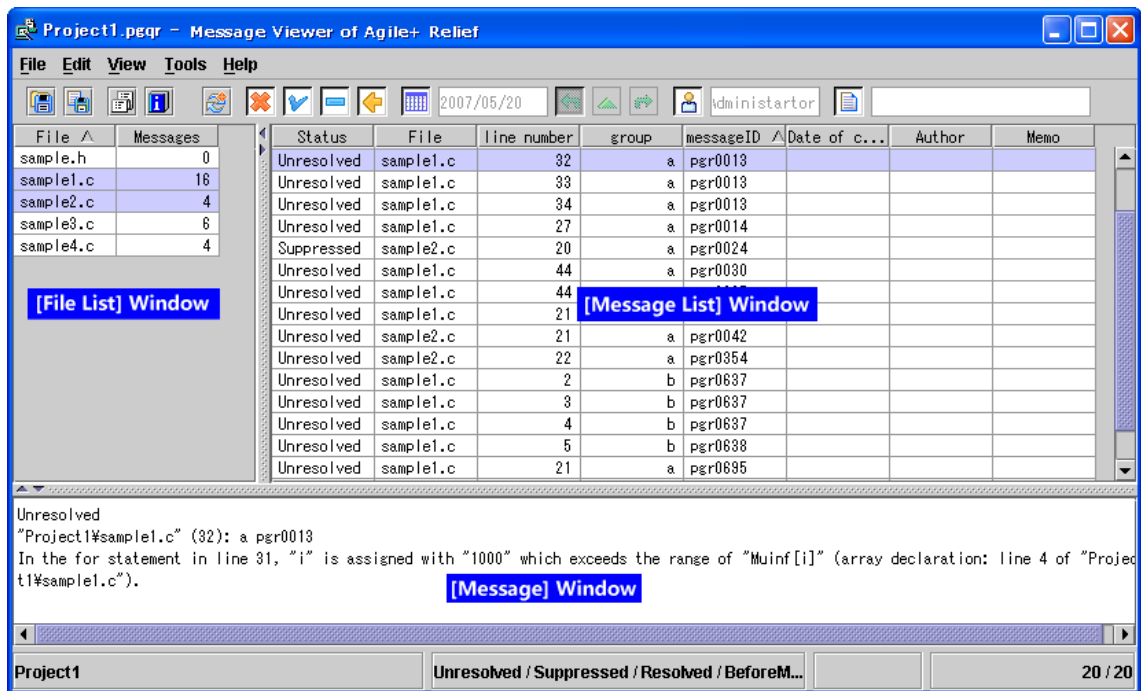


Select [OK] to finish the setting.

## 8.4. Confirm Messages

Confirm the detected messages.

First of all, it describes the structure of the window.



- The list of analysis objects files is displayed in the [File List] window.
- The detected messages of selected file in the file list are displayed in the [Message List] window.
- The content of messages selected from the message list is displayed in the [Message] window.

The following explains how to set the display conditions.

Display the messages in the [Message List] window according to the set display conditions. When setting the display conditions, select the [Message Filter...] menu of [View]. Display conditions can also be set through the toolbar.

The image shows a dialog box titled "Message Filter Preferences" with a blue title bar and a close button (X) in the top right corner. The dialog is divided into three main sections: "Status", "Date of confirmation", and "Note".

