FUJITSU Software
Agile$^+$ Relief C/C++ V1.1.1

Command Manual

B1WD-3609-01ENZ0(00)
January 2023

Foreword

Agile$^+$ Relief C/C++(referred to below as Agile$^+$ Relief) is an application that supports program review by analyzing source programs and header files written in C and C++.

This manual contains explanations of the messages that Agile$^+$ Relief may output when it finds possible program errors. More information about these error messages may be found in the "Command Manual".

Microsoft, Windows and Visual C++ are registered trademarks of Microsoft Corporation in the United States and other countries.

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

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

Note

- 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.

Contents

Tables

1. Command

In Agile^{+} Relief, the following commands are available.

- pgr5
  Messages for analysis of the specified C/C++ program will be output. pgr5 invokes the following commands and files.

Table 1 Command Name/File Name List

Command Name	Function Summary
epomcpp (preprocess module)	Performs preprocessing of source files and outputs messages as single file units.
pgrc (C analysis module)	Performs C language syntax analysis and outputs messages as single file units.
pgrcc (C++ analysis module)	Performs C++ language syntax analysis and outputs messages as single file units.
pgrlink (link module)	According to the result of C language syntax analysis, makes the cross-reference list, outputs messages for multi-file checks.
mes_display (Message output module)	Displays message information regarding the preprocess, C analysis, C++ analysis and link modules.
Files in the directory of EnvInfo	Files to support exclusive compiler syntax.
MSG.MGR file in the directory of MessageInfo	Files to group the messages.

- pgrmetrics

  Checks the result of pgr5 analysis (.ao file and .fd file) and outputs metrics messages.

- pgrfake

  Select the required option of pgr5 command from command parameters of the compiler, and execute the pgr5 command.

1.1 **pgr5** Command

1.1.1. Command Function

Inputs C/C++source files and outputs messages to standard output. Outputs error messages relevant to command errors (i.e. option errors) and runtime errors (i.e. analysis errors) to standard errors. [1]

[1] There are two categories of messages: single file check and multi-file check. For C++ source, multi-file check is unavailable.

1.1.2. Command Format

[pgr5 command] (hereafter referred to as pgr5)

```
% pgr5 [-V][-E][-c][-irstng][-l symbol file]
       [-K Compiler][-S sourcecode]
       [-D macro[=value]][-U macro][-A token]
       [-I private directory][-Y standard directory]
       [--include pre-reading_file]
       [-O option file][--slashcomment][--cplusplus]
       [-H type=type size][-B type=alignment value]
       [-F identifier file][--for_scope_disable]
       [--C99][--enable_gcc_keywords]
       [--relaxed_escaped_newline]
       [--output_code output code][--csv][-G group file]
       [--project project name][--qm]
       [-Z definition file for message checking]
       [--output_source_only]
       [--temp-file full path output file name]
       [-N naming rule file]
       [--auto_pch][--pgrdata_dir pch_dir][--recreate_pch]
       [--restart Wide-ranging Detective execution record file]
       [--cert]
       [--base][--diff]
       Input file1 [input file2...]
```

Note 1: [ ]means the contents are omissible.

Note 2: Environmental variable setting is required for pgr5 startup. Please see also [1.1.6 Environmental Variable].

Note 3: Even if "-irstng" and "-l symbol file" options supported in previous versions can be used, they have no effect on the execution and messages will be output for the use of the according option. For details regarding the modifications of purpose or usage on the other options, please see also Appendix B.

1.1.3. Options Instruction

The instructions for each option of pgr5 are as follows.

[Version Information]

-V	To display the Version, Level and Release Number of pgr5 command and each sub-program. Once this option is used, it may invalidate other options.
[Message output Selection]	
-Z definition file for message checking	If only want to output the needed messages, definition file for message checking must be created and specified. This option can be used more than once to specify multiple definition files for message checking. If this option is not specified, all messages will be output. Note that messages of ! group(fatal error) will be output unconditionally. Please make the definition file for message checking according to the following rules: ● The format of the first line is fixed: ;Rule=PGRelief ● If a line begins with semicolon (;), the line is treated as the comment. ● Please use the integer above 0 to the maximum 9999 for message indication ID. ● Only one Message ID or Message group can be specified in a line. ● Message group must be specified with single-byte lowercase English letter of one character. (a-z) ● If it is prohibited to output some objects, the output objects and prohibited objects should be specified. Then, the indications of all output objects except for the prohibited objects will be output. When an object is output, the user can enter "all" (half-width lowercase) indicating that all the objects are to be output, or enter one or more group names. A prohibited object includes a minus sign (-) before the indication ID or the group name. The message indication ID not existed cannot be recorded. Example 1: When pgr0519 and the "a" group are to be output. ;Rule=PGRelief 519 a Example 2: When it is prohibited to output pgr0519. ;Rule=PGRelief all

	-519 Example 3: When it is prohibited to output the "b" group. ;Rule=PGRelief all -b Example 4: When pgr0519 and all indications except for pgr0520 (an indication in "a" group) in the "a" group are to be output. ;Rule=PGRelief a -520 -519
--output_source_only	When this option is specified, messages relating to header files will not be output.
[Suppressing message output]	
--base	Sets the reference point for messages output by the pgr5 command. If this option is specified, no analysis will be performed. This option cannot be specified with the --diff option.
--diff	Marks and outputs messages that have been checked in the past or that have set a reference point. This option cannot be specified with the --base option.
[Specify naming rule check]	
-N naming rule file	When you wish to check the naming rule, you need to make a naming rule file and specify it. Also, this option does not support multiple specifications. For the method of recording naming rule files, and indicated conditions, please refer to [6 Naming rule check]

[Message Group]

-G group file	When you wish to modify the default group of a message, please create a user-defined group file and then use it in the -G option. If it is used multiple times, only the last designation is effective. Please create a user-defined group file in the following forms: • In comment, a semicolon (;) should appear at the beginning of a row. • A row consisting of two elements should have each element separated by a comma (,). **Message ID, Group Name** Whitespace is not allowed in front of comma(,). • A Message ID should be represented with integer values above 0

	(with the maximum being 9999). - A Group name should be represented with single-byte lowercase English characters of 1byte. The following provides n example illustrating how to modify pgr0009 and pgr0027 to be of *b* group: [Group File Example] ; MessageID,group 9,b 27,b * Write only Message ID and group these you want to change. The default group description: **Note: The *!* group(will cause a fatal error) is unchangeable.** !: fatal errors equivalent to compiler error a: defects resulting in program bugs b: low maintainability, portability and readability c: worse performance d: effected by the portability and the change upon the type handling f: [The Embedded C++] rule violation g: in firmware development, low maintainability, portability and readability h: low program health n: messages that are indicating the record that violates the naming rule. For more information regarding default groups of a Message ID, please see also MessageInfo/MSG.MGR in the installation directory. MSG.MGR is with the same form of user-defined group file. (If you want to group to be changed, you have to create a user-defined group file without copying MessageInfo/MSG.MGR.)

[Compiler]

-K Compiler	Compiler options are as follows. (Note the -K option can only be used once.) [DIAB DATA Company] **DIABDATA/DCCANSI** The reserved word of DIAB DATA Company D-cc compiler is supported. When the -Xansi option is selected in D-cc compiler, please

use DIABDATA/DCCANSI.

DIABDATA/DCCKANDR
The reserved word of DAIB DATA company D-cc compiler is supported. When the -Xk-and-r options are selected in D-cc compiler, please use DIABDATA/DCCKANDR.

DIABDATA/DCCSTRICT
The reserved word of DAIB DATA company D-cc compiler is supported. When the -Xstrict-ansi option is selected in D-cc compiler, please use DIABDATA/DCCSTRICT.

DIABDATA/DCC
The reserved word of DAIB DATA company D-cc compiler is supported. When options other than the above are selected in D-cc compiler, please use DIABDATA/DCC.

0

[GNU Products]

GNU/GCC
The reserved word of GNU project C Compiler 2.9x before is supported.

The content from // to the line shift is considered to be the comment.

If only whitespaces appear between \ and a new line feed, it will be ignored.

Also remember that #include_next <...> is the analysis object. Default type size is as follows:

The type size of *wchar_t* and *bool* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment *value* of *wchar_t* and *bool* is **32bit**. (C++ source)

GNU/GCC3
The reserved word of GNU project C Compiler 3.0 (command name gcc/g++) is supported.

The C99 syntax (i.e. the content from // to new line feed is regarded as the comment) is supported.

If only whitespace appears between \ and a new line feed, it will be ignored.

In addition, remember that #include_next <...> is the analysis object.

Default type size is as follows:

The type size of *wchar_t* and *bool* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of wchar_t and bool is **32bit**. (C++ source)

GNU/GCC3_C90
The reserved word of GNU project C Compiler 3.0 (command name gcc/g++) is supported. When C90 syntax is selected through the -ansi and -std= options, please use GNU/GCC3_C90. However, if both --C99 are used, the pg5 command will not be guaranteed.

The content from // to the line shift is considered to be the comment.

If only whitespaces appear between \ and a new line feed, it will be ignored.

In addition, remember that #include_next <...> is the analysis object.

Default type size is as follows:

The type of *wchar_t* and *bool* is **32bit**.(C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* and *bool* is **32bit**. (C++ source)

GNU/GCC4

The reserved word of C, C++Compiler 4.0-4.8 (command name gcc/g++) that GNU Compiler Collection has to offer is supported.

The C99 syntax (i.e. the content from // to new line feed is regarded as the comment) is supported.

If only whitespace appears between \ and a new line feed, it will be ignored.

In addition, remember that #include_next <...> is the analysis object.

Default type size is as follows:

The type size of *wchar_t* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of wchar_t is **32bit**. (C++ source)

GNU/GCC4_C90

The reserved word of C, C++Compiler 4.0-4.8 (command name gcc/g++) that GNU Compiler Collection has to offer is supported. When C90 syntax is selected through the -ansi and -std= options, please use GNU/GCC4_C90. However, if both -C99 are used, the pgr5 command will not be guaranteed.

The content from // to the line shift is considered to be the comment.

If only whitespaces appear between \ and a new line feed, it will be ignored.

In addition, remember that #include_next <...> is the analysis object.

Default type size is as follows:

The type of *wchar_t* and *bool* is **32bit**.(C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* and *bool* is **32bit**. (C++ source)

GNU/GCC4_C11

The reserved word of C, C++Compiler 4.0-4.8 (command name gcc/g++) that GNU Compiler Collection has to offer is supported. When C11 syntax is selected through the -ansi and -std= options, please use GNU/GCC4_C11.

The C99 syntax (i.e. the content from // to new line feed is regarded as the comment) is supported.

The C11 syntax (i.e. syntax such as _Atomic) is supported.

If only whitespace appears between \ and a new line feed, it will be ignored.

In addition, remember that #include_next <...> is the analysis object.

Default type size is as follows:

The type size of *wchar_t* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of wchar_t is **32bit**. (C++ source)

*To the customer who uses the compiler specification key to GNU/GCC4 and GNU/GCC4_C90 and GNU/GCC4_C11

- The error might occur when analyzing it because the setting of the following macro of the environment dependence is insufficient.

  - __SSE4A__ - __SSE4_2__ - __SSE2__ - __SSE3__

  - __SSSE3__ - __SSE__ - __MMX__ - __SSE4_1__

  Please add, and analyze the setting of macro according to the following procedure when an analytical error occurs.

  1) Following gcc/g++ command is executed, and information on macro is acquired.

     Example: gcc -v -E -dM a.c

     g++ -v -E -dM a.cc

     (The content must specify the File of the null line in 0 bytes for a.c and a.cc.)

  2) A macro optional setting when the macro that becomes an Element is described in information acquired in 1), the value of macro is set, and the analysis is executed.

     Example: When you describe "#define __SSE__ 1"

     pgr5 -KGNU/GCC4 **-D __SSE__=1** sample.cc

     (In the pgr5 command, the value of macro is set optional "-D".)

- The error might occur if the source program that does the following header file in include is analyzed.

  - bitset - type_traits

  Please analyze it after making an alternative header file according to the following procedure when an analytical error occurs.

  1) The corresponding header file is copied in the arbitrary directory.

  2) It is corrected that the alternative header file copied by 1) is

opened, and an analytical error does not occur.

3) The storage directory of the alternative header file made by 1) is added to the retrieval passing of the include file, and the analysis is executed.

Please add for the storage directory of an alternative header file to be specified from the storage directory of the header file that becomes a copy source ahead when the retrieval passing is added.

Example: When the header file of "C:\system\include" is copied in "C:\inc-custom", and an alternative header file is made

```
Pgr5 -KGNU/GCC4 -Y "C:\inc-custom"
                         -Y "C:\system\include" Sample.cc
```

(In the pgr5 command, the retrieval passing of the include file is set optional "-I" and "-Y".)

[Mentor Graphics Japan Co.Ltd. Products]

MGJ/MRIPC
The reserved word of Mentor Graphics Japan Co., Ltd. MRI compiler of PC is supported.

MGJ/MRISUN
The reserved word of Mentor Graphics Japan Co., Ltd. MRI compiler of UNIX is supported.

[HITACHI Products]

HITACHI/SH
The reserved word of HITACHI SH compiler is supported.

HITACHI/H8
The reserved word of HITACHI H8 compiler is supported.

Default type size is as follows:

The type size of *int* is **16bit**.

The type size of *long double* is **64bit**.

Default alignment value is as follows:

The alignment value of *int, long, float, double, long double* and *pointer* is **16bit**.

[FUJITSU Products]

FUJITSU/FCC401C
The reserved word of FUJITSU C Development Package 4.0.1 (command name fcc) is supported. When -Xc option is selected in compiler, please use FUJITSU/FCC401C.

FUJITSU/FCC401
The reserved word of FUJITSU C Development Package 4.0.1 (command name fcc) is supported. When the option other than above is

	selected, please use FUJITSU/FCC401. **FUJITSU/SOFTUNEF2MC16B** The reserved word of FUJITSU SOFTUNE F2MC-16C compiler is supported. When -B option is selected in compiler, please use FUJITSU/SOFTUNEF2MC16B. The content from // to the line shift is considered to be the comment. Default type size is as follows: The type size of *int* is **16bit**. The type size of *long double* is **64bit**. Default alignment value is as follows: The alignment value of *int, long, float, double, long double* and *pointer* to be of **16bit**. **FUJITSU/SOFTUNEF2MC16** The reserved word of FUJITSU SOFTUNE F2MC-16C compiler is supported. When the option other than above is selected, please use FUJITSU/SOFTUNEF2MC16. Default type size is as follows: The type size of *int* is **16bit**. The type size of *long double* is **64bit**. Default alignment value is as follows: The alignment value of *int, long, float, double, long double and pointer* is **16bit**. **FUJITSU/SOFTUNEFRB** The reserved word of FUJITSU SOFTUNE FR C compiler is supported. When -B option is selected, please use FUJITSU/SOFTUNEFRB. The content from // to the line shift is considered to be the comment. Default type size is as follows: The type size of *long double* is **64bit**. Default alignment value is as follows: The alignment value of *double* and *long double* is **32bit**. **FUJITSU/SOFTUNEFR** The reserved word of FUJITSU SOFTUNE FR C compiler is supported. When the option other than above is selected, please use FUJITSU/SOFTUNEFR. Default type size as follows: The type size of *long double* is **64bit**. Default alignment value is as follows: The alignment value of *double* and *long double* is **32bit**. [Microsoft Visual C++ Compiler]

MS/VC1

The reserved word of Microsoft Visual C++ compiler V1.x before is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variables in the initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement.

Default type size is as follows:

The type size of *int* is **16bit**.

The type size of *long double* is **64bit**.

Default alignment value is as follows:

The alignment value of *int* is **16bit**.

MS/VC4

The reserved word of Microsoft Visual C++ compiler V4.0 (MFC unused) is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variables in the initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

MS/VC4MFC

The reserved word of Microsoft Visual C++ compiler V4.0 (MFC used) is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variables in the initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

MS/VC42

The reserved word of Microsoft Visual C++ compiler V4.2 (MFC unused) is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variables in the initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

MS/VC42MFC

The reserved word of Microsoft Visual C++ compiler V4.2 (MFC used) is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variables in the initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the for statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

MS/VC5

The reserved word of Microsoft Visual C++ compiler V5.0 (MFC unused) is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variables in the initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

MS/VC5MFC

The reserved word of Microsoft Visual C++ compiler V5.0 (MFC used) is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variables in the initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

MS/VC6

The reserved word of Microsoft Visual C++ compiler V6.0 (Win 32bit

API and MFC unused) is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variable in initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

MS/VC6MFC

The reserved word of Microsoft Visual C++ compiler V6.0 (Win 32bit API and MFC used) is supported.

The content from // to the line shift is considered to be the comment.

Modify the indexing sequence of #include "include filename" to be the inherent sequence of Microsoft.

Set the scope of declared variable in initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the for statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

MS/VC6WIN64

The reserved word of Microsoft Visual C++ compiler V6.0 (Win 64bit API and MFC unused) is supported.

Comment is that the content begins with // until new line feed.

Modify the indexing sequence of #include "header file" to be the inherent sequence of Microsoft.

Set the scope of declared variable in initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement.

Default type size is as follows:

The type size of *long double* is **64bit**.

Default alignment value is as follows:

The alignment value of *pointer* is **64bit**.

MS/VC2003

The reserved word of Microsoft Visual C++.NET 2002(MFC not used in Win32bit API) and Microsoft Visual C++.NET 2003 (MFC not used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block

	containing for statement. Change the defaults of type size as follows: -long double type size is 64 bit However, it is not correspondent to the code (*) created in Managed Extensions for C++. (* When #using < mscorlib.dll > exists in the code, When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option needs to be specified during compilation etc.) **MS/VC2003MFC** The reserved word of Microsoft Visual C++ .NET 2002 (MFC used in Win32bit API) and Microsoft Visual C++ .NET 2003 (MFC used in Win32bit API) is supported. The content from // to the line feed is regarded as comment. The indexing sequence of #include "include file name" is changed to Microsoft special. The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement. Change the defaults of type size as follows: -long double type size is 64 bit However, it is not correspondent to the code (*) created in Managed Extensions for C++. (* When #using < mscorlib.dll > exists in the code, When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option needs to be specified during compilation etc.) **MS/VC2005** The reserved word of Microsoft Visual C++ 2005 (MFC not used in Win32bit API) is supported. The content from // to the line feed is regarded as comment. The indexing sequence of #include "include file name" is changed to Microsoft special. The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement. Change the defaults of type size as follows: -long double type size is 64 bit However, source(*) created through Managed Extensions for C++ or Common Language Runtime is not supported. (* When #using < mscorlib.dll > exists in the code, When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option needs to be specified during compilation etc.)

MS/VC2005MFC

The reserved word of Microsoft Visual C++ 2005 (MFC used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement.

Change the defaults of type size as follows:

-long double type size is 64 bit

However, source (*) created through Managed Extensions for C++ or Common Language Runtime is not supported.

(* When #using < mscorlib.dll > exists in the code,

When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option needs to be specified during compilation etc.)

MS/VC2008

The reserved word of Microsoft Visual C++ 2008 (MFC not used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement.

Change the defaults of type size as follows:

-long double type size is 64 bit

However, source(*) created through Managed Extensions for C++ or Common Language Runtime is not supported.

(* When #using < mscorlib.dll > exists in the code,

When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option needs to be specified during compilation etc.)

MS/VC2008MFC

The reserved word of Microsoft Visual C++ 2008 (MFC used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement.

Change the defaults of type size as follows:

-long double type size is 64 bit

However, source (*) created through Managed Extensions for C++ or Common Language Runtime is not supported.

(* When #using < mscorlib.dll > exists in the code,

When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option needs to be specified during compilation etc.)

MS/VC2010

The reserved word of Microsoft Visual C++ 2010 (MFC not used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement.

Change the defaults of type size as follows:

-long double type size is 64 bit

However, source(*) created through Managed Extensions for C++ or Common Language Runtime is not supported.

(* When #using < mscorlib.dll > exists in the code,

When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option needs to be specified during compilation etc.)

MS/VC2010MFC

The reserved word of Microsoft Visual C++ 2010 (MFC used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement.

Change the defaults of type size as follows:

-long double type size is 64 bit

However, source (*) created through Managed Extensions for C++ or Common Language Runtime is not supported.

(* When #using < mscorlib.dll > exists in the code,

When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option needs to be specified during compilation etc.)

MS/VC2012

The reserved word of Microsoft Visual C++ 2012 (MFC not used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement.

Change the defaults of type size as follows:

-long double type size is 64 bit

However, source(*) created through Managed Extensions for C++, Common Language Runtime or C++ Component Extensions is not supported.

(* When #using < mscorlib.dll > exists in the code,

When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option or the /ZW option needs to be specified during compilation etc.)

MS/VC2012MFC

The reserved word of Microsoft Visual C++ 2012 (MFC used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement.