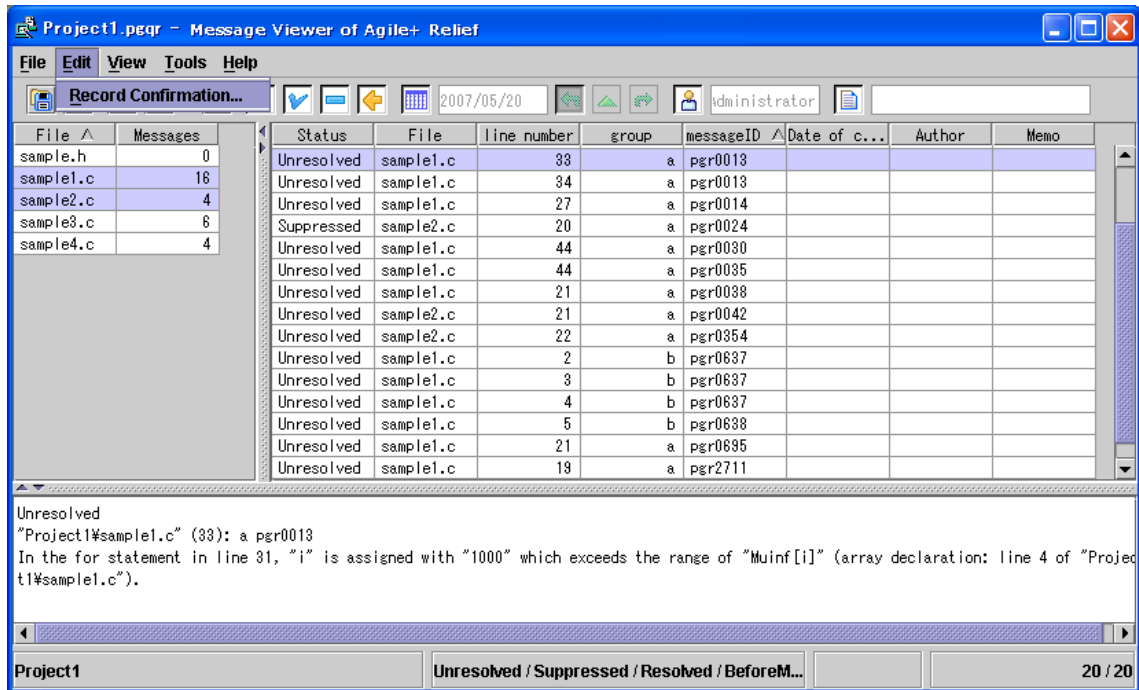
- Status:** Contains four checked checkboxes:  **U**nresolved,  **S**uppressed,  **R**esolved, and  **B**eforeManaged.
- Date of confirmation:** Contains a checked checkbox  **D**ate of confirmation, a text input field with the value "2007/05/31", and a "Today" button. To the right is an "Option" group with three radio buttons:  **E**arlier than,  **O**n the Day, and  **L**ater than.
- Note:** Contains a checked checkbox  **A**uthor with a text input field containing "Fujitsu Tarou", and an unchecked checkbox  **M**emo with an empty text input field.

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

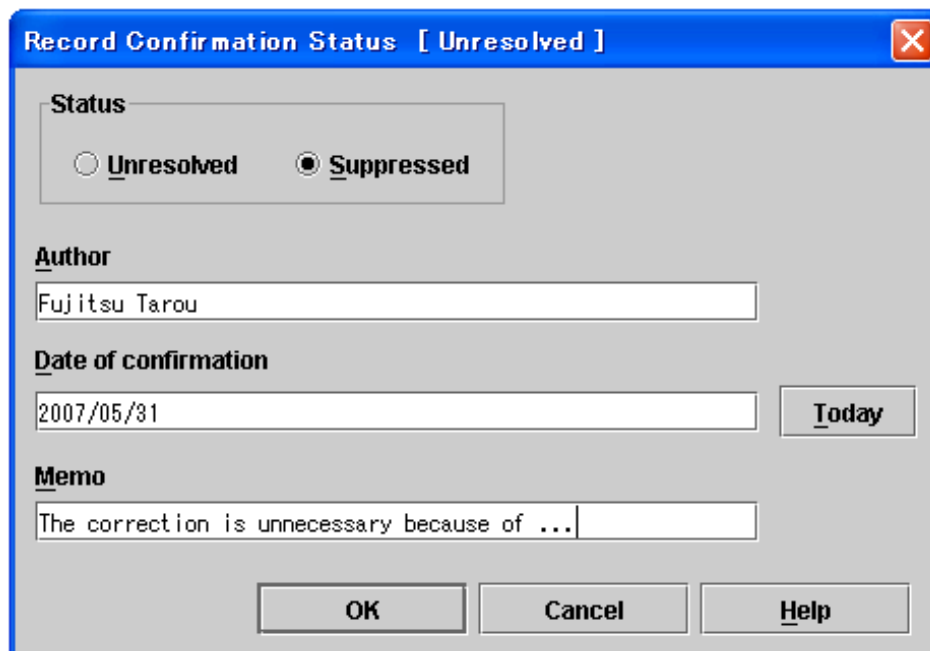
The following explains how to set processing records.