Change the defaults of type size as follows:

-long double type size is 64 bit

However, source(*) created through Managed Extensions for C++, Common Language Runtime or C++ Component Extensions is not supported.

(* When #using < mscorlib.dll > exists in the code,

When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option or the /ZW option needs to be specified during compilation etc.)

MS/VC2013

The reserved word of Microsoft Visual C++ 2013 (MFC not used in Win32bit API) is supported.

The content from // to the line feed is regarded as comment.

The indexing sequence of #include "include file name" is changed to Microsoft special.

The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement.

Change the defaults of type size as follows:

	-long double type size is 64 bit However, source(*) created through Managed Extensions for C++, Common Language Runtime or C++ Component Extensions is not supported. (* When #using < mscorlib.dll > exists in the code, When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option or the /ZW option needs to be specified during compilation etc.) **MS/VC2013MFC** The reserved word of Microsoft Visual C++ 2013 (MFC used in Win32bit API) is supported. The content from // to the line feed is regarded as comment. The indexing sequence of #include "include file name" is changed to Microsoft special. The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement. Change the defaults of type size as follows: -long double type size is 64 bit However, source(*) created through Managed Extensions for C++, Common Language Runtime or C++ Component Extensions is not supported. (* When #using < mscorlib.dll > exists in the code, When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option or the /ZW option needs to be specified during compilation etc.) **MS/VC2015** The reserved word of Microsoft Visual C++ 2015 (Win32bit API) is supported. Use this compiler specification key even if you are using MFC. The content from // to the line feed is regarded as comment. The indexing sequence of #include "include file name" is changed to Microsoft special. The scope of variables declared in the initial settings of for statement within C++ source, which is finished until the very end of outmost block containing for statement. Change the defaults of type size as follows: -long double type size is 64 bit However, source(*) created through Managed Extensions for C++, Common Language Runtime or C++ Component Extensions is not supported. (* When #using < mscorlib.dll > exists in the code, When Keyword= "ManagedCProj" exists in projectfile of .NET, and

when the /clr option or the /ZW option needs to be specified during compilation etc.)

[GAIO Company Product]

GAIO/XASS

The reserved word of GAIO Xass compiler is supported.

[Texas Instruments Company Products]

TEXAS_INSTRUMENTS/TMS320C6000

The reserved word of Texas Instruments Company TMS320C6000 compiler is supported.

Default type size is as follows:

The type size of long is **40bit**.

The type size of long double is **64bit**.

Default alignment value is as follows:

The alignment value of *long* is **40bit**.

The alignment value of *long double* is **64bit**.

TEXAS_INSTRUMENTS/TI

The reserved word of Texas Instruments Company TI compiler is supported.

This option shall be applied for the customer with -KTI in old version.

[ARM Company Products]

ARM_LTD/ARM

The reserved word of Advanced RISC Machines Company compiler is supported.

ARM_LTD/DEVELOPER_SUITE_V1_2

The reserved word of ARM Company SUITE V1.2 compiler is supported.

ARM_LTD/REALVIEW_V2_0

The reserved word of ARM Company REALVIEW V2.0 compiler is supported.

ARM_LTD/REALVIEW_V2_2

The reserved word of ARM REALVIEW V2.2complier is supported.

ARM_LTD/REALVIEW_V3_0

The reserved word of ARM REALVIEW V3_0 complier is supported.

The content from // to the line shift is considered to be the comment.

The default type SIZE is changed to the following.

	- type SIZE of long double is 64 bit that supports _ESCAPE_syntax. For C Language, it supports the specified initialize syntax. **ARM_LTD/REALVIEW_V3_1** The reserved word of ARM REALVIEW V3_1 complier is supported. The content from // to the line shift is considered to be the comment. The default type SIZE is changed to the following. - type SIZE of long double is 64 bit that supports _ESCAPE_syntax. For C Language, it supports the specified initialize syntax. **ARM_LTD/REALVIEW_V4_0** The reserved word of ARM REALVIEW V4.0 complier is supported. The content from // to the line shift is considered to be the comment. The default type SIZE is changed to the following. - type SIZE of long double is 64 bit that supports _ESCAPE_syntax. For C Language, it supports the specified initialize syntax. **ARM** The reserved word that is the same as ARM_LTD/ARM is supported. It is intended to be deleted in this project, so please use ARM_LTD/ARM. Support reserved words that are the same as ARM_LTD/ARM. It is intended to be deleted in this project, so please use ARM_LTD/ARM. [Oracle Company Products] **SUN/SPARC201** The reserved word of Oracle Company SPARCompiler 2.0.1 is supported. Default type size is as follows: The type size of *wchar_t* is **32bit**. (C++ source) Default alignment value is as follows: The alignment value of *wchar_t* is **32bit**. (C++ source) **SUN/SPARC3** The reserved word of Oracle Company SPARCompiler 3.0 is supported. Default type size is as follows: The type size of *wchar_t* is **32bit**. (C++ source) Default alignment value is as follows: The alignment value of *wchar_t* is **32bit**. (C++ source) **SUN/SPARC301C** The reserved word of Oracle Company SPARCompiler 3.0.1 is

	supported. When -Xc option is selected in compiler, please use SUN/SPARC301C. Default type size is as follows: The type size of *wchar_t* is **32bit**. (C++ source) Default alignment value is as follows: The alignment value of *wchar_t* is **32bit**. (C++ source) **SUN/SPARC301S** The reserved word of Oracle Company SPARCompiler 3.0.1 is supported. When -Xs option is selected in compiler, please use SUN/SPARC301S. Default type size is as follows: The type size of *wchar_t* is **32bit**. (C++ source) Default alignment value as follows: The alignment value of *wchar_t* is **32bit**. (C++ source) **SUN/SPARC301** The reserved word of Oracle Company SPARCompiler 3.0.1 is supported. When the option other than above is selected in compiler, please use SUN/SPARC301. Default type size is as follows: The type size of *wchar_t* is **32bit**. (C++ source) Default alignment value is as follows: The alignment value of *wchar_t* is **32bit**. (C++ source) **SUN/SPARC4C** The reserved word of Oracle Company SPARCompiler 4.0 is supported. When -Xc option is selected in compiler, please use SUN/SPARC4C. Default type size is as follows: The type size of *wchar_t* is **32bit**. (C++ source) Default alignment value is as follows: The alignment value of *wchar_t* is **32bit**. (C++ source) **SUN/SPARC4S** The reserved word of Oracle Company SPARCompiler 4.0 is supported. When -Xs option is selected in compiler, please use SUN/SPARC4S. Default type size is as follows: The type size of *wchar_t* is **32bit**. (C++ source) Default alignment value is as follows: The alignment value of *wchar_t* is **32bit**. (C++ source) **SUN/SPARC4** The reserved word of Oracle Company SPARCompiler 4.0 is

supported. When the option other than above is selected in compiler, please use SUN/SPARC4.

Default type size is as follows:

The type size of *wchar_t* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* is **32bit**. (C++ source)

SUN/SPARC42C

The reserved word of Oracle Company SPARCompiler 4.2 is supported. When -Xc option is selected in compiler, please use SUN/SPARC42C.

Default type size is as follows:

The type size of *wchar_t* is 32bit. (C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* is 32bit. (C++ source)

SUN/SPARC42S

The reserved word of Oracle Company SPARCompiler 4.2 is supported. When -Xs option is selected in compiler, please use SUN/SPARC42S.

Default type size is as follows:

The type size of *wchar_t* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* is **32bit**. (C++ source)

SUN/SPARC42

The reserved word of Oracle Company SPARCompiler 4.2 is supported. When the option other than above is selected in compiler, please use SUN/SPARC42.

Default type size is as follows:

The type size of *wchar_t* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* is **32bit**. (C++ source)

SUN/SPARC5C

The reserved word of Oracle Company SPARCompiler 5.0 is supported. When -Xc option is selected in compiler, please use SUN/SPARC5C.

Default type size is as follows:

The type size of *wchar_t* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* is **32bit**. (C++ source)

SUN/SPARC5S

The reserved word of Oracle Company SPARCompiler 5.0 is supported. When -Xs option is selected in compiler, please use SUN/SPARC5C.

Default type size is as follows:

The type size of *wchar_t* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* is **32bit**. (C++ source)

SUN/SPARC5

The reserved word of Oracle Company SPARCompiler 5.0 is supported. When the option other than above is selected in compiler, please use SUN/SPARC5C.

Default type size is as follows:

The type size of *wchar_t* is **32bit**. (C++ source)

Default alignment value is as follows:

The alignment value of *wchar_t* is **32bit**. (C++ source)

[Renesas Company Products]

RENESAS/M16C

The reserved word of renesas M16C complier is supported.

The default type SIZE is changed to the following.

Type size of *long double* is **64 bit**

The content from // to the line shift is considered to be the comment.

RENESAS/M16C80

The reserved word of renesas M16C80 complier is supported.

The default type SIZE is changed to the following.

Type size of *long double* is **64 bit**

The content from // to the line shift is considered to be the comment.

RENESAS/M32C

The reserved word of renesas M32C complier is supported.

The default type SIZE is changed to the following.

Type size of *long double* is **64 bit**

The content from // to the line shift is considered to be the comment.

RENESAS/M32R

The reserved word of renesas M32C complier is supported.

The default type SIZE is changed to the following.

Type size of *long double* is **64 bit**

The default alignment value is changed to the following.

Type size of *double*, *long double* are **32 bit**

The content from // to the line shift is considered to be the comment.

RENESAS/740

The reserved word of renesas 740 complier is supported.

The default type SIZE is changed to the following.

Type size of *int*, *pointer* are **16 bit**

Type size of *double*, *long double* are **32 bit**

The default alignment value is changed to the following.

Type size of *int*, *pointer* are **16 bit**

Type size of *double*, *long double* are **32 bit**

RENESAS/7770

The reserved word of renesas 7770 complier is supported.

The default type SIZE is changed to the following.

Type size of *int* is **16 bit**

Type size of *long double* is **32 bit**

The content from // to the line shift is considered to be the comment.

RENESAS/SuperH

The reserved word of renesas SuperH complier is supported.

The default type SIZE is changed to the following.

Type size of *long double* is **64 bit**

Type size of *bool* is **32 bit** (For C++ SOURCE)

The default alignment value is changed to the following.

Type size of *double, long double,* and *long long* are **32 bit**

Type size of *bool* is **32 bit** (For C++ SOURCE)

The content from // to the line shift is considered to be the comment.

RENESAS/H8S

The reserved word of renesas H8S complier is supported.

The default type SIZE is changed to the following.

Type size of *int* is **16 bit**

Type size of *long double* is **64 bit**

The default alignment value is changed to the following.

Type size of *int, long, float, double, long double, pointer* are **16 bit**

RENESAS/H8_300

The reserved word of renesas H8_300 complier is supported.

The default type SIZE is changed to the following.

Type size of *int* is **16 bit**

Type size of *long double* is **64 bit**

The default alignment value is changed to the following.

Type size of *int, long, float, double, long double*, *pointer* are **16 bit**

RENESAS/78K0R

The reserved word of renesas 78K0R complier is supported.

The default type SIZE is changed to the following.

Type size of *int* is **16 bit**

Type size of *double, long double* are **32 bit**

The default alignment value is changed to the following.

Type size of *int* is **16 bit**

Type size of *double, long double* are **32 bit**

RENESAS/RX

The reserved word of renesas RX complier is supported.

The default type SIZE is changed to the following.

Type size of *double, long double* are **32 bit**

The default alignment value is changed to the following.

Type size of *double, long double, long long* are **32 bit**

[NEC Company Products]

NEC/78K0

The reserved word of NEC 78K0 complier is supported.

The default type SIZE is changed to the following.

Type size of *int* is **16 bit**

Type size of *double, long double* are **32 bit**

The default alignment value is changed to the following.

Type size of *int* is **16 bit**

Type size of *double, long double* are **32 bit**

NEC/78K0S

The reserved word of NEC 78K0S complier is supported.

The default type SIZE is changed to the following.

Type size of *int* is **16 bit**

Type size of *double, long double* are **32 bit**

The default alignment value is changed to the following.

Type size of *int* is **16 bit**

Type size of *double, long double* are **32 bit**

NEC/78K4

The reserved word of NEC 78K4 complier is supported.

The default type SIZE is changed to the following.

	Type size of *int* is **16 bit** Type size of *double, long double* are **32 bit** The default alignment value is changed to the following. Type size of *int* is **16 bit** Type size of *double, long double* are **32 bit** **NEC/V850** The reserved word of NEC V850 complier is supported. The default type SIZE is changed to the following. Type size of *double* is **32bit** The default alignment value is changed to the following. Type size of *double* is **32bit** [IAR SYSTEMS Company Products] **IAR/EWARM7.4** The reserved word of IAR SYSTEMS IAR Embedded Workbench for ARM 7.4 compiler is supported. The default type SIZE is changed to the following. Type size of *long double* is **64bit**. **IAR/EWARM8.1** The reserved word of IAR SYSTEMS IAR Embedded Workbench for ARM 8.1 compiler is supported. The default type SIZE is changed to the following. Type size of *long double* is **64bit**.
--slashcomment	Enable // comment In addition, even if the option is not specified, //comment is enabled for C++ source file. Moreover, once a specific compiler by the -K option is selected, though this option is not specified, //comment will also be enabled. For details, please see also the -K option.
--for_scope_disable	Set the scope of declared variables in the initial settings of the *for* statement (initial settings; condition; update) in C++ source to end until the outmost block of the *for* statement. Moreover, once a specific compiler by the -K option is selected, though this option is not specified, //comment is also enabled. For details, please see also the -K option.
--C99	C99 syntax analysis option (ISO/IEC9899:1999). The option is to enable the following C analysis module of C99 syntax. However, when both this option and -KGNU/GCC3_C90 are enabled, pgr5 command will not be guaranteed. 1. Take *_Bool* as the keyword 2. Take *_Complex* and *_Imaginary* as the keywords

	3. Take *inline* as the keyword 4. Take *restrict* as the keyword 5. Enable //comment 6. Enable macro definition of variable parameter (*__VA_ARGS__*) 7. Enable *_Pragma* preprocess directive 8. Take *__STDC_HOSTED__* and *__STDC_VERSION__* as the defined macro.
--enable_gcc_keywords	Gcc syntax analysis option. This option enables C analysis module of gcc syntax. 1. Enable __label__ 2. Enable typeof 3. Enable __real__ and __imag__ 4. Enable __alignof__ 5. _Start Complex. 6. Start inline.
--relaxed_escaped_newline	If only whitespaces appear between \ and a new line feed, it will be ignored. Please use the option in special gcc syntax analysis.

[Preprocessor Information Options]

-D macro	Define macro as 1. Same as the following code #define *macro* 1 Note)-U option prior if -D and -U are used simultaneously.
-D macro= value	Define macro as value. Same as following code #define macro value
-U macro	The specified system macro will be invalid when macro name is denoted as follows: **__LINE__, __FILE__, __TIME__, __DATE__, __STDC__** When C++ is the source, the macro should be denoted as follows: **__cplusplus** When the C99 syntax options (e.g.-C99, etc) are specified, the macro should be denoted as follows: **__STDC_HOSTED__, __STDC_VERSION__**
-I private	Please specify the directory name for the private include file.

directory	The file specified with #include statement will be indexed in the following paths. When -I option is used more than once, the indexing will be performed in the specified sequence: [include file indexing sequence] *#include "include file"* **:** private include Find the include file within the provided path Directory of the file with this #include statement Private directory specified with the -I option (When the -I option is used more than once, the indexing will be performed in the specified sequence.) Standard directory specified with the -Y option (see also the -Y option) *#include <include file>* **:** standard include Find the include file within the provided path Private directory specified with the -I option (When the -I option is used more than once, the indexing will be performed in the specified sequence.) Standard directory specified with the -Y option (see also the -Y option)
-Y standard directory	To specify the directory for specified standard include file. When the -Y option is used more than once, the indexing will be performed in the specified sequence. Also, if this option is not specified, /usr/include other than Windows edition will be the standard directory. - As for the standard include file taken by specifying this option, message indication is not output excluding a part of message indication. (Please refer to [Message Indications Manual] separately offered for message indication.) - As for the standard include file taken by specifying this option, information on metrics is not output.
--include pre-reading_file	To specify the pre-reading_file. After specified the pre-reading_file is analyzed, the source file to be checked is analyzed. When this option is two or more specified, it processes it in specified order. - Describe the pre-reading_file in the source file and the same character-code and the language (C and C++, etc.). - Even if the pre-reading_file is updated when this option is used together with optional --auto_pch, the analytical result of the include file is not updated. As needed, --recreate_pch is optional and update the analytical result of the include file.
-A name	*Name* is not associated with any token. Same as following code #assert name
-A name(token)	Associate name with token.

	Same as following code #assert name(token) Note) Please notice that the parentheses in shell may be of special meaning.
-E	To output the result of preprocess to standard output. Output the preprocess result first and then the preprocess message. When this option is specified, pgr5 command is ended after preprocess. The result of preprocess includes the following Line No. information: **#line 1 "D:\sample\test.c"** This refers to the line 1 of test.c file.

[User Settings]

-F identifier file	This option is provided to specify the identifier file with identifier information to suppress some message or to output it. It cannot be used more than once. First, the pgr5 command uses the following identifier files in the same directory: In the case of C source (C90 syntax), EnvInfo/*/**C.idt will be used.** In the case of C source (C99 syntax), EnvInfo/*/**C99.idt will be used.** In the case of C++ source, EnvInfo/*/**C++.idt will be used.** If the -K option is not specified, * will be regarded as DEFAULT. Otherwise, * is the parameter of the -K option. In the above identifier file, the ANSI information, such as standard library function is documented. The specified identifier file of this option will be written as follows: Identifier File Pattern: A line beginning with '[' is processed as a label. A line beginning with '*' is processed as a comment. An identifier name is documented from the very beginning of a line. If you want to use multiple identifiers or numbers in a line, please do not place whitespaces among identifiers and numbers, but separate them with a comma (,). For more information regarding available label names and written formats in identifier files, please see also [1.1.3.1 Identifier File Label List]. Content from [label] to [next label] or [file end] is processed as the identifier information of a label.

	<Identifier File Example> *Comment : when -F/local/user/a.exit is specified, and a.exit is as follows: [RETURN_CHECK_FUNCTION] malloc realloc calloc fopen open [NULL_RETURN_FUNCTION] malloc realloc calloc fopen [NULL_POINTER_NG_FUNCTION] atof,1 atol,1 atoi,1 [PARALLEL_FUNCTION] func1,func2,func3 [AREA_RELEASE_FUNCTION] delete,1 [AREA_ACCESS_FUNCTION] memset,1,0,3 memcpy,1,2,3
-H type = type size	This option is provided to modify type size to be type size. Available types are listed below. Please use integer values like 8 or 16 for type size. pgr5 will not check type sizes for errors. char short int long longlong float double longdouble pointer

	enum**[1]** [1] If set to -Henum=O(ON), the type size of enum will be the size of the maximum enum member value. wchar_t (C++ source Only) bool (C++ source Only) _Bool [Example illustrating how to modify the type size of int and pointer to be of 16bit.] **-Hint=16 -Hpointer=16** Default type sizes are described below. If the -K option is used, the following default values may be changed. For details, please see also the -K option. char, signed char, unsigned char **type**:**8bit** short, signed short, unsigned short **type**:**16bit** int, signed int, unsigned int **type**:**32bit** long, signed long, unsigned long**, _Bool type**:**32bit** long long, signed long long, unsigned long long **type**:**64bit** float **type**:**32bit** double **type**:**64bit** long double **type**:**128bit** pointer type:**32bit** enum **type**:**32bit** wchar_t **type**:**16bit** (C++ source) bool **type**:**8bit** (C++ source)
-B type = alignment value	This option is provided to modify type's alignment value to be alignment value. Available types are listed below. Please use integer values like *8* or *16* for alignment value. pgr5 will not check alignment value for errors. char short int long longlong float double longdouble pointer

	enum**[1]** [1] If set to -Benum=0 (ON), the alignment value of enum will be the alignment value of maximum enum member type. wchar_t (C++ source Only) bool (C++ source Only) _Bool [Example to modify the alignment value of *int* and *pointer* to be of 16bit.] -Bint=16 -Bpointer=16 Default type sizes are described below. If *-K* option is used, the following default value may be changed. For details, please see also *-K* option. char**,** signed char**,** unsigned char type:**8bit** short**,** signed short**,** unsigned short type:**16bit** int**,** signed int**,** unsigned int**, _Bool** type:**32bit** long**,** signed long**,** unsigned long type:**32bit** long long**,** signed long long**,** unsigned long long type:**64bit** float type:**32bit** double type:**64bit** long double type:**64bit** pointer type:**32bit** enum type:**32bit** wchar_t type:**16bit** (C++ source) bool type:**8bit** (C++ source)
--cert	This option offers the presentation function of the message only for the CERT option. The license of the CERT option is necessary. Please refer to [Message Indications Manual] separately offered for message indication.