Select the message whose processing needs to be recorded from the message list. The messages of which the processing can be recorded are "Unresolved" and "Suppressed".

Select the message and select the [Record Confirmation...] menu of [Edit]. It can also be operated through the pop-up menu.

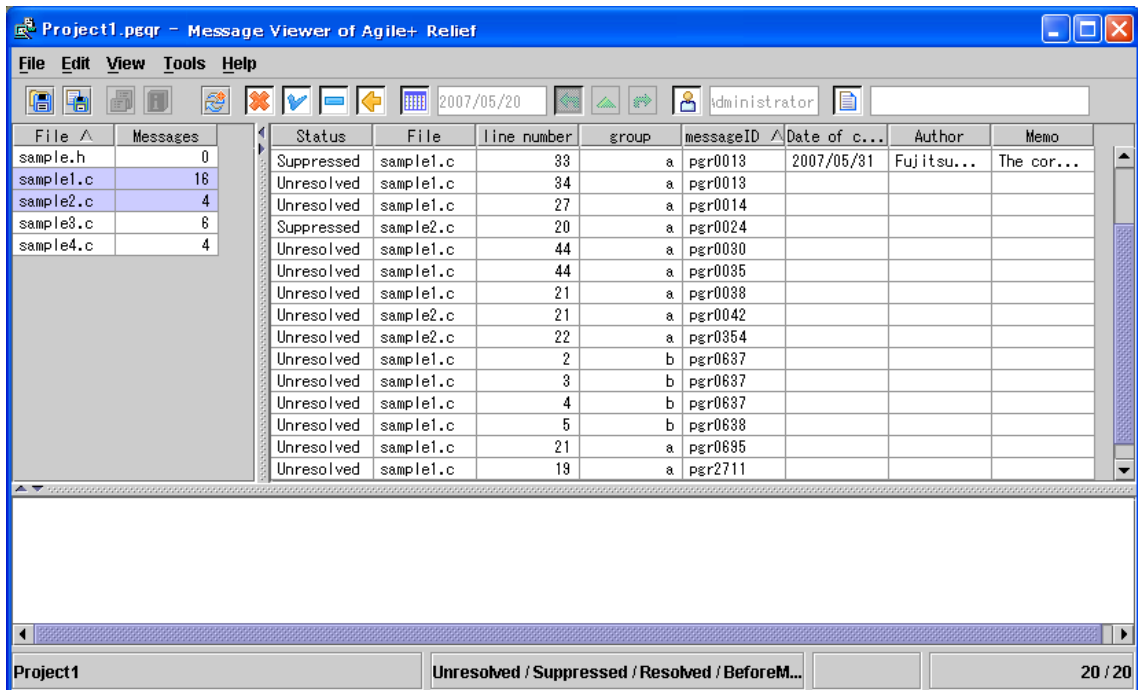


The [Record Confirmation Status] dialog box is displayed. Please discuss about the necessity of processing and input the processing record.

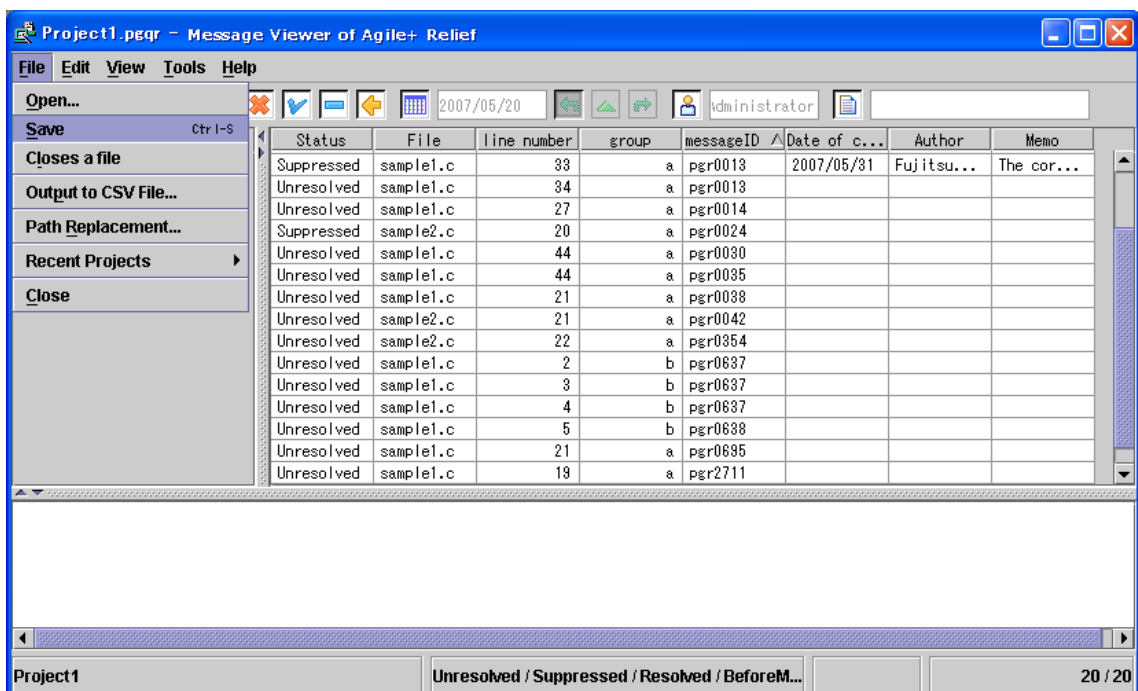




The records of processing will be reflected in the [Message List] window.



Save the input processing record. Select [Save] of the [File] menu. The input processing records will be reflected in project analysis information.



## 9. Agile+ Relief diagnosis report generation tool

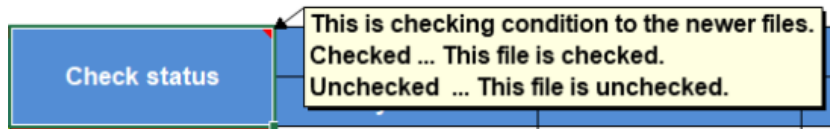
The following explains how to use the Agile+ Relief diagnosis report generation tool.

### 9.1. Composition of Agile+ Relief diagnosis report

The Agile+ Relief diagnosis report is generated as a file of the book form of "Microsoft(R) Office Excel(R)". The Agile+ Relief diagnosis report is composed of two or more sheets and describes the outline of each sheet as follows.

Details of the output information of each sheet is described by using the comment function of the cell of "Microsoft(R) Office Excel(R)" in the generated Agile+ Relief diagnosis report, and refer to that, please.

[Comment Example]



#### 9.1.1. Title

[Sheet name]

Title

[Description]

This sheet is a cover of the Agile+ Relief diagnosis report. The list of the project name of the Agile+ Relief diagnosis report has been described.

[Image]

Project information				
No	Project name	Agile+ Relief version	Agile+ Relief product name	Output date
1	SampleJava - Project1	V1.1.1	Agile+ Relief J	2020/05/29 15:37

## 9.1.2. General diagnosis report

[Sheet name]

Diagnosis\_C、Diagnosis\_J

[Description]

This sheet is a start page to see the Agile+ Relief diagnosis report.

Because the problem item is marked to yellow, the presence of the problem and advice to each problem can be easily confirmed. In addition, it is possible to cooperate to the part where it describes the problem more concretely.