[Settings of Parse Time Reduction]

--auto_pch	The parse result of include file that is read from the source file is saved to the PCH reserved directory that is designated by --pgrdata_dir. Specify when it needs to be used repeatedly in the future parsing. After this option has been specified, the parse time can be reduced. However, please beware that it requires large disk space. The conditions for repeatedly using parse result of include file is as

	follows. If the following conditions are not satisfied, a new parse of include file will be executed and the result will be saved. Information of the include file that was parsed last time has been saved under the PCH reserved directory. The sequencing of the include files and command options are same as the last time when specify --auto_pch and parse. (Whether the include file is updated or not is not included in the conditions.) Only the include file beginning with consecutive #include from the beginning of source file can save the parse result to the PCH reserved directory. At this time, comment line or empty line will be ignored. Example: a.h, b.h, c.h can be used repeatedly in case of the following source files. #include <a.h> #include <b.h> /* comment */ #include <c.h> #define D #include <e.h> When specifying this option, --pgrdata_dir should also be specified.
--pgrdata_dir pch_dir	Specify the location for saving the parse result of the include file read from source file. When specifying this option, --auto_pch should also be specified.
--recreate_pch	All the result saved to the PCH reserved directory should be updated unconditionally. When specifying this option, --auto_pch and --pgrdata_dir should also be specified

[Option File Settings]

-O option file	This option is provided to specify options or filenames. When the option is used more than twice, all option file are valid in order. When an input filename is written in an option file, this option should be specified at the end of a command line. In addition, the specified option file will be searched in the following sequence: in the path specified in option file present directory Please notice the following information regarding option file creation:

	The "-O" option will not appear in option file. Wildcard in option file will not be expanded. Please start the comment with '['. Please enclose the directory and filename with " as a whole. In addition, in Windows please use \\ instead of \ for the path delimiter in the part within ".

[Output Settings]

--output_code output code	To specify the output code of message. **SJIS** :SJIS code **EUC** :EUC code **GB2312** :GB2312 code (*) **UTF8** :UTF-8 code If this option is not specified, then it will be the following output code. - Chinese OS GB2312 code - OS of other language SJIS code for Windows; UTF-8 code for Red Hat Enterprise Linux 5 or 6; EUC code for others (*)Only when "English" is selected when In Agile$^+$ Relief, the following commands are available. is installed, GB2312 can be specified.
--project project name	To specify an output filename for a project name. When the option is specified, the file naming rule is as follows: **Link Result File Name: append .aa after project name** **Link Message File: append .mes after project name** When the option is not specified, the file naming rule is as follows: **Link Result File Name: append .link.aa after the previously specified input file name (with extension)** **Link Message File: append .link.mes after the previously specified input file name (with extension)**
--csv	To output Message in CSV format. [CSV output format]] The following contents are separated with a comma in turn and output. • filename (enclosed with "") • line number • group name (enclosed "") • Message ID (enclosed with "")

	- Message (enclosed with "") **"filename", line number, "group name", "Message ID", "Message"** Note: Double-quote in output message is output as **""**. [Output Example] "D:\sample\a.c", 1,"a", "pgrXXXX", "file ""a.h"" not found." "D:\sample\b.c",5,"b","pgrXXXX"," comma operator cannot be used in #if or #elif statement."
--temp-file full path output file name	Target location for saving analysis results (.ao file and .mes file etc.) should be specified with full path. If this option is not specified, then they will be saved to the same directory as input files. It cannot be specified when performing file checking (multiple files). Followings are the examples of outputs to "c:\output"directory. [Single file] %pgr5 --temp-file c:\output\a.c -c a.c %pgr5 --temp-file c:\output\b.c -c b.c Generate "c:\output\a.c.ao","c:\output\a.c.mes", "c:\output\b.c.ao", "c:\output\b.c.mes" etc. [Multiple files] %pgr5 --project c:\output\Aprj c:\output\a.c.ao c:\output\b.c.ao
--qm	This option is the output format for consolidation mode of Agile^{+} Relief (*1), and for CI (Continuous Integration and continuous integration) Tool Collaboration Jenkins Plugin(*2). [When you use the Agile^{+} Relief] Specify this option to use the analysis result consolidation mode of the Agile^{+} Relief. For the analysis result consolidation mode of the Agile^{+} Relief, please refer to "2.3.2 Procedure of using under analysis result consolidation mode" in "Agile^{+} Relief Manual". [When you use the CI Tool Collaboration Jenkins Plugin] Specify this option to use the Ant, Windows batch files and shell scripts on the Jenkins plugin. (*1) Agile^{+} Relief is a function that enables the visualization of problems related to quality by checking quality data that analyzed the source programs on a daily basis. Please note that using this Agile^{+} Relief requires a license. (*2) CI Tool Collaboration Jenkins Plugin CI (Continuous Integration) Tool Collaboration Jenkins Plugin is plugin to total with CI tool (Jenkins) and to display the analytical result of Agile^{+} Relief. Please note that using this CI Tool Collaboration Jenkins Plugin requires a floating license.

[Input Source Settings]

-S source code	To specify the source code of input source. **SJIS** :SJIS code **EUC** :EUC code **GB2312** :GB2312 code **UTF8** :UTF-8 code When this option is not specified, the source code will be as below. - Chinese OS GB2312 code - OS of other language SJIS code for Windows; UTF-8 code for Red Hat Enterprise Linux 5 or 6; EUC code for others
--cplusplus	An input file with .c extension is processed as a C++ source file.

[Other Options]

-c	Single compile unit check only. When inputting C source files while the option is not specified, both compile unit check and all linked objects check will be performed. When inputting C++ source files, only single compile unit check is available.
--restart Wide-ranging Detective execution record file	The Wide-ranging Detective function [1] during file check can be interrupted anytime, and the detected indication before interruption can be viewed. When continuing execution, specify this option. In addition, after this option has been specified, the other options and input file will be ignored. The Wide-ranging Detective execution record file is generated in the directory designated by the --project option, or the directory with C source. The generated file name is "project name.link.ini" if the --project option is specified, or "initially input file name.link.ini" if the --project option is not specified. [1]The Wide-ranging Detective function is to track the route of function invocation and indicate the place where is likely to have a failure. In addition, to use the Wide-ranging Detective function, the license of Wide-ranging Detective Option is needed.

[Input File Settings]

Input File	To specify C/C++ source files without compiler error or analytical result file. Multiple file names are allowed. In addition, the file category is determined by the extensions as illustrated below:

Extension	File Category
.c	C source file
.ao	analysis result file
other than above or the file without extension	C++ source file

[Ignored Options]

The following options can be specified, for exclusively used in previous products, it will be ignored even when used. In addition, message descriptions will be displayed in analysis.

-l	For portability and fatal errors. Option provided in previous versions.
-r	For reliability and fatal errors. Option provided in previous versions.
-s	For efficiency and fatal errors. Option provided in previous versions.
-t	For maintainability and fatal errors. Option provided in previous versions.
-n	For fatal errors. Option provided in previous versions.
-g	To output analytical result files (go) to the present directory. This option is provided in previous versions.
-l symbol file	Suppresses messages in which the -l symbol file is not used and referenced. This option also provided in previous version.

1.1.3.1 Identifier File Label List (-F option)

No	label name
	form
	relevant message
1	**[AMBIGUOUS_CHARACTERS] label**
	Easily mistaken characters are documented below. The same character can be registered twice. *Character 1, Character 2 2,Z
	Probably mistaken identifier (pgr0741): If message will be output when the difference of identifiers only arises in easily mistaken characters.
2	**[AREA_ACCESS_FUNCTION] label**

	Information regarding a function that seeks to access memory of specified size shall be written as follows. If there is no[memory parameter position 2] to check, please set it to 0. *function name, memory parameter position 1, memory parameter 2, size parameter position memcpy,1,2,3 memset,1,0,3
	Illegal memory access (pgr0548): If an argument of a registered function accesses beyond the according memory, the message will be output.
3	**[AREA_COMPARE_FUNCTION] label**
	Information regarding a function that compares memory shall be written as illustrated below: *function name, memory parameter position1, ..., parameter position n memcmp,1,2
	Illegal memory comparison (pgr1235, pgr1236): If struct, union and class are compared by registered function, a message will be output. Use of character string comparison function not recommended (pgr1237): If the char type pointer or the char type array is set to the argument that should specify the pointer of the registered function, a message will be output. Datas including the floating type are compared by the function (pgr6007): If datas are compared including the floating type by the registered function, a message will be output.
4	**[AREA_COPY_FUNCTION] label**
	Information of a function to copy from a specified memory to another shall be written follows: * memory copy function name, copy target parameter position, copy source parameter position strcpy,1,2 strcat,1,2
	Illegal memory copy (pgr0532): When the memory beyond the scope is copied by registered function, the message will be output. The return value of environment variable acquisition function is copied onto the array (pgr1164): When the return value of the unpredictable length of environment variable acquisition function is copied directly onto the array, a message will be output. Besides, the following label definitions are necessary. [VARIADIC_STRING_GETENV_FUNCTION]
5	**[AREA_INITIALIZE_FUNCTION] label**
	The information of a function to initialize an area shall be written as follows: *function name, the position of parameter for space, the position of parameter for size memset,1,3 wmemset,1,3

	Lack of initialization of an area (pgr0584): When the area to be initialized using the registered function is smaller than the allocated area, a message will be output. Not initialized of area (pgr1163): When using it without initializing the allocated area, a message will be output. Besides, the following label definitions are necessary. [AREA_RELEASE_FUNCTION] , [RESOURCE_GET_FUNCTION]
6	**[AREA_RELEASE_FUNCTION] label** The information of a function to deallocate dynamic resource shall be written as follows: *memory deallocating function, memory parameter position free,1 Static resource deallocation (pgr0528): If static resource is passed to registered function, the message will be output. Release of pointer pointing to static resources (pgr0833) It will be indicated, when static resource returned by localtime functions etc. is being transferred to registered functions. Not initialized of area (pgr1163): When using it without initializing the allocated area, a message will be output. Besides, the following label definitions are necessary. [AREA_INITIALIZE_FUNCTION] , [RESOURCE_GET_FUNCTION]
7	**[ARG_MUL_FUNCTION] label** The information of a function to multiplies by two arguments shall be written as follows: *function, position of argument for multiplication 1, position of argument for multiplication 2 calloc,1,2 Confirmation leakage of two arguments by which it multiplies in function (pgr1158): When two arguments of the registered function have not been confirmed before the call of the function, the message will be output.
8	**[ASSIGNMENT_FUNCTION] label** Information of the function that updates the content of variables and the location of the variables to be updated are described as follows. * Parameter location of the function that updates the content of variables, and the variables to be updated. fgets,1 memset,1 memcpy,1 memmove,1 strcpy,1 strncpy,1

	Reference of the struct-union member variable before it has been set (pgr0872): Because the registered function is judged as the function that updates variables, this message is disabled. Reliability check of format specification argument (pgr6020): The format argument has not been updated by a function in this label, the message will be output. Besides, the following label definitions are necessary. [FORMAT_CHECK_FUNCTION] , [HARMLESS_FUNCTION]
9	**[ASYNC_SIGNAL_SAFE] label**
	The information of a function that are asynchronous-signal-safe shall be written as follows: *function name _Exit _exit
	Use functions that are not asynchronous safe (pgr6052): If you are using a function that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [ASYNC_SIGNAL_UNSAFE], [SIGNAL_FUNCTION] Use variables that are not asynchronous safe (pgr6053): If you are using a variable that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [SIGNAL_ATOMIC_TYPE], [SIGNAL_FUNCTION]
10	**[ASYNC_SIGNAL_UNSAFE] label**
	The information of a function that cannot be safely used asynchronously shall be written as follows: *function name fopen fopen_s
	Use functions that are not asynchronous safe (pgr6050): If you are using a function that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [SIGNAL_FUNCTION] Use functions that are not asynchronous safe (pgr6052): If you are using a function that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [ASYNC_SIGNAL_SAFE], [SIGNAL_FUNCTION]
11	**[AUTO_VAR_ACCESS_FUNCTION] label**

	The information of a function in which auto variable cannot be used as argument shall be written as follows: The same information cannot be registered twice. *function name, auto variable parameter position 1, ..., auto variable parameter position putenv,1
	To check whether auto variable is used as argument. (pgr0564): If an auto variable is passed to registered function as an argument, the message will be output.
12	**[AVAILABLE_DIRECT_ACCESS_AREA] label**
	The information of a direct access active area in address shall be written as follows: The value is specified within the range of -9223372036854775808 - 9223372036854775807. *direct access active area start address, direct access active area end address 0x01,0xFF -128,128 0xFFFF
	Comparison between pointer type and integer constant (pgr0857): When an address outside the registered direct access active area is compared with the integer constant, the message will be output. Comparison between pointer type and integer constant (pgr0858): When an address in the registered direct access active area is compared with the integer constant, the message will be output.
13	**[BEFOREHAND_FUNCTION] label**
	The information on presetting function that should be called ahead of specific function shall be written as follows: *function name, presetting function name random,srandom rand,srand
	The function call is not defended (pgr1151): When the presetting function has not been called before the registered function is called, the message will be output.
14	**[BLOCKING_FUNCTION] label**
	Information of exclusively controlled function shall be written as follows: * blocking function name recv recvfrom
	Call of locked Blocking function (pgr6006): If the output content exceeds the space by the parameter of registered function, the message will be output. Besides, the following label definitions are necessary. [EXCLUSION_LOCK_UNLOCK_FUNCTION]

15	**[BUFFER_FORMATTED_IO_FUNCTIONS] label**
	The information of a function to output the format string shall be written as follows. The same information cannot be registered twice. *function name, memory parameter position, format parameter position sprintf,1,2
	Illegal write to memory (pgr0570): If the output content exceeds the space by the parameter of registered function, the message will be output.
16	**[BYTE_DATA_RECV_FUNCTION] label**
	The Name of the function for the data reception shall be written as follows. *name of function for reception, data argument position recv,2 recvfrom,2
	Untranslation of byte order (pgr6010): When the byte order has not been converted before the data arguments of the function is used, the message will be output. Besides, the following label definitions are necessary. [BYTE_DATA_SEND_FUNCTION], [BYTE_ORDER_CONVERT_FUNCTION]
17	**[BYTE_DATA_SEND_FUNCTION] label**
	The Name of the function for the data transmission shall be written as follows. *name of function for transmission, data argument position send,2 sendto,2
	Untranslation of byte order (pgr6010): When the byte order has not been converted before the data arguments of the function is used, the message will be output. Besides, the following label definitions are necessary. [BYTE_DATA_RECV_FUNCTION], [BYTE_ORDER_CONVERT_FUNCTION]
18	**[BYTE_ORDER_CONVERT_FUNCTION] label**
	The Name of the byte order convert function shall be written as follows. *byte order convert functions name from Network to host, byte order convert functions name from host to Network, position of return value(0:return, 1:argument 1, 2:argument 2, …), data type name ntohl,htonl,0,long ntohs,htons,0,short
	Untranslation of byte order (pgr6010): When the byte order has not been converted before the data arguments of the function is used, the message will be output. Besides, the following label definitions are necessary. [BYTE_DATA_SEND_FUNCTION], [BYTE_ORDER_CONVERT_FUNCTION]
19	**[CATCH_IN_MAIN_FUNCTIONS] label**

	The name of a function that catches the exception shall be written as follows. *function name main
	Exception check leak (pgr2241): If the catch handler of registered function is not ended with catch(...), the message will be output.
20	**[CHANGE_NEW_DELETE] label**
	The name of a function which can be substituted with new or delete operator shall be written as follows: *function name 1, ..., function name n malloc,free
	Usage of function which can be substituted with new or delete operator(pgr1006): If a registered function is invoked, message will be output.
21	**[CHANGE_PROCESS_PERMISSION_FUNCTION] label**
	The name of functions that perform root privilege operations and information regarding a function that generates child processes shall be written as follows: The same information can be registered twice. * function name, the name of the function that generates child processes, return value to child process from the child process generating function. setuid,fork,0 setgid,fork,0
	To check whether the privilege function is invoked in child process startup (pgr0566) When the privilege function is not invoked before the function generated in the registered child process, the return value of the function generated in the registered child process is [return value to child process]. The privilege function will then not be invoked and the message will be output.
22	**[CHARACTER_INPUT_FUNCTION] label**
	The name of the function that is to receive input characters shall be written as follows. The same information can be registered twice. *Function name fgetc
	Whether the extraction on the review of space alignment is checked (pgr0563): If the registered function is invoked in a loop, the message will be output.
23	**[CHECK_FILES_ACCESSIBILITY_FUNCTION] label**
	Before opening a file, information regarding recommended and not recommended file state checking functions shall be written as follows. [Function name 3 to be opened] cannot be registered twice. *recommended function name 1, parameter position of file name within function name 1, not recommended function name2, parameter position of file name within function name 2, function name 3 to be opened, parameter position of file name within function name 3 lstat,1,access,1,open,1

	To check whether the file to be opened is symbolic link or not (pgr0569): If the not recommended function is invoked before the registered [Function name 3 to be opened], and the recommended function is not invoked in the latest front of the registered [Function name 3 to be opened], the message will be output.
24	**[CHECK_FILES_STATE_FUNCTION] label** When opening a file, the information of the recommended file state checking function shall be written as follows. If [the location of file stream/file descriptor of function name 2] is 0, it will be processed as a return value. If it is not, it will be processed as the parameter location. The same information can be registered twice. *Before opening file, function name 1 to check file state, parameter position of function name 1, open function name 2, parameter position of function 2, parameter position of file name within function name 2, position of file stream/file descriptor of function name 2, file stream is converted to function name 3 of descriptor (omissible), parameter position of file stream within function name 3 (omissible), after the file is opened, function name 4, parameter position of file name within function name 4. lstat,1,open,1,0,,0,fstat,1 lstat,1,freopen,1,3,fileno,1,fstat,1 lstat,1,freopen,1,0,fileno,1,fstat,1 Check whether the file to be opened is symbolic link or not (pgr0567): Before the registered open function is invoked, the file state checking function is not invoked; or after the registered open function is invoked, and the file state checking function is not invoked, the message will be output.
25	**[CHECK_HEADER] label** To check whether a header file is processed with #include. The same information can be registered twice. * header file name 1,... , header file name n locale.h,debug.h time.h To check whether a header file is processed with #include (pgr0258): If the registered header file name is processed with #include, the message will be output.
26	**[CHECK_IDENTIFIER] label** When to check if the identifier is used or not, the identifier shall be written as illustrated below: *identifier name scanf errno Illegal use of identifier (pgr0252-pgr0256): If the registered identifier is used, the message will be output.
27	**[CHECK_KEYWORD] label**

	The reserved word that cannot be used for macro name shall be written as follows: *reserved word int while The reserved word is used for macro name (pgr0804): If macro name of registered reserved word and same name is defined, a message will be output.
28	[CHECK_LITERAL_TYPE] label The type of literal and the Message output setting that prohibits using shall be written as follows: [Specifiable literal type] CHARACTER : Character (The escape sequence character is excluded) STRING : String ESCAPE : Escape sequence character (\n and \0, etc.) INTEGER : Integer other than 0 or 1 FLOAT : Floating point number ZERO : Integer constant 0 ONE : Integer constant 1 OTHER : Literal other than the above (Complex/UserDefined/nullptr etc.) SOURCE_ONLY: Literal in source for check (The literal of the header file is excluded) [Message output setting] ALL : All the detection parts are output. C1 : Only one place is output by the source file(Each literal type). (Default) (*) The message output setting cannot be done to "SOURCE_ONLY". *Literal type,Message output setting(omissible) INTEGER,C1 Use of literal (pgr1395): If the literal of the registered type is used, a message will be output.
29	[CHECK_OPERATIONS_ON_FILES_FUNCTION] label Information regarding a symbolic link expansion function (converted to the file path without symboliclink) and a function to write in files shall be written as illustrated below: [name of the function to write in file] cannot be registered twice. *symboliclink expansion function name, the parameter position of file path after symboliclink is expanded, the name of the input file receiving function , the parameter position of file path of function to write in file realpath,2,stat,1 To check whether the file to be processed is symbolic link or not (pgr0565): When the file path of argument in registered function is not the file path after symbolic link is expanded, the message will be output.