[Image]

**General diagnosis report**

**1. Generalization**

Diagnosis result: **NG** (Legend: Unchecked)

Opinion and comments:

**2. Diagnosis report**

**(1) List of indication messages diagnosis report**

Check	Advice
List of indication messages has unapproved items. Make sure all items, approve them.	
A lot of indications of reliability (in the pgj10200s and pgj20200s) have been detected. Especially a lot of [Initialization gap], [Dangerous record] detected. please by the following viewpoints. - Please check that is not exist a variable used without initializing. - Please check that is no description as to induce a mistake.	
There is a lot of same thing of the cause when a lot of same points have been detected. Please find keyword pattern that becomes a feature if some points are confirmed and the cause is the same. If the keyword pattern is used, other same points can be efficiently confirmed. Moreover, there is a possibility being detected for a lot of same messages depending on the characteristic of the project. In that case, please examine whether to apply the rule.	

**- Approve status of indication messages**

Status	Amount
Approved indication messages	0
Not approved indication messages	28
Total	28

**- Tendency to detection of indication messages**

**1. Tendency to detection of quality characteristic**

**2. Tendency to indication with problem characteristic to focus on reliability**

### 9.1.3. List of indication messages diagnosis report

[Sheet name]

MessageList\_C、MessageList\_J、MessageList\_Guideline

\*MessageList\_Guideline is a peculiar sheet to the evidence of the Agile+ Relief C/C++.

[Description]

This sheet is a list of the indication messages detected with Agile+ Relief C/C++ or Agile+ Relief J. Please examine the action of the program correction etc. about the message marked in yellow.

[Image]

List of indication messages diagnosis report							
Option functions		Legend					
Take in past diagnosis report...		Unconfirmed					
<b>(1) Overview of diagnosis status</b>							
Status				Amount			
Approved indication messages				0			
Not approved indication messages				28			
Total				28			
				Authorizer		Authorized date	
				Authorizer comment			
<b>(2) Detail of diagnosis status</b>							
- List of indication messages							
No	Approve	Project name	File	Line	Importance degree	Rule code	Message
1		SampleJava - Project1	E:\ppr\soft\SampleJava\Project1\src\sample\gasstand\GasStand.java	7	S-Reliability	pg10214	"Explicitly initialize all the static field variables." is being violated.
2		SampleJava - Project1	E:\ppr\soft\SampleJava\Project1\src\sample\gasstand\GasStand.java	9	S-Reliability	pg10212	The field variable may not be initialized in the constructor.
3		SampleJava - Project1	E:\ppr\soft\SampleJava\Project1\src\sample\gasstand\GasStand.java	37	ntainability/Rea	pg10068	Literal is not described on the left of comparison expression.
4		SampleJava - Project1	E:\ppr\soft\SampleJava\Project1\src\sample\gasstand\GasStandSimulation.java	9	S-Reliability	pg10212	The field variable may not be initialized in the constructor.
5		SampleJava - Project1	E:\ppr\soft\SampleJava\Project1\src\sample\gasstand\GasStandSimulation.java	13	W-Reliability	pg10244	"Do not perform standard output or the output that may cause stand
6		SampleJava - Project1	E:\ppr\soft\SampleJava\Project1\src\sample\gasstand\GasStandSimulation.java	16	S-Reliability	pg10239	There is no invocation of java.io.FilterInputStream.close() in the Fina
7		SampleJava - Project1	E:\ppr\soft\SampleJava\Project1\src\sample\gasstand\GasStandSimulation.java	19	W-Reliability	pg10244	"Do not perform standard output or the output that may cause stand
8		SampleJava - Project1	E:\ppr\soft\SampleJava\Project1\src\sample\gasstand\GasStandSimulation.java	25	W-Reliability	pg10244	"Do not perform standard output or the output that may cause stand

## 9.1.4. Source code metrics diagnosis report

[Sheet name]

Metrics\_C, Metrics\_J

[Description]

This sheet is a result of the diagnosis of the number of lines of the program, number of methods, and the complexity, etc. Please confirm the item marked by dyeing by the viewpoint of maintainability and readability.