30	**[CHECK_SOURCE_CODE_EXIST_COMMENT] label**

	The strings to determine the code part shall be written as follows:
	*strings
	if(
	while(
	return;

	Comment out in code part (pgr0849): If the registered strings appears in the comment, a message will be output.
31	**[CLOSE_FILE_FUNCTION] label**
	The name of close file function shall be written as follows: *function name, parameter position of file pointer fclose,1
	Multiple open file (pgr0887): When the same file is opened multiple times by the function which is registered in [OPEN_FILE_FUNCTION] label, the message will be output. Besides, the following label definitions are necessary. [OPEN_FILE_FUNCTION], [OPEN_FILE_READ_MODE], [OPEN_FILE_WRITE_MODE] Write into the file which is opened with read mode (pgr0888): When writing into the file which is opened with read mode by the function which is registered in [OPEN_FILE_FUNCTION] label, the message will be output. Besides, the following label definitions are necessary. [OPEN_FILE_FUNCTION], [WRITE_FILE_FUNCTION], [OPEN_FILE_READ_MODE] The file has not been shut before the process is created (pgr1153): When the opening file has not been closed before the process is created with the registered function, the message will be output. Besides, the following label definitions are necessary. [OPEN_FILE_FUNCTION], [OPEN_FILE_FUNCTION]
32	**[COMMAND_EXECUTE_FUNCTION] label**
	Information regarding functions that can call programs and information regarding strings that cannot be passed to a program shall be written as follows. If -n appears in [the end of parameters to program], it means it will end at the parameter that is 'n' prior. For example, -1 means ending with the last parameter, -2 means ending with parameter one prior to the last parameter. If [the end of parameters to program] is omitted, the range check will not be performed, only [the start of parameters to program] will be checked. The same information cannot be registered twice. *The function name, the parameter position for passing the program name, the program name, the starting position of the parameters that are passed to the program, the ending position of the parameters that are passed to the program, dangerous string 1, ..., dangerous string n execle,1,sendmail,2,-2,| execve,1,sendmail,2,-1,|,- execle,1,sendmail,2,,|,-

	To check the string that cannot be passed to the program (pgr0561, pgr0582): If the argument of the registered function has [program name] and the parameter that is passed to the program starts with a [dangerous string], a message will be output.
33	**[COMPATIBLE_CHECK_FUNCTION] label** Information on the function that is compared and copied between different types shall be written as follows: *function name, argument to be confirmed 1, argument to be confirmed 2 memcpy,1,2 memmove,1,2 memcmp,1,2 Area comparison and copy by different type (pgr1239): If the comparison and the copy between different types are done by using the registered function, a message will be output.
34	**[CONDITION_WAITING_FUNCTION] label** The information of the function that sets the wait condition of the thread of control shall be written as follows: *name of function that stands by until satisfying, position of conditional variable(1:argument 1, 2:argument 2, …), position of mutex(1:argument 1, 2:argument 2, …) pthread_cond_wait,1,2 pthread_cond_timedwait,1,2 Call of thread of control wait condition setting function in different conditional variable and mutex (pgr6012): When the function is called by combining a specific conditional variable and the mutex, the message will be output.
35	**[CONSTANT_ARGUMENT_NG_FUNCTION] label** The information on function that should not specify constant for argument shall be written as follows: *function name, argument position in which constant should not be specified srandom,1 srand,1 Constant specification for specific argument (pgr1152): When the constant is specified for a specific argument of the registered function, the message will be output.
36	**[DIRECT_INPUT_OUTPUT_FUNCTION] label** The information of a function to access [data length * data number] space shall be written as follows. The same information cannot be registered twice. *function name, parameter position of memory, parameter position of data length, parameter position of data number. fwrite,1,2,3

	Illegal memory access (pgr0554): If the memory size by the argument of the registered function is smaller than data length * data number, the message will be output.
37	**[EOF_CHECK_FUNCTION] label** The information of a function to check EOF shall be written as follows. *function name, parameter position of stream fgetc,1 Mismatch between return type of function that checks EOF and type of stored variable (pgr6021): The return type of the function in this label does not match the type of the variable that stores the return value, the message will be output. Use without checking status of stream (pgr6022): When you use a stream without checking the state of the stream, the message will be output. Besides, the following label definitions are necessary. [FILE_STREAM_CHECK_FUNCTION]
38	**[ERRNO_SET_FUNCTION] label** The Information of library function in which errno is set shall be written as follows. *function name, function type(1:Function that should confirm errno. , 2:The Confirmation of errno is a function that is optional. , 3:Function that does not confirm errno.), Variable like errno(default: errno), Initializing value of errno.(default: 0) alphasort,1 acos,1 Uninitialization of errno (pgr6015): When the initialization of errno has not been done before it is called for the function that sets errno, the message will be output.
39	**[EXCLUDE_IDENTIFIER_PATTERN] label** The Identifier that controls check shall be written as follows. *Agile$^+$ Relief message number that controls check (Only pgr6000 and pgr6005 are specifiable), Identifier (Specification by arbitrary string by .* is possible.) pgr6000,pthread_mutex_lock <- "Pthread_mutex_lock" corresponds. pgr6000,.*lock <- The end is an identifier of "lock". mylock etc. pgr6005,.*mutex.* <- Identifier including "Mutex". p_mutex_func， mutex etc.

	It is unlocked though the mutex is not locked (pgr6000): When the unlock function is called before the lock function of same mutex is called, the message will be output. Annulment of mutex lock/not unlocked (pgr6001): When neither the lock function, the unlock function nor the initialization function have been called before the annulment function of same mutex is called, the message will be output. The sequential order where the lock/unlock function is called is different (pgr6004): When the sequential order where the lock function and the unlock function of same mutex are called is different, the message will be output. The call frequency of the lock/unlock function is different (pgr6005): When the call of the lock function is more than the call of the unlock function, the message will be output.
40	**[EXCLUSION_LOCK_UNLOCK_FUNCTION] label** Information of exclusively controlled function shall be written as follows: *lock function name, unlock function name, init function name, destroy function name, position of lock object of unlock function (0:return, 1:argument 1, 2:argument 2, …), position of lock object of init function (0:return, 1:argument 1, 2:argument 2, …), position of lock object of destroy function (0:return, 1:argument 1, 2:argument 2, …) pthread_mutex_lock,pthread_mutex_unlock,pthread_mutex_init,pthread_mutex_destroy,1,1,1,1 sem_wait,sem_post When only the lock function and the unlock function are specified, only pgr1160 is checked.

	Check on exclusive control in multi-thread function (pgr1160): When you use static variable and the external variable without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [THREAD_CREATE_FUNCTION] Check on exclusive control in multi-thread function (pgr1256): When you use bit field without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [THREAD_CREATE_FUNCTION], [MULTI_THREAD_FUNCTION], [SIGNAL_FUNCTION] Check on exclusive control in multi-thread function (pgr1257): When you use variable qualified with volatile without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [THREAD_CREATE_FUNCTION], [MULTI_THREAD_FUNCTION], [SIGNAL_FUNCTION] It is unlocked though the mutex is not locked (pgr6000): When the unlock function is called before the lock function of same mutex is called, the message will be output. Annulment of mutex lock/not unlocked (pgr6001): When neither the lock function, the unlock function nor the initialization function have been called before the annulment function of same mutex is called, the message will be output. The sequential order where the lock/unlock function is called is different (pgr6004): When the sequential order where the lock function and the unlock function of same mutex are called is different, the message will be output. The call frequency of the lock/unlock function is different (pgr6005): When the call of the lock function is more than the call of the unlock function, the message will be output. Call of locked blocking of data, locking of data function (pgr6006): When the function with the possibility of blocking it is called while the lock function locks, the message will be output. Besides, the following label definitions are necessary. [BLOCKING_FUNCTION] Use of thread-unsafe function (pgr6008): When a thread-unsafe function is called in the multi-thread function, the message will be output. Besides, the following label definitions are necessary. [MULTI_THREAD_FUNCTION], [THREAD_CREATE_FUNCTION], [THREAD_UNSAFE_FUNCTION]
41	**[EXIT] label**

	The name of a function not return to the invoke side shall be written as follows: *function name exit Referenced before assignment, 0 address is referenced etc. (pgr0060, pgr0520, pgr0541, pgr0689): The registered function call is regarded not return, thus the incorrect message can be suppressed. Miss of resource deallocation(pgr0524), unexecuted statement (pgr0667): The registered function call is regarded not return, thus more accurate message will be output.
42	**[EXIT_HANDLER_NG_FUNCTION] label** The information of a function that should not be registered in program termination handler shall be written as follows: *function name Exit Double call of program end handler (pgr6016): When the program termination handler is called twice, the message will be output. Besides, the following label definitions are necessary. [EXIT_REGISTER_FUNCTION]
43	**[EXIT_REGISTER_FUNCTION] label** The information of a function to register program termination handler shall be written as follows: *function name, argument position of program termination handler atexit,1 Double call of program end handler (pgr6016): When the program termination handler is called twice, the message will be output. Besides, the following label definitions are necessary. [EXIT_HANDLER_NG_FUNCTION]
44	**[EXTERNAL_LINKAGE_IDENTIFIER_SIGNIFICANT_INITIAL_CHARACTERS] label** The string length that can identify the identifier with the external linkage shall be written as follows: *C Version, length C90,6 C99,31 C11,31 String length check of identifier with external linkage (pgr0814): If the length of the identifier with the external linkage exceeds the registered string length of each ANSI-C version, a message will be output.
45	**[FILE_STREAM_CHECK_FUNCTION] label**

	The information of a function to check file stream shall be written as follows: *function name, parameter position of stream feof,1 ferror,1
	Use without checking status of stream (pgr6022): When you use a stream without checking the state of the stream, the message will be output. Besides, the following label definitions are necessary. [EOF_CHECK_FUNCTION]
46	**[FORMAT_CHECK_FUNCTION] label**
	The information of a function with format string shall be written as follows. The same information cannot be registered twice. *function name, parameter position of format string snprintf,3 scanf,1
	To check whether the width of string is specified in format string (pgr0556, pgr0580): If the width of string is not specified in the arguments of the registered function, the message will be output. Reliability check of format specification argument (pgr6020): The format argument has not been updated by a function in this label, the message will be output. Besides, the following label definitions are necessary. [ASSIGNMENT_FUNCTION] , [HARMLESS_FUNCTION]
47	**[FORMATTED_OUTPUT_CONVERSION_FUNCTION] label**
	Information of functions with format shall be written as follows. Repeated information cannot be registered. * Parameter position of function name and format printf,1 fprintf,2 sprintf,2
	Type check for format conversion specifier and argument (pgr0822) It will be indicated, when the type of format conversion specifier recorded in the parameter of registered functions is not in accordance with the argument. Number check for format conversion specifier and argument (pgr0823) It will be indicated, when the number of format conversion specifier recorded in the parameter of registered functions is not in accordance with the argument.
48	**[FUNCTION_ASSUME_PARAMETER_TYPE] label**

	The function that becomes undefined operation when the value not expressible by the type assumed to be an ideal to the argument of the function is used shall be written as follows: *function name, parameter position, type assumed to be ideal isalpha,1,unsigned char
	The value not expressible by the type assumed to be an ideal to the functional argument is used (pgr0820): If the value not expressible by the type assumed to be an ideal is used as an argument of the registered function, a message will be output.
49	**[HARMLESS_FUNCTION] label**
	The information of a function to function to render harmless shall be written as follows: *function name, parameter position, fmtchk,1
	Reliability check of format specification argument (pgr6020): The format argument has not been updated by a function in this label, the message will be output. Besides, the following label definitions are necessary. [ASSIGNMENT_FUNCTION] , [FORMAT_CHECK_FUNCTION]
50	**[IDENTIFIER_LENGTH] label**
	If want to modify the valid length of identifier of 31 characters, the valid length shall be written as follows. The same information cannot be registered twice. *the valid length of identifier 20
	The character number of identifier is longer than the valid length. (pgr0703): When the identifier is longer than the registered valid length, the message will be output.
51	**[IDENTIFIER_SIGNIFICANT_INITIAL_CHARACTERS] label**
	The string length that can identify the identifier without the external linkage shall be written as follows: *C Version, length C90,31 C99,63 C11,63
	String length check of identifier without external linkage (pgr0816): If the length of the identifier without the external linkage exceeds the registered string length of each ANSI-C version, a message will be output. String length check of macro name (pgr0817): If the string length of macro name exceeds the registered string length of each ANSI-C version, a message will be output.
52	**[INSPECTION_LEVEL_ADJUSTMENT] label**

	The check level shall be written as follows: *check level (pgrid=flag) pgr2216=normal
	NULL check leak (pgr0520): Register the Agile$^+$ Relief indication level if the return value type of the function in the [NULL_RETURN_FUNCTION] label is unknown. If you register "pgr0520=normal", Agile$^+$ Relief will not point that out. If you register "pgr0520=checkfunccallexp", Agile$^+$ Relief points out that condition. *If this label is not registered, the behavior is the same as registering "pgr0520=normal". Besides, the following label definitions are necessary. [NULL_RETURN_FUNCTION] Use of macro function (pgr2216): Registers the Agile$^+$ Relief indication level if the macro function substitution expression includes the # and # # operators. If you register "pgr2216=normal", Agile$^+$ Relief will not point that out. If you register "pgr2216=excludeNumberSign", Agile$^+$ Relief points out that condition. *If this label is not registered, the behavior is the same as registering "pgr2216=normal". Besides, the following label definitions are necessary. [POSITION_RELATED_IDENTIFIER] Checking for copy beyond array size (pgr6030): Registers the Agile$^+$ Relief indication level if the copy size is unknown or the destination array size is unknown. If you register "pgr6030=normal", Agile$^+$ Relief will not point that out. If you register "pgr6030=verbose", Agile$^+$ Relief points out that condition. *If this label is not registered, the behavior is the same as registering "pgr6030=normal". Besides, the following label definitions are necessary. [MEMORY_COPY_CHECK_FUNCTION] Checking for copy beyond area size (pgr6031): Registers the Agile$^+$ Relief indication level if the copy area size is unknown or the destination area size is unknown. If you register "pgr6031=normal", Agile$^+$ Relief will not point that out. If you register "pgr6031=verbose", Agile$^+$ Relief points out that condition. *If this label is not registered, the behavior is the same as registering "pgr6031=normal". Besides, the following label definitions are necessary. [MEMORY_COPY_CHECK_FUNCTION], [STRING_BUFFER_MALLOC_CHECK_FUNCTION]
53	**[MACRO_RESERVED_IDENTIFIER] label**

	The system external name used shall be written as follows: This label is only valid for pgr0786, [RESERVED_LIBRARY_IDENTIFIER]label is valid for both pgr0680 and pgr0786, [RESERVED_IDENTIFIER]label is only valid for pgr0680. The same information can be registered twice. *external name fucabc
	Illegal use of system external name (pgr0786): If the declaration and definition of the same name with the registered external name, the message will be output.
54	**[MEM_NO_POD_FUNCTIONS] label**
	A function that does not pass non POD type as an argument in the function call. The function information shall be written as follows. * function name, the position of parameter for copy destination/source memcpy,1 memcpy,2 memcmp,1 memcmp,2
	Illegal use of non POD type (pgr2238): If the non POD type is passed to the specified argument of the registered function, the message will be output.
55	**[MODIFY_ENV_FUNCTION] label**
	A function that changes environmental variable shall be written as follows: * function name setenv putenv
	Refer to the third argument of main function (pgr6009): If the registered function which changes environmental variable refer to the third argument of main function, the message will be output.
56	**[MEMORY_COPY_CHECK_FUNCTION]label**
	*Memory copy function name, parameter position of copying target location, parameter position of copy source, parameter position of copy SIZE memcpy,1,2,3 memmove,1,2,3

	No copy terminal character (pgr0832) In the registered function, when copying from an array in larger size to an array in smaller size, it will be indicated if the terminal character is not included. Checking for copy beyond array size (pgr6030): Registers the Agile^{+} Relief indication level if the copy size is unknown or the destination array size is unknown. Besides, the following label definitions are necessary. [INSPECTION_LEVEL_ADJUSTMENT] Checking for copy beyond area size (pgr6031): Registers the Agile^{+} Relief indication level if the copy area size is unknown or the destination area size is unknown. Besides, the following label definitions are necessary. [STRING_BUFFER_MALLOC_CHECK_FUNCTION]
57	**[MORE_SECURE_FILESYSTEM_FUNCTION] label**
	Information of function to identify file by using file name shall be written as follows: * function name, the response method chown,Please consider using "fchown" which can avoid the file-related race condition.
	Use of function to identify file by using file name (pgr1171): When you use the registered function, a message will be output.
58	**[MORE_SECURE_FUNCTION] label**
	Information of non-recommended function or out-of-date function shall be written as follows: * non-recommended function name, recommended function names atoi,"strtol" or "strtoul" * '"strtol" or "strtoul"' is considered to be 'recommended function names'.
	Use of non-recommended function or out-of-date function (pgr1170): When you use the registered function, a message will be output.
59	**[MULTI_THREAD_FUNCTION] label**
	Multithread function name shall be written as follows: The same information can be registered twice. *function name getbuf

	By using multithread function, to update external variable or static variable (pgr0578): If updating external variable or static variable in the registered function definition, the message will be output. Check on exclusive control in multi-thread function (pgr1256): When you use bit field without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [THREAD_CREATE_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION], [SIGNAL_FUNCTION] Check on exclusive control in multi-thread function (pgr1257): When you use variable qualified with volatile without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [THREAD_CREATE_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION], [SIGNAL_FUNCTION] Using the thread-unsafe function (pgr6008): When the registered thread-unsafe function is called in multi tread function, the message will be output. Moreover, the following label is required. [MULTI_THREAD_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION], [THREAD_CREATE_FUNCTION]
60	**[MUTEX_SEMAPHORE_TYPE] label** The data type used as mutex shall be written as follows: * data type of variable pthread_mutex_t mtx_t Using mutex which is neither external variable nor static variable (pgr6003): if the registered data type is used as mutex which is neither external variable nor static variable, the message will be output.
61	**[NOCHECK_NOUSE_IDENTIFIER] label** The variable name or function name that is not required to check whether the registered variable or function is used shall be written as follows: *function name or variable name main Unused variable or function (pgr0690, pgr0696): If the registered variable or function is regarded as used, this message will be suppressed.
62	**[NON_VALUEOFSIZEOF_NG_FUNCTION] label** The information of a function in which a parameter is defined as size shall be written as follows: *function name, parameter position 1 of size, ..., parameter position n of size malloc,1

	The function call in which an argument is defined as size, i.e.: the standard library function such as sizeof or strlen, is not used as an argument. (pgr0242, pgr0243): If the value is not evaluated by the standard library function, such as sizeof or strlen, it is passed to the argument of the registered function and a message will be output.
63	**[NOT_EXPAND_MACRO] label**

	The macro function not expected to expand shall be written as follows. When [macro function name] recurs, only the first value is valid.
	*macro function name, replacement string for macro definition (omissible)
	getchar,int getchar(void);
	isalpha,int isalpha(int);

	When several messages unnecessary for macro expansion are output, please check the validity of the messages. If the definition of the registered macro function name is replaced with [the replacement string for macro definition], macro expansion will not be performed. When [the replacement string for macro definition] is omitted, the defined line is regarded as an empty line.
64	**[NULL_POINTER_NG_FUNCTION] label**
	A function cannot pass 0 as an argument in the function call. The function information shall be written as follows: *function name, parameter position unable to pass 0, ..., parameter position n atoi,1
	0 address is referenced (pgr0060, pgr0520, pgr0541, pgr0689): If 0 is passed to the specified argument of the registered function a message will be output.
65	**[NULL_RETURN_FUNCTION] label**
	The name of a function may return NULL and shall be written as follows: *function name calloc
	NULL check leak (pgr0520): If the return value of the registered function is used before being checked whether it is NULL or not by the *if* statement, a message will be output.
66	**[OMITTED_ARGUMENT_NG_FUNCTION] label**
	The information on function that should not omit specification of argument shall be written as follows: *function name, argument position in which specification should not be omitted open,3
	Argument omission when function is called (pgr1155): When the argument at a specified position of the registered function is omitted when the function is called, a message will be output.
67	**[OPEN_FILE_FUNCTION] label**

	The name of open file function shall be written as follows: *function name, parameter position of file name, parameter position of operator mode fopen,1,2
	Multiple open file (pgr0887): When the same file is opened multiple times by the registered function, the message will be output. Besides, the following label definitions are necessary. [CLOSE_FILE_FUNCTION], [OPEN_FILE_READ_MODE], [OPEN_FILE_WRITE_MODE] Write into the file which is opened with read mode (pgr0888): When writing into the file which is opened with read mode by the registered function, the message will be output. Besides, the following label definitions are necessary. [CLOSE_FILE_FUNCTION], [WRITE_FILE_FUNCTION], [OPEN_FILE_READ_MODE] The file has not been shut before the process is created (pgr1153): When the opening file has not been closed before the process is created with the registered function, the message will be output. Besides, the following label definitions are necessary. [CLOSE_FILE_FUNCTION], [PROCESS_FUNCTION]
68	**[OPEN_FILE_READ_MODE] label**
	Access mode when opening file with read mode shall be written as follows: *identifier of read mode "r" "rt"
	Multiple open file (pgr0887): When the same file is opened multiple times by the function which is registered in [OPEN_FILE_FUNCTION] label, the message will be output. Besides, the following label definitions are necessary. [OPEN_FILE_FUNCTION], [CLOSE_FILE_FUNCTION], [OPEN_FILE_WRITE_MODE] Write into the file which is opened with read mode (pgr0888): When writing into the file which is opened with registered read mode by the function which is registered in [OPEN_FILE_FUNCTION] label, the message will be output. Besides, the following label definitions are necessary. [OPEN_FILE_FUNCTION], [CLOSE_FILE_FUNCTION], [WRITE_FILE_FUNCTION]
69	**[OPEN_FILE_WRITE_MODE] label**
	When opening file with read mode, the access mode shall be written as follows: *identifier of write mode "w" "wt"