In the diagnosis, the presence of abnormality is judged based on the threshold set beforehand. The threshold can be changed. Please refer to "[9.2.1 Customize Agile+ Relief Diagnosis report template](#)" for the change of the threshold.

[Image]

**Source code metrics diagnosis report**

Option functions: Take in past diagnosis report...

Legend: Unconfirmed, Exceeded violation values, Exceeded warning values

**(1) Overview of diagnosis status**

**- File metrics**

Status	Number of files
Approved violation/warning metrics	0
Not approved violation/warning metrics	0
No problem	9
Total	9

**- Method metrics**

Status	Number of methods
Approved violation/warning metrics	0
Not approved violation/warning metrics	0
No problem	18
Total	18

**(2) Detail of diagnosis status**

**- File metrics**

No	Approve	Project name	File	All step size	Executed step size	Number of methods	Number of indication messages	Number of detect messages (per line)	Developer comment
				Diagnostic index	Violation value[>=]	-	4000	-	-
				Warning value[>=]	-	2000	-	-	-
1	-	SampleJava - Project1	E:\ogr\soft\SampleJava\Project1\src\sample\gasstand\Automobi.java	21	14	3	0	0.00	No problem
2	-	SampleJava - Project1	E:\ogr\soft\SampleJava\Project1\src\sample\gasstand\DiscountGasStand.java	26	18	3	0	0.00	No problem

## 9.2. Preparation of Agile+ Relief diagnosis report generation tool use

### 9.2.1. Customize Agile+ Relief Diagnosis report template

In the diagnosis of the source metrics, the problem part is detected by comparing it with the threshold set beforehand. The threshold of default is set to the Agile+ Relief diagnosis report template that Agile+ Relief offers. Please go according to the following procedures when you change the threshold.

- \* The Agile+ Relief diagnosis report template is the one sharing in the product (project group). It becomes impossible to verify the diagnose result by the same standard when the developer uses each template. Before using the Agile+ Relief Diagnosis report generation tool, please prepare the common Agile+ Relief diagnosis report template.
- \* Specify the same set value by this customization when [Dangerous /Attention Field Definition File] specified at Agile+ Relief execution time is edited and used.

**Step 1. Copy Agile+ Relief diagnosis report template file**

Copy Agile+ Relief diagnosis report template file that provided by Agile+ Relief to any folder. Agile+ Relief Diagnosis report template is saved in the name of [ARRReport\_Template\_qm.xlsm] in the [EvidenceTool] folder under the install folder of Agile+ Relief.

**Step 2. Open Agile+ Relief diagnosis report template file**

Open Agile+ Relief diagnosis report template file that is copied by step 1. Thresholds are set in the following two sheets.

Set sheet name	Kind of threshold
Metrics_C	Threshold concerning C/C++ program
Metrics_J	Threshold concerning Java program

**Step 3. Change the threshold**

Change the threshold on set sheet of threshold that described by step 2. The change part is a cell enclosed in a red line in the following figures. Each item that sets the threshold is explained in the cell where the title of the item in the Agile+ Relief diagnosis report template is described by using the comment function of the cell of "Microsoft(R) Office Excel(R)" in the generated Agile+ Relief diagnosis report.

[Threshold concerning file]

Method name		All step size	Executed step size	Number of methods	Number of indication messages	Number of detect messages (per line)
Diagnostic index	Violation value(>=)	-	4000	-	-	-
	Warning value(>=)	-	2000	-	-	-

[Threshold concerning function (or method)]

Method name		All step size	Executed step size	Nesting levels	Complexity 1	Complexity 2	return	break	continue	if1	while1	dowhile1	for1	switch1	if2	while2	dowhile2	for2	switch2	Number of indication messages	Number of detect messages (per line)	
Diagnostic index	Violation value(>=)	600	400	20	50	-	-	-	-	20	20	20	20	20	-	-	-	-	-	-	-	-
	Warning value(>=)	400	200	10	20	-	-	-	-	10	10	10	10	10	-	-	-	-	-	-	-	-

There are "Violation value" and "Warning value" in the threshold, and the level of the problem can be divided into two stages. The threshold of both needs not always is set.