	Multiple open file (pgr0887): When the same file is opened multiple times by the function which is registered in [OPEN_FILE_FUNCTION] label, the message will be output. Besides, the following label definitions are necessary. [OPEN_FILE_FUNCTION], [CLOSE_FILE_FUNCTION], [OPEN_FILE_READ_MODE]
70	**[PAIR_NEW_DELETE] label**
	The name of open file function shall be written as follows: * function name for allocation resources and for deallocation resources malloc,free
	Illegal allocate(delete) object(pgr2237): If the object allocated by the new operator (or the object released by the delete operator) is passed to the specified argument of the registered function, the message will be output.
71	**[PARA_SIZE_CHECK_FUNCTION] label**
	The information of a function with a limited memory size process shall be written as follows: The same information can be registered twice. *function name, parameter location of space, upper limit size realpath,2,1024
	Illegal write to memory (pgr0568): If the space size specified by the parameter of the registered function doesn't reach the upper size limit, the message will be output.
72	**[PARALLEL_FUNCTION] label**
	Parallel function name shall be written as follows. All registered [function name] are regarded as parallel functions, and cannot be divided. *function name1, ..., function name n func1,func2
	Common variable is not volatile-qualified (pgr0536, pgr0688): If the variable is commonly used and not volatile-qualified in the registered function, a message will be output.
73	**[PASSWD_IDENTIFIER] label**
	Variable identifier that stores password shall be written as follows: *variable name pwd
	Illegal treatment of sensitive data (pgr1162): When the comparison of password storage variable and string literal is done or string literal to password storage variable is substituted, a message will be output. Besides, the following label definitions are necessary. [STRING_COMPARE_FUNCTION]
74	**[POINT_STATIC_AREA_FUNCTION] label**

	The function name that returns the pointer pointing to static memory shall be written as follows. *function name ctime localtime
	Release of pointer pointing to static resources (pgr0833): It will be indicated, when static resource returned by localtime functions etc. is being transferred to registered functions.
75	**[POINTER_CLASS] label**
	The smart pointer shall be written as follows. *pointer name auto_ptr shared_ptr unique_ptr observer_ptr
	NULL check leak (pgr0520): Besides, the following label definitions are necessary. [NULL_RETURN_FUNCTION], [INSPECTION_LEVEL_ADJUSTMENT]
76	**[POSITION_RELATED_IDENTIFIER] label**
	The Identifier included in substitution expression of macro function shall be written as follows. *Identifier __FILE__ __LINE__time
	Use of macro function (pgr2216): This is indicated when the registered identifier is contained in a macro function substitution expression.
77	**[PRIVILEGE_REQUIRE_FUNCTION] label**
	Function identifier that needs root privilege for run time shall be written as follows: *function name bind
	Privilege modify leakage before generate of process (pgr1161): When the authority will not be changed by the time the process generating function is executed after the function for which the privilege is necessary for execution is executed, a message will be output. Besides, the following label definitions are necessary. [PRIVILEGE_SET_FUNCTION], [PROCESS_FUNCTION]
78	**[PRIVILEGE_REVOCATION_CONFIRM] label**

	Function identifier to which privilege is granted shall be written as follows: * function name, setuid,
	Use of Function to which privilege is granted (pgr1174): When you use the registered function, a message will be output.
79	**[PRIVILEGE_REVOCATION_ORDER_CONFIRM] label**
	Function identifier to which privilege is granted when it is root unauthorized than run time shall be written as follows: * function name, setgid,
	Use of Function identifier to which privilege is granted when it is root unauthorized than run time (pgr1173): When you use the registered function, a message will be output.
80	**[PRIVILEGE_SET_FUNCTION] label**
	Function identifier to which privilege is granted shall be written as follows: * function name setuid
	Privilege modify leakage before generate of process (pgr1161): When the authority will not be changed by the time the process generating function is executed after the function for which the privilege is necessary for execution is executed, a message will be output. Besides, the following label definitions are necessary. [PRIVILEGE_REQUIRE_FUNCTION], [PROCESS_FUNCTION]
81	**[PROCESS_FUNCTION] label**
	The information on function that creates process shall be written as follows. *function name system execl
	The file has not been shut before the process is created (pgr1153): When the opening file has not been closed before the process is created with the registered function, the message will be output. Besides, the following label definitions are necessary. [CLOSE_FILE_FUNCTION], [OPEN_FILE_FUNCTION] Privilege modify leakage before generate of process (pgr1161): When the authority will not be changed by the time the process generating function is executed after the function for which the privilege is necessary for execution is executed, a message will be output. Besides, the following label definitions are necessary. [PRIVILEGE_REQUIRE_FUNCTION], [PRIVILEGE_SET_FUNCTION]
82	**[PTHREAD_SET_CANCEL_TYPE] label**

	The function that sets operation when thread is canceled shall be written as follows. *function name, the positon of position, the value of parameter pthread_setcanceltype,1,PTHREAD_CANCEL_DEFERRED
	Cancellation of asynchronous thread (pgr6011): If the setting which delays until the cancel point is applied to the registered function which sets operation when asynchronous thread is canceled, the message will be output.
83	**[RELATIVE_PATH_FUNCTION] label**
	The information of a function that can execute a program and the path for a program shall be written as follows. The same information can be registered twice. *function name, parameter position of program name, necessary string 1, ..., necessary string n execle,1,/
	Check the path of the program (pgr0562, pgr0583): If the program name is used as the parameter of the registered function and is not a string beginning with [necessary string], a message will be output.
84	**[RESERVED_IDENTIFIER] label**
	This label is only valid for pgr0680. [RESERVED_LIBRARY_IDENTIFIER]label is valid for both pgr0680 and pgr0786. [MACRO_RESERVED_IDENTIFIER]label is only valid for pgr0786. The same information cannot be registered twice. The system external name shall be written as follows: *external name EOF NULL
	Illegal use of system external name (pgr0680): If the declaration or definition have the same name as the registered external name, a message will be output.
85	**[RESERVED_LIBRARY_IDENTIFIER] label**
	The system external name shall be written as follows. This label is valid for both pgr0680 and pgr0786. [MACRO_RESERVED_IDENTIFIER] label is only valid for pgr0786. [RESERVED_IDENTIFIER]label is only valid for pgr0680. The same information cannot be registered twice. *external name exit printf
	Illegal use of system external name (pgr0680, pgr0786): If the declaration or definition is of the same name as the registered external name, a message will be output.

86	**[RESOURCE_GET_FUNCTION] label** According to the following information, describe the resource allocation function and the allocated size. *Allocation function name, argument position of resource size of allocation function malloc,1 realloc,1 Out-of-range access of array (pgr5021) Indicate when out-of-range access occurred for the range returned by registered allocation function. Not initialized of area (pgr1163) When using it without initializing the allocated area, a message will be output. Besides, the following label definitions are necessary. [AREA_INITIALIZE_FUNCTION], [AREA_RELEASE_FUNCTION]
87	**[RESOURCE_PAIR_FUNCTION] label** The information of a function to allocate resources and a function to deallocate resources shall be written as follows. If the function is without [name of function to reallocate resource] omit it and then [resource parameter position of function to reallocate resource] is set to 0. Set the function allocating the resources to the return value for [name of function to reallocate resource] and [resource parameter position of function to reallocate resource]. *the name of function to allocate resource, the name of function to reallocate resource, resource parameter position of function to reallocate resource, the name of function to deallocate resource, resource parameter position of function to deallocate resource malloc,realloc,1,free,1 tmpfile,,0,fclose,1 Use deallocated resource (pgr0522): If the registered function to deallocate resources is invoked and then the deallocated resources are used, a message will be output. Resource deallocation leak (pgr0524): If the registered function to allocate resources, the function to reallocate resources and the function to deallocate resources are not of the same group, a message will be output. Out-of-range access of array (pgr5021) Indicate when out-of-range access occurred for the range returned by registered allocation function. Resource deallocation leak (pgr5031): From the beginning till the end of the program, it will indicate if the resource that has been allocated using allocation function is not freed by the freeing function but the program is terminated. Resource repeated freeing (pgr5041): From the beginning till the end of the program, it will indicate if the resource that has been allocated using allocation function is freed repeatedly by the freeing function but the program is terminated.

88	**[RETURN_CHECK_FUNCTION] label**
	The name of a function that takes an exception as the return value shall be written as follows: *function name fopen
	judgment leak of return value When the return value of the registered function is not checked with the *if* statement, a message will be output.
89	**[RETURN_UNUSE_FUNCTION] label**
	The name of a function which does not need to check its return value shall be written as follows. The same information cannot be registered twice. *function name strcpy
	To check the return value of function (pgr0519): If the return value of function is not checked, pgr0519 will be output. It can be used to suppress pgr0519.
90	**[SET_VARIABLE_FUNCTION] label**
	The function name and the external variable name that are checked immediately after the function call shall be written as follows. When [function name] recurs, only first one is valid. *function name, external variable name getms,ERR
	Checks whether the external variable judgment exists after the function call. (pgr0735): If no registered external variable judgment exists after the registered function is invoked, a message will be output.
91	**[SIGNAL_ATOMIC_TYPE] label**
	The information of a types treated as atomic types shall be written as follows: *type name sig_atomic_t
	Use variables that are not asynchronous safe (pgr6051): If you are using a variable that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [SIGNAL_FUNCTION] Use variables that are not asynchronous safe (pgr6053): If you are using a variable that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [ASYNC_SIGNAL_SAFE], [SIGNAL_FUNCTION]
92	**[SIGNAL_FUNCTION] label**

	The information of a function to register signal handler shall be written as follows: *function name, position of signal handler function pointer signal,2
	Check on exclusive control in multi-thread function (pgr1256): When you use bit field without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [THREAD_CREATE_FUNCTION], [MULTI_THREAD_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION] Check on exclusive control in multi-thread function (pgr1257): When you use variable qualified with volatile without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [THREAD_CREATE_FUNCTION], [MULTI_THREAD_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION] Use functions that are not asynchronous safe (pgr6050): If you are using a function that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [ASYNC_SIGNAL_UNSAFE] Use variables that are not asynchronous safe (pgr6051): If you are using a variable that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [SIGNAL_ATOMIC_TYPE] Use functions that are not asynchronous safe (pgr6052): If you are using a function that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [ASYNC_SIGNAL_SAFE], [ASYNC_SIGNAL_UNSAFE] Use variables that are not asynchronous safe (pgr6053): If you are using a variable that is not asynchronous safe, the message will be output. Besides, the following label definitions are necessary. [ASYNC_SIGNAL_SAFE], [SIGNAL_ATOMIC_TYPE]
93	**[STRING_BUFFER_CHECK_FUNCTION] label**
	The information of a function to specify the copy size of string copy shall be written as follows. The same information cannot be registered twice. *function name, parameter position of copy target, parameter position of copy source, parameter position of copy size. strncpy,1,2,3
	Null character is not taken into account in memory size. (pgr0558, pgr0559): If the null character is not taken into account in the copied character number, specified by the argument of the registered function, a message will be output.

94	**[STRING_BUFFER_CONCATENATE_FUNCTION] label** The information of a function to concatenate strings shall be written as follows: *function name, parameter position of concatenation area, parameter position of concatenation size strncat,1,3 wcsncat,1,3 Illegal memory access (pgr0585): When the size of the strings to be concatenated using the registered function is the same size with the string in the concatenation location, a message will be output.
95	**[STRING_BUFFER_MALLOC_CHECK_FUNCTION] label** The information of a function to allocate the string memory shall be written as follows. The same information cannot be registered twice. *function name, parameter location of space size, parameter location of element count(optional) malloc,1 calloc,2,1 Null character is not taken into account in memory size. (pgr0557): If the memory size of the null character is not taken into account in the argument of the registered function, a message will be output Checking for copy beyond area size (pgr6031): Registers the Agile^{+} Relief indication level if the copy area size is unknown or the destination area size is unknown. Besides, the following label definitions are necessary. [MEMORY_COPY_CHECK_FUNCTION]
96	**[STRING_COMPARE_FUNCTION] label** Information of string comparison function shall be written as follows: * function name,arg1 pos,arg2 pos strcmp,1,2 Illegal treatment of sensitive data (pgr1162): When the comparison of password storage variable and string literal is done or string literal to password storage variable is substituted, a message will be output. Besides, the following label definitions are necessary. [PASSWD_IDENTIFIER]
97	**[STRING_INPUT_FUNCTION] label** The information on function to acquire character string shall be written as follows. *function name, argument position in which acquisition character string is stored fgets,1

	Size confirmation leakage of acquisition character string (pgr1154): When using it to calculate the index of the reference sequence without confirming the size of the character string acquired in the registered function, a message will be output. Besides, the following label definitions are necessary. [STRING_LENGTH_FUNCTION]
98	**[STRING_LENGTH_FUNCTION] label** The information of a function to acquire string length, in the case of no null characters, shall be written as follows. The same information cannot be registered twice. *function name, parameter position of string strlen,1 Null character is not taken into account in memory size. (pgr0557, pgr0559): Null characters are not taken into account when string length is evaluated by the registered function. Size confirmation leakage of acquisition character string (pgr1154): When using it to calculate the index of the reference sequence without confirming the size of the character string acquired in the registered function, a message will be output. Besides, the following label definitions are necessary. [STRING_INPUT_FUNCTION]
99	**[TEMPDIR_FILE_OPEN_FUNCTION] label** The information of a function to perform file operation shall be written as follows. The same information cannot be registered twice. *function name, parameter position of file name, dangerous string 1, ..., dangerous string n fopen,1,/tmp Checks whether the file name in the parameter of the file operation function is valid. (pgr0560, pgr0581): If the file name passed to the argument of the registered function begins with an invalid string, a message will be output.
100	**[THREAD_CREATE_FUNCTION] label** Information of function that makes thread of control shall be written as follows: * function name, Position of function pointer of child thread, Position of arguments of child thread CreateThread,3,4 AfxBeginThread,1 When only function name and position of function pointer of child thread are specified, only pgr1160 is checked.

	Check on exclusive control in multi-thread function (pgr1160): When you use static variable and the external variable without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [EXCLUSION_LOCK_UNLOCK_FUNCTION] Check on exclusive control in multi-thread function (pgr1256): When you use bit field without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [MULTI_THREAD_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION], [SIGNAL_FUNCTION] Check on exclusive control in multi-thread function (pgr1257): When you use variable qualified with volatile without using the exclusive control function in the multi-thread function, a message will be output. Besides, the following label definitions are necessary. [MULTI_THREAD_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION], [SIGNAL_FUNCTION] Passing neither external variable nor static variable to the child-thread (pgr6002) If neither external variable nor static variable is passed to registered multithread function, the message will be output. Using the thread-unsafe function (pgr6008): When the registered thread-unsafe function is called in multi tread function, the message will be output. Moreover, the following label is required. [MULTI_THREAD_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION], [THREAD_CREATE_FUNCTION]
101	**[THREAD_UNSAFE_FUNCTION] label** The thread-unsafe function shall be written as follows: * function name asctime basename Using the thread-unsafe function (pgr6008): When the registered thread-unsafe function is called in multi tread function, the message will be output. Moreover, the following label is required. [MULTI_THREAD_FUNCTION], [EXCLUSION_LOCK_UNLOCK_FUNCTION], [THREAD_CREATE_FUNCTION]
102	**[VARIADIC_STRING_GETENV_FUNCTION] label** Function identifier to acquire environmental variables of unpredictable length shall be written as follows: * function name getenv

	The return value of environment variable acquisition function is copied onto the array. (pgr1164): When the return value of the unpredictable length of environment variable acquisition function is copied directly onto the array, a message will be output. Besides, the following label definitions are necessary. [AREA_COPY_FUNCTION]
103	**[VOID_POINTER_RETURN_FUNCTION] label** The function that may convert the return value of void pointer-type into other pointer-types shall be written as follows: * function name malloc calloc Conversion of void pointer-type into other pointer-types (pgr0826): A message will be not output because it is determined that the return value of the registered function may convert it into other pointer-types.
104	**[VULNERABLE_COMMAND_INJECTION_FUNCTION] label** Function weak compared with commands injection shall be written as follows: * function name, notes system, Use of Function weak compared with command injection (pgr1172): When you use the registered function, a message will be output.
105	**[WORSE_FUNCTION_BETTER_FUNCTION] label** The recommended function name and the not recommended function name shall be written as follows. When [not recommended function] recurs, only the first one is valid. *not recommended function name, the recommended function name strcpy,strncpy When the not recommended function is invoked (pg0579): If the registered [not recommended function] is invoked, a message will be output.
106	**[WRITE_FILE_FUNCTION] label** The name of function which is writing into file shall be written as follows: *function name, parameter position of file pointer fprintf,1 Write into the file which is opened with read mode (pgr0888): When writing into the file which is opened with read mode by the registered function, the message will be output. Besides, the following label definitions are necessary. [OPEN_FILE_FUNCTION], [CLOSE_FILE_FUNCTION], [OPEN_FILE_READ_MODE]

1.1.3.2 How to Suppress Messages (--base, --diff options)

You can suppress the output of messages that you no longer want to check, for example, because you do not need to modify the detected messages. You can easily exclude previously checked messages from being checked, and you can check messages efficiently.

The following describes how to suppress the output of messages that do not require checking.

Step 1. Backing Up the Message Suppression Record Directory

If you have previously run the pgr5 command with the --base, --diff options, back up the message suppression recording directory (.qm).

The message suppression recording directory is stored in the directory specified by the --temp-file option, the --project option, or the directory where the C/C++ sources are located.

Note:If you have not previously run the pgr5 command with the --base and --diff options, the message suppression recording directory does not exist.

Step 2. Output of a message

Execute the pgr5 command (Not specify the --base option) to set the message to suppress.

Step 3: Setting a Reference Point for messages

Execute the pgr5 command (Specify the --base option) to set the message to suppress.

Specify an option to the pgr5 command that is equivalent to the option specified in step 2.

Note:If you have set a reference point in the past, the setting is overwritten.

Note:If you accidentally set a reference point, delete the message suppression recording directory and restore the message suppression recording directory that you backed up in Step 1.

Step 4. Output of a new message

Execute the pgr5 command (Specify the --diff option) and print a message.

Specify an option to the pgr5 command that is equivalent to the option specified in step 2.

At this time, a "Confirmed" column is appended to the message to indicate whether confirmation is required. For more information about output formats, please refer to “2.2.2 Output format using --base, --diff options”.

Step 5. Review messages

Check the output message to see if the program needs to be modified.

Messages with 1, 4, or 32 in the "Confirmed" column do not need to be checked because they have been checked in the past.

Note:Messages with "Confirmed" columns 1 and 32 are set to unchecked in a different way from setting the Reference Point. For more information about the "Confirmed" column, please refer to “2.2.2 Output format using --base, --diff options”.

[Command example]

- For Single File Check

```
%pgr5 KGNU/GCC4 -DWIN -c --csv c:\output\a.c > msg_old.csv              (1)
%pgr5 --base -KGNU/GCC4 -DWIN -c --csv c:\output\a.c                    (2)
%pgr5 --diff -KGNU/GCC4 -DWIN -c --csv c:\output\a.c > msg_new.csv      (3)
```

(1) Use the -c option to perform a single file check.

(2) Specify the --base option to set the reference point.

(3) Use the --diff option to perform a single file check.

- For Multiple File Check

```
%pgr5 --csv c:\output\a.c.ao c:\output\b.c.ao > msg_old.csv             (1)
%pgr5 --base --csv c:\output\a.c.ao c:\output\b.c.ao                    (2)
%pgr5 --diff --csv c:\output\a.c.ao c:\output\b.c.ao > msg_new.csv      (3)
```

(1) Check multiple files.

(2) Specify the --base option to set the reference point.

(3) Specify the --diff option to perform a multi-file check.

1.1.4. Return Value

If pgr5 ends normally it will return 0. If an error occurs pgr5 will return a value other than 0.

1.1.5. Output File

(1) Output File

When the pgr5 analysis ends normally the following files are generated as output files.

a) Analytical Result File (.ao)

The syntax analytical result file of the C/C++ analysis module is generated in the same directory as the C/C++ source(*1).

b) Message File(.mes)

The message files of the preprocess, syntax analysis and link processes are generated in the directory of C/C++ source(*1) by default or the directory specified by --project option.

c) Link Result File (.aa)

The file that is output after the mutual reference of the identifiers occurring in link syntax will be generated in the directory of C/C++ source[1] by default or the directory specified by --project option.

When -c option is not selected the link results of all the specified source files are generated.

d) File to Reserve File Information (.fd)

It is the file information file generated by preprocessing. It is generated on the directory where the C/C++ source(*1) is located.

e) Option record file (.ar)

The option during parting that was generated by C parsing section is recorded in the file. It is generated at the directory with C source (*1).

f) beginning point information file of Wide-ranging Detective (.ap)

The beginning point information file needed for Wide-ranging Detective indication (*2) generated at link processing part is extracted. This file is generated at the directory with C source and the directory specified by the --project option.

In addition, the license of Wide-ranging Detective Option is needed for generating this file.

g) Route file (.srinf)

It is the problem route file of Wide-ranging Detective indication (*2) generated by link processing part. For each indication, generated the file at the directory with C source (*1) and the directory specified by the --project option.

In addition, the license of Wide-ranging Detective Option is needed for generating this file.

h) Files under the Message suppression record directory (.qm)

Files that records the status of the C/C++ analysis module and the suppression of messages generated by the link processing part when using the --base, --diff options. It is generated in the directory where the C/C++ source is located (*1) or in the directory specified by the --project option.

(*1) In the single file check, if you wish to output to other places than the directory where the C/C++ source is located, then specify --temp-file option. For --temp-file option, please refer to the desctription of --temp-fileoption in "1.1.3 option instruction"

(*2) The Wide-ranging Detective function is to track the route of function invocation and indicate the place where is likely to have a failure. In addition, to use the Wide-ranging Detective function, the license of Wide-ranging Detective Option is needed.

(2) Naming Rule

The following explains what kind of file names will be generated when a single or multiple source files are specified.

[Example1] when one source file is specified:

% pgr5 a.c -> outputs messages for the single file and the link result

(After the command is executed)

a.c.ao, a.c.aa, a.c.fd, a.c.mes, a.c.ar, a.c.ap, a.c.link.aa, a.c.link.mes will be generated.

In addition, **one and unique .sfrinf** will be generated for each Wide-ranging Detective indication.

[Example2] When multiple source files are specified:

% pgr5 a.c b.c c.c -> outputs messages for the single file and the link result

(After the command is executed)

a.c.ao, a.c.aa, a.c.fd, a.c.mes, a.c.ar, a.c.ap, a.c.link.aa, a.c.link.mes, a.c.link.ap, b.c.ao, b.c.aa, b.c.fd, b.c.mes, b.c.ar, b.c.ap, c.c.ao, c.c.aa, c.c.fd, c.c.mes, c.c.ar, c.c.ap will be generated.

In addition, **one and unique .sfrinf** will be generated for each Wide-ranging Detective indication.

[Example3] In the case of multiple (a.c, b.c, c.c) source files, the messages will be extracted by each source file separately.

% pgr5 -c a.c -> only outputs messages for the single file

(After the command is executed)

a.c.ao, a.c.aa, a.c.fd, a.c.mes, a.c.ar, a.c.ap will be generated.

% pgr5 -c b.c -> only outputs messages for the single file

(After the command is executed)

b.c.ao, b.c.aa, b.c.fd, b.c.mes, b.c.ar, b.c.ap will be generated.

% pgr5 -c c.c -> only outputs messages for the single file

(After the command is executed)

c.c.ao, c.c.aa, c.c.fd, c.c.mes, c.c.ar, c.c.ap will be generated.

% pgr5 --project Sample a.c.ao b.c.ao c.c.ao -> outputs messages for the link result[1]

(After the command is executed)

Sample.aa, Sample.mes, Sample.ap will be generated.

In addition, **one and unique .sfrinf** will be generated for each Wide-ranging Detective indication.

[1] When the --project option is not used to specify the project name, the message of link module will be output to a.c.link.mes and the link result of all source files will be output to a.c.link.aa.

1.1.6. Environmental Variable

The following environmental variables are used in pgr5:

(The following environmental variables are invalid when using a GUI.)

Table 2 Environmental Variables

Environmental Variable	Description	Note
EPOMDIR	Please specify the directory for pgr5 command (pgr5.exe in Windows, pgr5 in Others) under the installation directory.	
TMP	Please specify the directory for temporary file. If no TMP predefined, the current directory is by default.	Only for Windows
TMPDIR	Please specify the directory for temporary file. If no TMPDIR predefined, /tmp is by default.	Other than Windows
PGR_CHECKLEVEL	During the extraction of Wide-ranging Detective [1] indication, set the nested array of function invocation [2] to be parsed from the range of 1 to 100. If a value is not selected or a value outside of the range is specified, this will run as if 20 had been set. [1]The Wide-ranging Detective function is to track the route of function invocation and indicate the place where is likely to have a failure. In addition, to use the Wide-ranging Detective function, the license of Wide-ranging Detective Option is needed.	

	[2]Count the number of nested function invocation from the following locations per message ID. pgr5011 : Indication location for pgr0541 pgr5021 : Location with an array reference and a variable subscript pgr5031, pgr5041 : Functions that are not called from anywhere (except the static type functions)	
PGR_PARALLEL _ANALYZE	When the command multi-Executing function is made effective, turning on is set. The command multi-Executing function operates assuming that the void when it omits, and the values other than turning on are set. The command multi-Executing function is a function that the following commands can be executed up to four at the same time by one C/C++ analysis license on one terminal. - pgr5 - pgrmisra (*1) - pgrsec (*2) - pgrcert (*3) It is possible to use it by the analysis by the make command that specifies an optional concurrency and the simulataneous analysis Executing by two or more batch files, etc. Please note the following points when you use the command multi-Executing function. - Invalidate and execute "--pch" optional when you use the command multi-Executing function by the pgr5 command. - Redirect the execution result to the File in each command execution when you use the command multi-Executing function by the pgr5/pgrmisra/pgrsec/pgrcert command. - Confirm whether the command executed at the same time exceeds four when the following message is output. "LICENSE_3002 Process number of terminal already reached."	

	(*1) The pgrmisra command is a command that analyzes the source program written in the C/C++ language, and inspects the violation of MISRA. To use the pgrmisra command, the MISRA option is needed. (*2) The pgrsec command is a command that analyzes the source program written in the C/C++ language, and inspects the IPA/SEC - C/C++ coding manners. (*3) The pgrcert command is a command that analyzes the source program written in the C/C++ language, and inspects the violation of CERT. To use the pgrcert command, the CERT option is needed. However, the license more than to execute pgrcert commands number is needed.	

1.2 **pgrmetrics** Command

1.2.1. Command Functionality

Use the Analytical result file (the .ao file and the .fd file) created in pgr5 command as the input for pgrmetrics, and a metrics result will output to the specified file. Only the .ao file is specified for pgrmetrics and pgrmetrics will reference .fd file automatically at the same time. Other command option error messages and implementation behavior error messages are output to standard error.

1.2.2. Command Format

[pgrmetrics command](hereinafter pgrmetrics)

```
% pgrmetrics (-V)(-S source code) -o output file
             (--pgrdata_dir PCH directory)
             (-i ao file list)(ao file1  ao file2...)
```

Note:()means omissible.

1.2.3. Option Instruction

The instructions for the options of pgrmetrics are given as follows:

[Version Settings]

-V	Displays pgrmetrics version, level and release number. When this option is specified all other options will be ignored.

[Input File Settings]

-S source code	Used to specify source file encoding. **SJIS** :SJIS code **EUC** :EUC code **GB2312** :GB2312 code **UTF8** :UTF-8 code When this option is not specified, the source code will be as below. -Chinese OS GB2312 code -OS of other language SJIS code for Windows; UTF-8 code for Red Hat Enterprise Linux 5 or 6; EUC code for others
-i ao file list	The .ao file documented in .ao file list is taken as input file. The .ao file is generated by pgr5. Please create ao file list in the following format: ● The line starts with .ao file name. ● The .ao file name should be delimited with a line feed. ● File name should not be enclosed with "". ● After .ao file name do not record the space character and tab

	character.

[Output File Settings]

-o output file	Uses the specified output file to create metrics files. Please make sure this option is used. Use output-file-name.mfile as the output file for file metrics. Use output-file-name.mfunc as the output file for function metrics. In the case of multiple input files the metrics will be combined into a single output file.

[Specification relating to the shortening of parsing time]

--pgrdata_dir PCH directory	If --pgrdata_dir option has been selected in pgr5, specify the PCH directory.

[Input File Settings]

.ao file	Specifies the analytical result file (.ao) created in pgr5. Multiple .ao file names can be specified.

1.2.4. Return Value

When pgrmetrics ends normally, it will return 0. If an error occurs a value other than 0 will be returned.

1.2.5. Output File

(1) Output File

When pgrmetrics ends normally the following files, generated in the directory specified with the -o option, will be output.

a) file metrics (mfile)

Outputs the following file information:

- Number of total lines
- Number of execution lines
- Number of functions

b) Function metrics (mfunc)

Outputs the following function information:

- Number of total lines
- Number of execution lines
- Number of nesting level

- Complexity
- Number of various control statements

1.3 pgrfake command

1.3.1 Command function

The pgrfake command will select the required option of pgr5 command, and execute it.

When pgrfake is used in the environment with defined set of compilation work such as make file or batch file, it is no need to set in the pgr5 command.

To use it, replace the compiler's invocation command written in the make file with the pgrfake command, and add the pgrfake command which has the same command parameter as compiler. Afterwards, execute the make file that has been updated so that Agile$^+$ Relief parse can be performed.

1.3.2 Command format

[pgrfake command] (called pgrfake hereafter)

```
% pgrfake  Command parameter of compiler
```

The command parameter that was designated to pgrfake will be converted to the command parameter of pgr5 according to the content of command parameter definition file

1.3.3 Command parameter definition file

● File creation method and configuration

There is a template file under the directory [FakeInfo] that is same as pgrfake. Use the template file and perform necessary setting according to the format described by "Format", "Section & key".

The created command parameter definition file is disposed to either of the following places and used.

- It disposes it to [FakeInfo] that exists in the same directory as pgrfake in file name [cmptable.ini].

- It disposes it to the arbitrary place by the arbitrary name.

(It is necessary to specify the file disposed by the environment variable "PGRFAKE_INI_FILE". Please refer to "1.3.6. Environmental Variable".)

* When the command parameter definition file is disposed to both, pgrfake uses the file specified by the environment variable "PGRFAKE_INI_FILE".

- Format

Format is as follows. In addition, if the line begins with ";" (semicolon), then the action will be considered to be invalid.

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

- Section & key

The section and key to set is shown in the following table:

Section & key	Description
[PGRCOMMAND]	This is the section for setting the options that are relevant to Agile^{+} Relief parse.
PGROPT	Specify the option to be converted to pgr5. If the template file is to be used, please confirm the settings of the -K option (compiler specification key). However, the following options do not become effective even if it specifies it. Please specify it with the key that corresponds respectively. -Z option : RULEFILE key -G option : GROUPFILE key --temp-file option : OUTPUTDIR key -S option : CODE key
RULEFILE	Specify the check comment definition file to be applied. If it is not specified, the default value will be applied. The default value is defined in the file [default-pgr.rul] under the [MessageInfo] directory that is the same as pgrfake.
GROUPFILE	Specify the group modification file to be applied. If it is not specified, the default value will be applied. The default value is defined in the file [MSG.MGR] under the [MessageInfo] directory that is the same as pgrfake.

OUTPUTDIR	Specify the output directory of pgrfake. If it is not specified, it will be generated under the same directory as the source file. The generated output file name is [source file name.message]. In addition, if identical source file names exist, then add a unique number to differentiate. When the Linux edition is used, "~" (HOME directory) can be specified for output directory. Example: OUTPUTDIR = ~/result This key is required when analyzing multiple files.
CODE	To specify the source code of input source.. SJIS : SJIS code EUC : EUC code GB2312 : GB2312 code UTF8 : UTF8 code
DELETEANAFILE	Set whether to delete the file output by the pgr5 command. True : Delete(Default) false : Do not delete When file indication extraction is being performed, it should be set to false, and the pgr5 command should be executed separately.
PGRLINKCHECK	Specifies whether to analyze multiple files. true : Analyze If you specify a file to analyze as an argument to the pgrfake command, a single file is analyzed. The following keys are required to analyze multiple files. OUTPUTDIR key(Required) DELETEANAFILE key Specify false. However, if the PGRLINKCHECK key is true and the DELETEANAFILE key is omitted, the DELETEANAFILE key is assumed to be false. false : Do not analyze(Default) If the source file is specified on the command line, the single file is analyzed and the results are output in the directory specified by the OUTPUTDIR key. If the source file is not specified on the command line, the analysis results that exist in the directory specified by the OUTPUTDIR key are used to analyze multiple files.

METRICS	Specifies whether metrics are output. true: Output false: Do not output(Default) If this key is true, the pgrmetrics command outputs the metrics as input to the analysis result file (The file extension is .ao) created by the pgr5 command.
METRICSOPT	Specify the option to be converted to pgrmetrics. However, do not specify the following options in this key as they are assembled internally by pgrfake. -S option -o option -i option ao file Please refer to "1.2.3. Option Instruction".
PGRLINKRESULT NAME	Specifies the name of the multi-file analysis results file. If not specified, defaults "pgrlink_result". Outputs the results of multiple file analysis under the directory specified by the OUTPUTDIR key.
[OPTION]	This is the section for setting the comparison option of compiler and Agile$^+$ Relief.
Compiler options	Specify the option of pgr5 command. This option is corresponding to the compiler option. If hyphen (-) or slash (/) needs to be specified during command execution, please specify the hyphen (-) or slash (/) as well.
[ENVIRONMENT]	This is the section for setting the compiler environment.
Compiler environment variables	Specify the value of environment variable of compiler to the value of a certain option of pgr5.
[ANALYZERULE]	This is the section for setting the parse rule of compiler command parameter.
CPPEXT	Specify the extension of C++ source file. CPPEXT cannot specify extension "c". The standard is registered to extension "c" as C source file.
DELIMITERKEY	Specify the separator to be used when multiple option values exist.
CMDFILEKEY	Specify the beginning character of a file. These beginning character are used for identifying the file with command parameter being written.

SKIPKEY	Specify this option when the parse of command parameter is to be disabled. The command parameter that is specified after the designated character will not be parsed.
NOANALYZEKEY	Specify the option of the compiler to control the execution of the pgr5 command (option/M of gcc compiler, option/E of Visual C++ compiler (cl command), etc.). When the option set by this item is specified for an option of the pgrfake command, the pgr5 command is not executed.
[GUIDELINE]	Section for configuring checks for coding guideline violations.
MISRA_VER	Specify the MISRA version to check. 1: MISRA-C V1 2: MISRA-C V2 3: MISRA-C V3 P1: MISRA-C++ V1 If you specify MISRA_VER Key, specify the -F option (identifier file) for the PGROPT key in the PGRCOMMAND section. - You need a license for the MISRA option to check for MISRA violations.
MISRA_OPT	Specify the parameters of the command to check for MISRA violations (pgrmisra). However, the following options are not enabled. Use the corresponding key. -T option : MISRA_VER key -Z option : MISRA_RULEFILE key For more information about options, please refer to "2.3 Command Options" in "MISRA Option Manual".
MISRA_RULEFILE	Specifies the check indication definition file used by the command to check for MISRA violations (pgrmisra). If it is not specified, the default value will be applied. The default value is defined in the file [default-misraXXX.rul] under the [MessageInfo] directory that is the same as pgrmisra. "XXX" is the character that identifies the MISRA version and depends on the value of the MISRA_VER key. v1: MISRA-C V1 v2: MISRA-C V2 v3: MISRA-C V3 pv1: MISRA-C++ V1

SEC_VER	Specify the IPA/SEC version to check. 1: IPA/SEC-C V1 2: IPA/SEC-C V2 3: IPA/SEC-C V3 P1: IPA/SEC-C++ V1 P2: IPA/SEC-C++ V2 If you specify SEC_VER Key, specify the -F option (identifier file) for the PGROPT key in the PGRCOMMAND section.
SEC_OPT	Specify the parameters of the command to check for IPA/SEC-C/C++ coding manners (pgrsec). However, the following options are not enabled. Use the corresponding key. -T option : SEC_VER key -Z option : SEC_RULEFILE key For more information about options, please refer to "2.3 Command Options" in "Manual for IPA/SEC-C/C++ Check".
SEC_RULEFILE	Specifies the check indication definition file used by the command to check for IPA/SEC-C/C++ coding manners (pgrsec). If it is not specified, the default value will be applied. The default value is defined in the file [default-secXXX.rul] under the [MessageInfo] directory that is the same as pgrsec. "XXX" is the character that identifies the IPA/SEC version and depends on the value of the SEC_VER key. v1: IPA/SEC-C V1 v2: IPA/SEC-C V2 v3: IPA/SEC-C V3 pv1: IPA/SEC-C++ V1 pv2: IPA/SEC-C++ V2
CERT_VER	Specify the CERT version (Directory that stores CERT C Definition Files) to check. Example : CERT_VER = C_20180401_pgr2018 If you specify CERT_VER Key, specify the -F option (identifier file) for the PGROPT key in the PGRCOMMAND section and the --cert option for the CERT_OPT key in the GUIDELINE section. - You need a license for the CERT option to check for CERT violations.

CERT_OPT	Specify the parameters of the command to check for CERT violations (pgrcert). However, the following options are not enabled. Use the corresponding key. -T option : CERT_VER key -Z option : CERT_RULEFILE key For more information about options, please refer to "2.4.3 Command Options" in "CERT Option Manual".
CERT_RULEFILE	Specifies the check indication definition file used by the command to check for CERT violations (pgrcert). If it is not specified, the default value will be applied. The default value is defined in the file [default-certc.rul] under the directory that stores CERT C Definition Files.

- Description example

The description example is as below:

```
[PGRCOMMAND]
  PGROPT    = -K MS/VC2005
  RULEFILE  = C:\Rule.txt
 GROUPFILE = C:\Group.mgr
  OUTPUTDIR = C:\PGR\OUTPUT
  CODE = SJIS

[OPTION]
  /D = -D
  -D = -D
  /i = -i
  -i = -i

[ENVIRONMENT]
 INCLUDE = -Y

[ANALYZERULE]
  CPPEXT       = cpp
  DELIMITERKEY = ;
  CMDFILEKEY   = @
  SKIPKEY      = /link
  NOANALYZEKEY = /E
```

1.3.4 Return value

If end normally, pgrfake returns 0; If error occurs, it returns the value other than 0.

1.3.5 Output file

When operation finished normally, pgrfake will output the information to the file with [message] as extension under the designated directory of OUTPUTDIR or under the directory of source file. Each message contains the following informations.

- The checked message (If it is 1, then it displays the message for disabling command processing)
- Source file name
- Line number
- Group
- Message ID
- Message

If the MERTICS key in the PGRCOMMAND section is true, file metrics (Files with the extension mfile) and function metrics (Files with the extension mfile) are output in addition to the output of pgr5. For details of the output, Please refer to "1.2.5. Output File".

If you specify MISRA_VER key in the GUIDELINE section, it outputs pgrmisra input file (Files with the extension messageALL) and pgrmisra output file (Files with the extension misra) in addition to pgr5 output. For details of the output, please refer to "2.5 Output Files" in "MISRA Option Manual".

If you specify SEC_VER key in the GUIDELINE section, it outputs pgrsec input file (Files with the extension messageALL) and pgrsec output file (Files with the extension sec) in addition to pgr5 output. For details of the output, please refer to "2.5 Output Files" in "Manual for IPA/SEC-C/C++ Check".

If you specify CERT_VER key in the GUIDELINE section, it outputs pgrcert input file (Files with the extension messageALL) and pgrcert output file (Files with the extension cert) in addition to pgr5 output. For details of the output, please refer to "2.4.5 Output Files" in "CERT Option Manual".

1.3.6 Environmental Variable

The following environmental variables are used in pgrfake:

Table 3 Environmental Variables

Environmental Variable	Description	Note
PGRFAKE_INI_FILE	Please specify the referred command parameter definition file with the path. When the environment variable "PGRFAKE_INI_FILE" is not specified, it is considered that "cmptable.ini" under the "FakeInfo" of the same directory as "pgrfake" was specified.	