Explains the value that can be set as a threshold as follows.

Value that can be set	Meaning
Integral Values	The value of 0 or more is set when want to check the violation and the warning. "Violation value" sets a value that is bigger than "warning value".
-(hyphen)	When neither the violation nor warning are checked, "-" is set.

#### Step 4. Share Agile+ Relief diagnosis report template

The Agile+ Relief diagnosis report template is saved and closes when the change of the threshold is completed according to step 3.

The Agile+ Relief diagnosis report template is necessary for executing the Agile+ Relief Diagnosis report generation tool. The customized Agile+ Relief diagnosis report template is distributed to the member of the development project or it stores in the file server etc. and it shares.

### 9.3. Generate Agile+ Relief diagnosis report

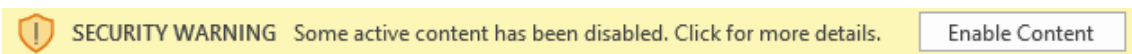
The Agile+ Relief Diagnosis report generation tool is executed, and the Agile+ Relief diagnosis report is generated. Please operate it according to the following steps.

#### Step 1. Run Agile+ Relief Diagnosis report generation tool

Please run the Agile+ Relief Diagnosis report generation tool from the start menu of Windows(R). "Microsoft(R) Office Excel(R)" runs, and the Agile+ Relief Diagnosis report generation tool is displayed.

#### Step 2. Make the macro effective

The Agile+ Relief Diagnosis report generation tool is being offered as VBA macro of "Microsoft(R) Office Excel(R)". When the Agile+ Relief Diagnosis report generation tool is started, the message of the security warning might be displayed. For that case, please click [Enable Content] button and make the macro effective.



#### Step 3. Generate Agile+ Relief diagnosis report

Please execute the Agile+ Relief Diagnosis report generation tool and generate the Agile+ Relief diagnosis report. Please execute the Agile+ Relief Diagnosis report generation tool according to the procedure described in [Diagnosis report generation] sheet in the Agile+ Relief Diagnosis report generation tool.

### 9.4. Cautions

Please refer to [Diagnosis report generation] sheet in the Agile+ Relief Diagnosis report generation tool for notes and output messages concerning the Agile+ Relief Diagnosis report generation tool.

## 10. Improvements

### 10.1. Improvement in V.1.1.1

It demonstrates the changes and improvements made from [Agile+ Relief V1L10] to [Agile+ Relief V1.1.1].

#### 10.1.1. Support to C/C++

Add C/C++ to the parsing target. For analyzing source code of C/C++ by using source analysis mode of Agile+ Relief, Agile+ Relief C/C++ is required.

#### 10.1.2. Correspondence to the latest OS/MW environment

The following has been added as the OS/MW environment that Agile+ Relief C/C++ operates.

[OS Environment]                   \* Only the x64 edition corresponds.  
- Windows(R) 11

[OS Environment]                   \* Only the floating license version for 64bit OS corresponds.  
- Windows Server(R) 2022  
- Red Hat(R) Enterprise Linux(R) 8

[Server virtualization software]                   \* Only the floating license version corresponds.  
- Microsoft(R) Windows Server(R) 2022 Hyper-V(R)  
- VMware vSphere® 7.0

[Microsoft(R) Office Excel(R)]                   \* Using "Diagnosis report generation tool".  
- Microsoft(R) Office Excel(R) 2021



## 11. To the customers using V1L10

### 11.1. Change of Project Structure Definition File

C/C++ is supported since Agile+ Relief V1.1.1. Hence, LANG key of [SOURCES] section in project structure definition file must be specified now. For details on how to specify, please refer to "[4.1.1 Project Structure Definition File](#)".

The following functions can be used when analyzing C/C++ code only.

Function	Note
Indication message viewer	Cooperating with Agile+ Relief C/C++ is required.
MISRA violation check	Cooperating with Agile+ Relief C/C++ is required.
IPA/SEC-C/C++ violation check	Cooperating with Agile+ Relief C/C++ is required.