1.3.7 Notes

- Before executing the make file or batch file recorded by pgrfake, please delete the previous execute result from the directory designated by OUTPUTDIR or the directory of source file. This is not the case if you want to analyze multiple files.
- In command parameter definition file, when the setting of same type is required, write the same key in multiple lines.
  Example: When the extension of C++ source file has [cpp] and [cc]

  ```
  CPPEXT = cpp
  CPPEXT = cc
  ; It cannot be written as the following:
  ; SRCEXT=cpp, cc
  ```

2. Message

2.1 Message Category

Messages output in pgr5 can be classified into the following groups:

! group

Fatal errors equivalent to compile errors. This problem must be addressed immediately.

a group

Program defects that will most likely lead to errors. a group messages must be checked in all programs.

b group

For common programming mistakes, compiler translation differences, or low readability. In order to increase maintainability, portability and readability, please use b group. It is recommended that this group is used when programs are initially written.

c group

For likely increases in execution speed or code size reduction. Please take into account program structure to accommodate updated program performance.

d group

For type size changes resulting from portability. d group messages apply to all portable program.

f group

For violation of [The Embedded C++][1] rule. Please apply to the embedded development program.

[1] [Then Embedded C++] means:

Embedded C ++ language specification. The specification of syntax and functionality take into account invisible code creation, memory consumption and execution time, etc.

g group

Mainly for the violation of firmware coding rules. Please apply these messages to met firmware coding rules.

h group

This group indicates a decrease in program robustness.

n group

It indicates the record that violates the naming rule.

2.2 Message Format

2.2.1 Normal Output Format

The format of messages output by pgr5 is illustrated as follows.

pgr5 sample1.c sample2.c

[pgr5 Message Example]

```
"sample1.c"(13) a pgr0205 If char is unsigned, "(-1)" cannot be
equal to "ban".

"sample1.c"(18) a pgr0000 Variable "apple" may be referenced
before it has been set with a value.

"sample2.c"( 4) b  pgr0103  The processing speed would be
faster and the code more compact if the parameter "lemon" were
a pointer type or reference type.

"sample1.c"(12)  b  pgr0313 The 1st argument "pea" of the
function "func2" and the corresponding parameter "gra" in the
function definition in line 9 of "sample2.c" are of different types
(argument: struct T*, parameter: struct U *).

"sample1.c"(25)  a  pgr0511 The argument "p" at the line 2 of
"sample1.c" accessed to 0 address.
( Route: "x1.srinf" )
```

filename(line)   group MessageID   Message

Type information is output in the format of (: , :) at the end of message.

The route file is displayed in the format of (Route: "File name").

Note) File name is generally displayed with a path name.

Here for the easier understanding path name is omitted.

2.2.2 Output format using --base, --diff options

If the --diff option is specified with the pgr5 command, the "Confirmed" column is output.This column can be used to determine whether a check is required for the indicated message.

```
16 "sample.c"(10) a pgr0000 Variable "x" may be referenced before it has been set
with a value.

32 "sample.c"(10) a pgr0000 Variable "x" may be referenced before it has been set
with a value.
```

"Confirmed" column

The following table lists the numbers in the "Confirmed" column and their meanings.

Table 4 The numbers in the "Confirmed" column and their meanings

Number	Check	Meaning
1	Not Required	Message with suppressed comment[1].
3	Required[2]	Message with ! group and a suppressed comment[1].
4	Not Required	Message with reference point.
6	Required[2]	Message with ! group and a reference point.
8	Required	previously detected unacknowledged message[3].
10	Required	! group and previously detected unacknowledged message [3].
16	Required	Newly detected message.
18	Required	! group and newly detected messages.
32	Not Required	Message with require no action[4].
34	Required[2]	Message with ! group and require no action[4].

[1] “Suppressed comment” is a function which set that no check is required for the message by embedding message ID in comment in source file.

[2] ”Message with ! group” may cause important problem like a compile error, therefore recommend to check, despite it set that no check is required.

[3] "Previously detected unacknowledged message" is message that it not set that no check is required for message when it detected in the past.

[4] "Message with require no action" is outputted, if set that no check for required for message by using GUI.

2.2.3 Route file

The route file is the file in which the route where Wide-ranging Detective [1] has been executed is saved. Each indication will generate one such file.

[1]The Wide-ranging Detective function is to track the route of function invocation and indicate the place where is likely to have a failure. In addition, to use the Wide-ranging Detective function, the license of Wide-ranging Detective Option is needed.

The following information has been output in CSV format by the route file.

- Type

The type of each route, including beginning position, declaration, and source of invocation etc.

- Information

Supplement information of type, which output the reasons for judgment as expression, variable name and problem.

- File

Source file name of the route.

- Line

The line number of route.

"Example of output"

```
SJIS
"Type","Information","File","Line"
     " Beginning ","p","D:\sample\sample.c",2
     "   Function Definition","func1","D:\sample\sample.c",1
     "   Argument","p","D:\sample\sample.c",1
     "      Function Call","func1","D:\sample\sample.c",7
     "       Argument ","x","D:\sample\sample.c",7
     "       Resolve"," Assign to NULL ","D:\sample\sample.c",6
     "       Indication","pgr5011","D:\sample\sample.c",7
```

Code set of this file,

Title

3. Metrics

3.1 metrics structure

metrics is a statistical indicator of program makeup.

metrics output by the pgrmetrics command consist of file metrics and function metrics.

1) File metrics

The following information is output in CSV format:

- filename
  The source file name, or the header file name included in the source file
- number of total lines:
  The number of total lines in source file
- number of execution lines
  The number of lines in the source file after removing the comments and empty lines.
- number of functions
  The number of functions defined in the source file

2) Function metrics

The following information is output in CSV format:

- filename
  The source filename with function definitions.
- line number
  The line number of the beginning of a function definition. The beginning of a function definition is defined as follows: type function-name (parameter) {code} , the [type] of such function definition. When type is omitted, it means [function name].
- function name
- number of total lines
  The number of total lines is from the beginning of a function definition to the end of the function definition. The end of function definition means: type function name (parameter) {code}, '}'. Where '}' signals the end of the function.
- number of execution lines
  The number of lines in a function definition after removing the comments and the empty lines.
  However, when #line is used in a file with the function definition, the number of execution lines will be 0.
- number of nesting levels
  The maximum number of nesting levels of the if, while, do while, for, switch statements and number of blocks within a function definition. If no if, while, do while, for, and switch statement exists in function definition, the number of nesting levels will be 0.
  Additionally, in the syntax of: if(...) .. else if(...) ... else if(...) ...else, no matter how many else if statements exist the number of nesting levels is regarded to be 1.
- complexity 1 and complexity 2
  A value to represent the program complexity and how difficult it is to understand. Agile$^+$ Relief complexity is based on the methods (please see also note) provided by

McCabe, and can be evaluated from the number of the if, while, do while, for, and switch statements (equal to the number pi of judgment condition in note).
When ignoring complexity 1, &&, ||, and case statement.
The greater the gap between complexity 1 and complexity 2, the more complex the description.

- goto, return, break, continue
  The number of the goto, return, break, and continue statement in the function definition.
- if1 and if2
  Total number of if statements in the function definition is equal to if1 plus if2.
- while1 and while2
  Total number of while statements in the function definition is equal to while1 plus while2.
- dowhile1 and dowhile2
  The number of do-while statements in the function definition, when ignoring dowhile1.
- for1 and for2
  The number of for statements in the function definition, when ignoring for1.
- switch1 and switch2
  The number of switch1 in the function definition is the number of switch statements, and the number of switch2 in the function definition is the number of case and default statements.

Note) McCabe Complexity

The line number of the source code is generally used as an indicator. In contrast, some viewpoints state: [software is the product of human brain labor. Thus such physical and superficial attributes are not suitable to indicate the quantity of software development. A better choice is to use a corresponding indicator for brain labor.] A lot of discussions on software complexity have discussed these issues. McCabe complexity is a member of this school of thought.

McCabe thought: Software complexity is determined by control structure rather than physical factors. So the control structure complexity can be computed through the basic path number (V(G)) of the program.

If the number of judgment conditions is PI, the complexity V(G) can be evaluated from:

$$V(G) = PI + 1$$

More than half of development expenses are spent on test work, moreover, a large expense is required for maintenance. Generally believed: Total expense is closely related with the value of V(G) put forward by McCabe, making it an excellent indicator of program complexity.

Reference:[Quality Control of Software]
Union of Japanese Scientists and Engineers
Written by kanno ayatomo, Masanori Kataoka

3.2 Usage of Metrics

The recommended indicator values provided by Agile$^+$ Relief developer are as follows.

Table 5 The Indicator Value of Function Metrics

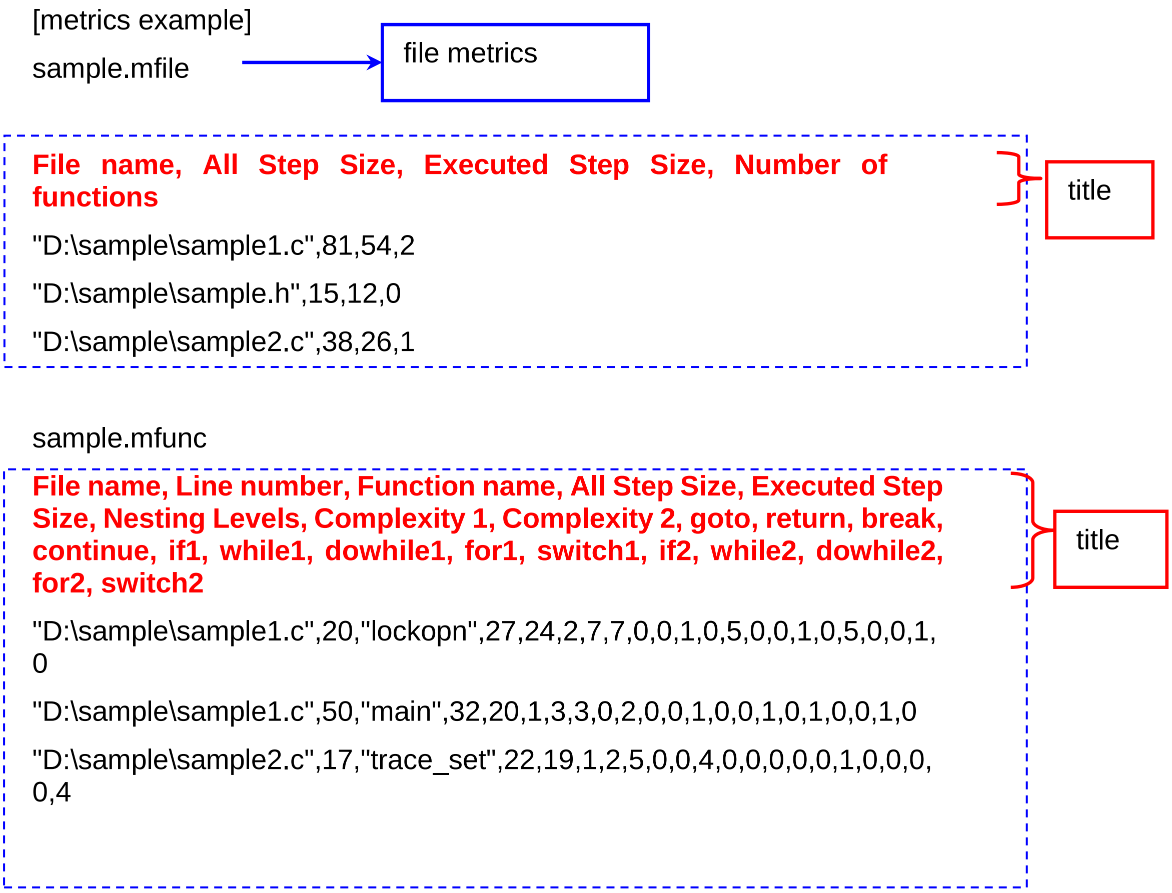
Metrics	**Indicator value**	**Description**
number of nesting levels	< 9	If the number of nesting levels exceeds 9 the source is hard to read.
complexity1	< 20	If complexity1 is beyond 20, it will be time-consuming to understand and if over 50, nobody but the author will be able to understand.

For metrics not mentioned in the above table, if a signification deviated value is identified in the deviation of metrics analysis, then the review of this file or function should be enhanced.

3.3 Metrics Output Format

Taking the metrics output in the following executions as example, the output format is described.In addition, sample1.c.ao and sample2.c.ao are the analytical result file (.ao) created by pgr5.

pgrmetrics -o sample sample1.c.ao sample2.c.ao



[metrics example]

sample.mfile

file metrics

```
File name, All Step Size, Executed Step Size, Number of
functions
"D:\sample\sample1.c",81,54,2
"D:\sample\sample.h",15,12,0
"D:\sample\sample2.c",38,26,1
```

title

sample.mfunc

```
File name, Line number, Function name, All Step Size, Executed Step
Size, Nesting Levels, Complexity 1, Complexity 2, goto, return, break,
continue, if1, while1, dowhile1, for1, switch1, if2, while2, dowhile2,
for2, switch2
"D:\sample\sample1.c",20,"lockopn",27,24,2,7,7,0,0,1,0,5,0,0,1,0,5,0,0,1,
0
"D:\sample\sample1.c",50,"main",32,20,1,3,3,0,2,0,0,1,0,0,1,0,1,0,0,1,0
"D:\sample\sample2.c",17,"trace_set",22,19,1,2,5,0,0,4,0,0,0,0,0,1,0,0,0,
0,4
```

title

4. Error Messages

4.1 Error Messages of pgr5 Command

PGRDRV_0001 Please set environmental variable EPOMDIR, or use a startup command with full path name.

Command startup error. Please set the environmental variable EPOMDIR, or use a startup command with a full path name.

PGRDRV_0002 Input file is not specified.

Input file name error. Please specify the correct source file name.

PGRDRV_0003 Inappropriate option @1.

Option @1 error. Please specify an appropriate option.

PGRDRV_0004 Parameter of option @1 is not specified.

Parameter error in option @1. Please specify an appropriate option.

PGRDRV_0005 Parameter @2 error in option @1.

Parameter @2 error in option @1. Please specify an appropriate option.

PGRDRV_0006 Internal error in option analysis. @1

Internal error. Please re-execute after the OS has been restarted.

PGRDRV_0007 Failed to acquire the current directory.

Failed to acquire the current directory. Please re-execute after the OS has been restarted.

PGRDRV_0008 Option cannot be specified after specifying input file.

Improper command format. Please check the command format before re-executing.

PGRDRV_0009 Error occurred in the option parameter @1.

Parameter @1 error. Please specify an appropriate option and parameter.

PGRDRV_0011 @1 and @2 multiple compiler specifications are specified by the -K option.

@1 and @2 multiple compiler specifications are specified by the -K option.

PGRDRV_0012 Directory @1 cannot be specified for the input file.

The directory name @1 is specified in the input file name. Please specify the correct file name.

PGRDRV_0013 The file @2 specified by the option @1 is a directory.

@2 specified for the option @1 is a directory.

PGRDRV_0014 The parameter corresponding to option @2 cannot be found in the option file @1.

The parameter corresponding to option @2 cannot be found in the option file @1. Please specify an appropriate option and parameter.

PGRDRV_0015 Unable to create message file @1.

Failed to open message file @1. Please check available disk space and the writing access to message file @1.

PGRDRV_0016 Cannot open option file @1 specified by -O option.

Failed to open the option file name @1 specified by the -O option. Please specify the correct file name.

PGRDRV_0017 Option file @1 error.

Option file @1 error. Please check the option file description and correct any errors.

PGRDRV_0018 The file name @1 specified by @2 option is not a full pass.

The file name @1 specified by @2 option is not a full pass. Please specify option with a full path name.

PGRDRV_0019 System error.@1

Because the OS was notified of an error, the process cannot continue.

The message is an error message output by the OS. Please check system for errors. Sometimes, such errors result from:

- Being out of memory:

Please re-execute the pgr5 command before performing other processes.

- Insufficient disk capacity:

Please delete any unnecessary files before re-executing the pgr5 command.

PGRDRV_0020 Unable to write message file @1.

Unable to write message file @1. Please check available disk space and the writing access to message file @1.

PGRDRV_0021 Multiple identifier files are specified for @1 and @2 by -F option.

Multiple identifier files are specified by the -F option. Please specify an appropriate option. Also, the -F option cannot perform multiple designations.

PGRDRV_0022 Cannot open input file @1.

Failed to open input file @1. Please check if file @1 exists and whether you are able to access it.

PGRDRV_0023 The -O option cannot be specified in the option file @1.

The -O option is specified in option file @1. Also, the -O option cannot be specified in an option file so specify other contents or delete this line.

PGRDRV_0024 Unable to create intermediate file @1.

Failed to create intermediate file @1. Please check available disk space and the writing access to the target output directory.

PGRDRV_0025 Option @1 is ignored.

Option @1 exclusive to PGReliefV4.0 is specified. This option will be ignored.

PGRDRV_0026 Option @1 and parameter @2 are ignored.

Option @1 exclusive to PGReliefV4.0 is specified. This option will be ignored. Please modify it to the -F option or change the format, for example by appending the contents of parameter @2 to the [NOCHECK_NOUSE_IDNETIFIER] label. For more details, please also see the -F option.

PGRDRV_0027 Can not open the identifier file @1 specified by the -F option.

Failed to open the identifier file @1 specified by the -F option. Please check if the identifier file @1 exists and whether you are able to access it.

PGRDRV_0028 Directory @1 specified by --pgrdata_dir option cannot be used.

Unable to access the directory @1 specified by --pgrdata_dir. Please confirm the existence of directory @1 and its access privilege.

PGRDRV_0029 @2 option is not specified though @1 is specified.

The @2 option that is necessary when specifying the @1 option is not specified. Please specify the @2 option.

PGRDRV_0030 Multiple naming rule files are specified for @1 and @2 by -N option.

Multiple naming rule files are specified for @1 and @2 by -N option. Please specify one

-N option only.

PGRDRV_0031 Cannot open the naming rule file @1 specified by the -N option.

Cannot open the naming rule file @1 specified by the -N option. Please check the access to file @1.

PGRDRV_0032 @1 option can not coexistent with @2 option.

The @1 and @2 options were specified together. Specify the options according to your needs.

PGRDRV_0033 Unable to create intermediate directory @1.

The intermediate directory @1 could not be created. Check your hard disk space and write permissions for the directory.

PGRDRV_0100 A system error occurred during @1 startup.

An internal error occurred when executing @1. Please reinstall Agile^{+} Relief.

PGRDRV_0101 Failed to read process code of @1.

An internal error occurred when obtaining the ending code of @1. Please check the command format and then re-execute.

PGRDRV_0102 Quit due to interruption of preprocess module processing.

Preprocess ended abnormally. Please check the output message before this message is displayed.

PGRDRV_0103 Quit due to interruption of C analysis module processing.

The C language analysis ended abnormally. If the message was output before this message was displayed, please check this message. Other than Windows, if no message is output, an error notice might be displayed for insufficient stack size.So please extend stack size with limit, ulimit or unlimit command, and then reexecute pgr5.

PGRDRV_0104 Quit due to interruption of C++ analysis module processing.

The C++ language analysis ended abnormally. Please check the output message displayed before this message.

PGRDRV_0105 Quit due to interruption of link module processing.

Link processing ended abnormally. Please check the message displayed before this message.

PGRDRV_0106 Quit due to interruption of message extraction module processing.

Message extraction processing ended abnormally. Please check the message displayed before this message. This message may be output for one of the following reasons:

- Being out of memory:

Please re-execute the pgr5 command after making more memory accessible.

- Insufficient disk capacity:

Please delete any unnecessary files before re-executing the pgr5 command.

PGRDRV_0107 Quit due to interruption of message output module processing.

Message output processing ended abnormally. Please check the message displayed before this message.

If this message is output while setting the reference point, execute analysis and then set the reference point again.

PGRDRV_9999 Error occurred during Agile$^+$ Relief execution.(code= @1)

Because an error occurred when executing Agile$^+$ Relief, the process was interrupted. Please contact the inquiry window our company about the message number and @1 code.

4.2 epomcpp Error Messages

EPOMCPP_0000 Cannot open file @1.

Failed to open file @1. Please check available disk space and access to file @1.

EPOMCPP_0001 Inappropriate option @1.

An error occurred in option @1. Please specify the appropriate option.

EPOMCPP_0002 No argument found in the option @1.

An argument error occurred in option @1. Please specify the appropriate option.

EPOMCPP_0003 Argument @1 error in -A option.

An argument @1 error occurred in option -A. Please specify an appropriate option.

EPOMCPP_0004 Argument @1 of -D option is not an identifier.

An error relating to argument @1 occurred in -D option. Please specify a proper option.

EPOMCPP_0005 File @1 specified by -K option to delete key word cannot be opened.

An error occurred in the compiler specification by the parameter of -K option, or failed to open the file to delete key word of compiler option @1. If an error occurred in the compiler specification, please enter the correct compiler specification. If the file to delete key word of compiler specification @1 could not be opened, please reinstall Agile^{+} Relief. Please execute the pgr5 command after reinstallation.

EPOMCPP_0006 Argument @1 error in -S option.

Input source file name error. Please specify the correct input source file name.

EPOMCPP_0007 Argument @1 of -U option is not an identifier.

@1 specified by -U option is not an identifier. Please specify an appropriate option.

EPOMCPP_0008 pre-reading file @1 specified by the --include option is not opened.

Failed to open the file @1 specified by the --include option. Please check if the pre-reading_file @1 exists and whether you are able to access it.

EPOMCPP_0009 Failed to open config file @1 specified by --config option.

Internal error.

EPOMCPP_0010 Failed to open option file @1 specified by -O option.

Failed to open file @1 specified by -O option. Please check access to the option file @1.

EPOMCPP_0011 Failed to acquire the current directory.

Failed to acquire the current directory.

EPOMCPP_0012 Specification error occurred in line @2 of Config File @1. @3

An error occurred in line @2 of Configuration File @1. Please re-execute the pgr5 command.

EPOMCPP_0013 Macro @1 is redefined with different values in option.

Macro @1 specified in option is redefined with different values.

EPOMCPP_0014 Insufficient memory in @1.

There was insufficient memory to complete @1 process. Please re-execute after making more memory accessible.

EPOMCPP_0015 Insufficient yacc stack in @1.

There was insufficient memory to complete @1 process. Please re-execute after making more memory accessible.

EPOMCPP_0016 An error occurred in the key word deletion file.

A format error occurred in the key word deletion file in the specified compiler. The key word deletion file may have been corrupted. Please reinstall Agile$^+$ Relief and then re-execute the pgr5 command.

EPOMCPP_0017 Ending token @2 corresponding to beginning token @1 cannot be found.

The ending token "@2" corresponding to the beginning token "@1" cannot be found. This error can result from:

- Installation Failure:

The key word deletion file in the compiler may have been corrupted. Please reinstall Agile$^+$ Relief and then re-execute the pgr5 command.

- Syntax error in preprocess:

A syntax error may have occurred in preprocess. Please check messages outputted before this message.

EPOMCPP_0018 Beginning token 4 @4 which is corresponding to beginning token 1, 2 and 3 @2 @3 @4 of type @1 cannot be found.

In keyword delete file of compile key, Beginning token 4 @4 which is corresponding to beginning token 1, 2 and 3 @2 @3 @4 of type @1 cannot be found. The beginning token 4 @1 of type 3 within the key word deletion file in the compiler cannot be found. The key word deletion file in the compiler specification may have been corrupted. Please reinstall Agile$^+$ Relief and then re-execute the pgr5 command.

EPOMCPP_0019 Ending token 4 @6 which is corresponding to beginning token 1, 2, 3 @2 @3 @4 and beginning token 4 @5 of type @1 cannot be found.

In keyword delete file of compile key, Ending token 4 @6 which is corresponding to beginning token 1, 2, 3 @2 @3 @4 and beginning token 4 @5 of type @1 cannot be found. The ending token 4 @1 of type 3 within the key word deletion file in the compiler specification cannot be found. The key word deletion file may have been corrupted. Please reinstall Agile$^+$ Relief and then re-execute the pgr5 command.

EPOMCPP_0020 Ending token 1 or 2 of @1 type 4 or 5 cannot be found.

The ending token 1 or 2 of @1 type 4 or 5 within the key word deletion file in the compiler specification cannot be found. The key word deletion file may have been corrupted. Please reinstall Agile$^+$ Relief and then re-execute the pgr5 command.

EPOMCPP_0021 System error. @1

Since the OS was notified of an error, the process cannot be continued.

The message is an error message output in the OS. Please check for system errors. Sometimes such errors result from:

- Being out of memory:
  Please execute the pgr5 command before performing other processes.
- Insufficient disk capacity:
  Please delete any unnecessary files before executing the pgr5 command.

EPOMCPP_0022 The option cannot be specified after source file.

The option cannot be specified after the source file. Please specify in sequence of options and source file.

EPOMCPP_0024 Failed to open output file @1.

The output file @1 cannot be opened. Please check available disk space, access to the output file @1 and the target output directory.

EPOMCPP_0027 @3 in line @2 of the identifier file @1 @4.

@4 corresponds to one of the following:

- is not an identifier.
- contains whitespace.
- has a pattern error in the label.
- contains an error.

It is also possible that an error may have occurred in the identifier information specified by the -F option parameter, or that an error may have occurred in the default identifier information. Please check the identifier information and modify it to establish the correct settings.

EPOMCPP_0028 File @3 included by stack in line @2 of file @1 is a directory.

If the file included with the #include statement is a directory, the message will be output.

4.3 pgrc Error Messages

EPOMC_0000 Failed to open input file @1.

Failed to open input file @1. Please recheck the access privilege to input file @1.

EPOMC_0003 Out of memory.

Memory was insufficient to perform the C analysis. Please execute again after making more memory accessible.

EPOMC_0004 Insufficient yacc stack.

Memory was insufficient to perform the C analysis. Please execute again after making more memory accessible.

EPOMC_0005 System error. @1

Since the system was notified, the process cannot be continued.

The message is an error message output in the OS. Please check for system errors.

Sometimes, such errors result from:

- Being out of memory:
  Please execute the pgr5 command before performing other processes.
- Insufficient disk capacity:
  Please delete any unnecessary files before executing the pgr5 command.

Other than Windows, when @1 is "SIGSEGV", this message may be output for insufficient stack size. So please extend stack size with limit, ulimit or unlimit command, and then reexecute pgr5.

EPOMC_0006 Attempt to refer the data not found in file @1.

File @1 is not in the Agile^{+} Relief format. File @1 generated by Agile^{+} Relief should be deleted; otherwise, it should be renamed or moved and then the pgr5 command should be re-executed.

EPOMC_0007 Process request exceeds the maximum capacity of file @1.

The process request may exceed the maximum data processing capacity for a single compiling unit. Please consider splitting the source file.

EPOMC_0008 File @1 cannot be opened.

Failed to open file @1. Please check available disk space, access to file @1 and the file @1 directory.

EPOMC_0009 File @1 format error.

File @1 is not in the Agile^{+} Relief format. File @1 generated by Agile^{+} Relief should be deleted; otherwise, please rename or move it before re-executing the pgr5 command.

4.4 pgrcc Error Messages

EPOMCC_0000 System error. @1

Since the OS was notified of an error, the process cannot be continued.

The message is an error message output in the OS. Please check for system errors. Sometimes, such errors result from:

- Being out of memory:
  Please execute the pgr5 command before performing other processes.
- Insufficient disk capacity:
  Please delete any unnecessary files before executing the pgr5 command.

Other than Windows, when @1 is "SIGSEGV", this message may be output for insufficient stack size. So please extend stack size with limit, ulimit or unlimit command, and then reexecute pgr5.

EPOMCC_0005 File @1 format error.

File @1 is not in the Agile$^+$ Relief format. File @1 generated by Agile$^+$ Relief should be deleted; otherwise, please rename or move it and then re-execute the pgr5 command.

EPOMCC_0006 Attempt to refer the data not found in file @1.

File @1 is not in the Agile$^+$ Relief format. File @1 generated by Agile$^+$ Relief should be deleted; otherwise, please rename or move it before re-executing the pgr5 command.

EPOMCC_0007 Process request exceeds the maximum capacity of file @1.

The process request may exceed the maximum data processing capacity of a single compiling unit. Please consider splitting the source file.

EPOMCC_0008 Cannot open file @1.

File @1 cannot be opened. Please check available disk space, access to file @1 and the directory of file @1.

4.5 pgrlink Error Messages

EPOMLINK_0000 System error. @1

Since the OS was notified of an error, the process cannot be continued.

The message is an error message output in the OS. Please check for system errors. Sometimes, such errors result from:

- Being out of memory:
  Please execute the pgr5 command before performing other processes.
- Insufficient disk capacity:
  Please delete any unnecessary files before executing the pgr5 command.

EPOMLINK_0005 File @1 format error.

File @1 is not in the Agile$^+$ Relief format. File @1 generated by Agile$^+$ Relief should be deleted; otherwise, please rename and move it before re-executing the pgr5 command.

EPOMLINK_0006 Attempt to refer the data not found in file @1.

File @1 is not in the Agile$^+$ Relief format. File @1 generated by Agile$^+$ Relief should be deleted; otherwise, please rename and move it before re-executing the pgr5 command.

EPOMLINK_0007 Process request exceeds the maximum capacity of file @1.

The process request may exceed the maximum data processing capacity of a single compiling unit. Please consider splitting the source file.

EPOMLINK_0008 Failed to open file @1.

Failed to open file @1. Please check available disk space, access to file @1 and the directory of file @1.

EPOMLINK_0009 --pgrdata_dir option is not specified.

--pgrdata_dir option is not specified. If --pgrdata_dir option has been selected from pgr5, please specify the --pgrdata_dir option.

4.6 pgrc/pgrcc/pgrlink Common Error Messages

PGRCDCHK_0002 Inappropriate option is specified.@1

An error occurred in the specified option. Please specify an appropriate option.

PGRCDCHK_0003 A value error in option was detected:@1 @2.

A value @2 error in option @1 was detected. Please specify an appropriate option.

PGRCDCHK_0004 Invalid file input to this program.@1

File name input error. Please check whether the file name is correct and then delete all generated files before re-executing the pgr5 command.

PGRCDCHK_0005 An invalid value was found @1 @2 and @3 @4.

Options @1 @2 and @3 @4 are invalid. Please specify an appropriate option.

PGRCDCHK_0006 Options @1 @2 and @3 @4 cannot be specified concurrently.

Options @1 @2 and @3 @4 cannot be specified concurrently. Please specify an appropriate option.

PGRCDCHK_0007 Indispensable option @1 is not specified.

Option @1 is not specified. Please specify an appropriate option.

PGRCDCHK_0008 No analytical result file is specified.

An analytical result file name (.ao) error has occurred. Please check whether it is correct and then delete all generated files before re-executing the pgr5 command.

PGRCDCHK_0009 No link result file is specified.

A link result file name error has occurred. Please specify the correct link result file name and then re-execute the pgr5 command.

PGRCDCHK_0010 Extraction message file name @1 error.

An extraction message file name @1 error has occurred. Double-byte string including

Japanese may be entered in the C source file name. Please specify the correct C source file name and then re-execute the pgr5 command.

PGRCDCHK_0011 Failed to acquire the current directory.

Failed to acquire the current directory. Please check access to the current directory.

PGRCDCHK_0100 The label name @3 error occurred in line @2 of the identifier file @1.

Label name @3 error in line @2 of the identifier file @1 specified by -F option. The label name refers to a name present in a line beginning with '[' that the string enclosed with '[' and ']', including RETURN_CHECK_FUNCTION, NULL_RETURN_FUNCTION.

PGRCDCHK_0101 The same identifier @3 is specified in line @2 of the identifier file @1.

The identifier @3 of the same name occurred in line @1 of the identifier file specified by the -F option. Please check the identifier names for errors.

PGRCDCHK_0102 An error occurred in line @2 of the identifier file @1.

An error occurred in line @2 of the identifier file @1 specified by the -F option. Please check for errors.

PGRCDCHK_0103 An error occurred in line @2 of the identifier file @1 specified by the -F option. Please check for errors.

There is no argument of option file @1 specified by the -O option in option @2.

PGRCDCHK_0104 An error occurred in line @2 of the naming rule file @1.

An error occurred in line @2 of the naming rule file @1 specified with -N option. Please refer to "6. Naming rule check", and correct the record of naming rule file.

PGRCDCHK_0200 Open file @1 error.

Open file @1 error. Please re-execute the pgr5 command.

PGRCDCHK_0201 An unexpected end of file occurred when reading file @1.

An unexpected end of file occurred when reading file @1. The file might have been corrupted. Please check the contents of file @1.

PGRCDCHK_0202 An error occurred when writing data to file @1.

An error occurred when writing data to file @1. Please check writing access to file @1.

PGRCDCHK_0300 The analytical result file @1 does not exist. Please ignore it and continue.

The nonexistent analytical result file @1 has been input. Please continue the message extraction.

PGRCDCHK_0301 The link result file @1 does not exist.

The nonexistent link result file @1 has been input. Please check the settings of the project option before re-executing the pgr5 command.

PGRCDCHK_0302 The empty analytical result file @1 was input. Please ignore it and continue.

The empty analytical result file @1 was input. Please continue the message extraction.

PGRCDCHK_0303 The empty link result file @1 was input.

The empty link result file @1 was input. Please re-execute the pgr5 command.

PGRCDCHK_0304 An analytical result file @1 that cannot be processed was entered.

The entered analytical result file @1 is invalid. The specified file might have been corrupted, or may have created files for other applications. Please re-execute the pgr5 command and the create the analytical result file.

PGRCDCHK_0305 A link result file @1 that cannot be processed was entered.

The entered link result file @1 is invalid. The specified file @1 might have been corrupted, or it may have created files for other applications. Please re-execute the pgr5 command and then create the analytical result file.

PGRCDCHK_0400 Open group file @1 error.

Open group file @1 error. Please reinstall Agile$^+$ Relief.

PGRCDCHK_0401 Open the definition file @1 for message checking error.

Failed to open the definition file @1 for message checking. Please check whether the file exists.

PGRCDCHK_0402 Incorrect message ID @3 occurred in line @2 of group file @1.

Incorrect message ID @3 occurred in line @2 of group file @1. Please reinstall Agile$^+$ Relief.

PGRCDCHK_0403 Message ID @3 in line @2 of group file @1 exceeds the range of ID.

Message ID @3 in line @2 of group file @1 exceeds the range of ID. Please reinstall Agile$^+$ Relief.

PGRCDCHK_0404 Incorrect group @4 of message ID @3 occurred in line @2 of group file @1.

Incorrect group @4 of message ID @3 occurred in line @2 of group file @1. Please reinstall Agile+ Relief.

PGRCDCHK_0405 Group @4 of message ID @3 in line @2 of group change file @1 is registered twice.

Group @4 of message ID @3 in line @2 of group change file @1 is registered twice. Please reinstall Agile+ Relief.

PGRCDCHK_0406 The pgr item corresponding to message ID @3 in line @2 of group file @1 cannot be found.

The pgr item corresponding to message ID @3 in line @2 of group file @1 cannot be found. Please reinstall Agile+ Relief.

PGRCDCHK_0408 Message ID @3 in line @2 of definition file @1 for message checking exceeds the range of ID.

Message ID @3 in line @2 of definition file @1 for message checking exceeds the range of ID. Please check its contents before re-executing the pgr5 command.

PGRCDCHK_0410 Message ID @3 in line @2 of definition file @1 is not found.

The non-existed message cannot be recorded. Please be sure and then reexecute pgr5 command.

PGRCDCHK_0411 Corresponding Message ID @1 not found.

Failed to open group change file @1. Please check whether the file @1 exists.

PGRCDCHK_0412 Incorrect message ID @3 occurred in line @2 of group change file @1.

Incorrect message ID @3 occurred in line @2 of group change file @1. Please check its contents before re-executing the pgr5 command.

PGRCDCHK_0413 Message ID @3 in line @2 of group change file @1 exceeds the range of ID.

Message ID @3 in line @2 of group change file @1 exceeds the available range of ID. Please check its contents before re-executing the pgr5 command.

PGRCDCHK_0414 Incorrect group @4 of message ID @3 occurred in line @2 of group change file @1.

Incorrect group @4 of message ID @3 occurred in line @2 of group change file @1. Please check its contents before re-executing the pgr5 command.

PGRCDCHK_0415 Because a fatal error occurred in message ID @3 in line @2 of group change file @1, the group cannot be modified.

Because a fatal error occurred in message ID @3 in line @2 of group change file @1, the group cannot be modified. Please check its contents before re-executing the pgr5 command.

PGRCDCHK_0416 An error @3 occurred in line @2 of definition file @1.

Errors occur in @3 in line @2 of the check-item definition file @1. Check the content again, and then run pgr5.

PGRCDCHK_0500 Out of memory.

There was insufficient memory to complete the message extraction. Please re-execute after more memory has been made accessible.

PGRCDCHK_0700 Processing interrupted.

Agile+ Relief processing is interrupted. Please modify the error message or fatal error message displayed in Agile+ Relief and then re-execute the pgr5 command.

PGRCDCHK_0900 Internal failure (#@1 @2 @3)

Internal error. Please re-execute after the OS has been restarted.

4.7 mes_display Error Messages

MSGOUT_0000 Open message file @1 error.

Failed to open message file @1 (MessageInfo/MSG.MGR). Please reinstall Agile+ Relief.

If the --base or --diff option is specified, it may not be consistent with the pgr5 command options before setting the reference point. See "1.1.3.2 How to Suppress Messages (--base, --diff options)" for more information.

MSGOUT_0008 Open group file @1 error.

Open group file @1 error. Please reinstall Agile+ Relief.

MSGOUT_0009 Open group change file @1 error.

Failed to open group change file @1. Please check whether the file name @1 exists.

MSGOUT_0010 Failed to acquire the current directory.

Failed to acquire the current directory during mes_display execution. Please check access to the current directory.

MSGOUT_0011 The message file is not specified.

Because no message file has been specified, the message cannot be output. Sometimes, such errors result from:

Insufficient disk space:

- Windows
  For the directory set using the environmental variable TMPDIR, please delete any unnecessary files and then re-execute the pgr5 command.
- Others
  Please delete any unnecessary files under the directory set using the environmental variable TMP and then re-execute the pgr5 command.

MSGOUT_0015 Incorrect Message ID @3 occurred in line @2 of group file @1.

Incorrect Message ID @3 occurred in line @2 of group file @1. Please reinstall Agile$^+$ Relief.

MSGOUT_0016 Message ID @3 in line @2 of group file @1 exceeds the range of ID.

Message ID @3 in line @2 of group file @1 exceeds the range of ID. Please reinstall Agile$^+$ Relief.

MSGOUT_0017 Incorrect group @4 of Message ID @3 occurred in line @2 of group file @1.

Incorrect group @4 of Message ID @3 occurred in line @2 of group file @1. Please reinstall Agile$^+$ Relief.

MSGOUT_0018 Incorrect Message ID of Message ID @3 occurred in line @2 of group change file @1.

Incorrect Message ID of Message ID @3 occurred in line @2 of group change file @1. Please check its contents before re-executing the pgr5 command.

MSGOUT_0019 Message ID @3 in line @2 of group change file @1 exceeds the range of ID.

Message ID @3 in line @2 of group change file @1 exceeds the available range of ID. Please check its contents before re-executing the pgr5 command.

MSGOUT_0020 Incorrect group @4 of Message ID @3 occurred in line @2 of group change file @1.

Incorrect group @4 of Message ID @3 occurred in line @2 of group change file @1. Please check its contents before re-executing the pgr5 command.

MSGOUT_0021 Group @4 of Message ID @3 in line @2 of group change file @1 is

registered twice.

Group @4 of Message ID @3 in line @2 of group change file @1 is registered twice. Please reinstall Agile$^+$ Relief.

MSGOUT_0022 The message of Message ID @3 in line @2 of group change file @1 cannot be found.

The message of Message ID @3 in line @2 of group change file @1 cannot be found.

MSGOUT_0023 Because a fatal error occurred in Message ID @3 in line @2 of group change file @1, the group cannot be modified.

Because a fatal error occurred in Message ID @3 in line @2 of group change file @1, the group cannot be modified. Please check its contents before re-executing the pgr5 command.

MSGOUT_0024 Corresponding Message ID @1 not found.

Corresponding Message ID @1 not found. Please reinstall Agile$^+$ Relief.

MSGOUT_0025 Open message file @1 error.

Open message file @1 error. Please reinstall Agile$^+$ Relief.

4.8 pgrmetrics Error Messages

PGRMETRICS_0000 Inappropriate option @1

Option @1 error. Please specify an appropriate option.

PGRMETRICS_0001 The output file is not specified.

The output file is not specified. Please specify the output file name using the -O option.

PGRMETRICS_0002 The input file is not specified.

The input file is not specified. Please specify the correct input file name.

PGRMETRICS_0003 Parameter error in option @1.

Parameter error in option @1. Please specify an appropriate parameter.

PGRMETRICS_0004 Cannot open file @1.

Failed to open file @1. Please check if file @1 exists and whether you have access to it.

PGRMETRICS_0005 Internal error: @1.

Because the OS was notified of an error, the process cannot be continued.

The message is an error message output in the OS. Please check for system errors. Sometimes, such errors result from:

- Being out of memory:
  Please execute the pgr5 command before performing other processes.
- Insufficient disk capacity:
  Please delete any unnecessary files before executing the pgr5 command.

PGRMETRICS_0006 An error occurred when writing file @1.

An error occurred when writing file @1. Please check the writing access to file @1.

PGRMETRICS_0007 An unexpected error has been detected.

Please reinstall Agile$^+$ Relief.

PGRMETRICS_0008 Failed to read database:@1.

Input file @1 is not an ao file. Please check the analytical result file (ao file) output by using the pgr5 command.

PGRMETRICS_0009 @1 is not an ao file.

The input file @1 extension is not ao. Please specify the input file output by using the pgr5 command and ao extension.

PGRMETRICS_0010 The list of ao files @1 contains an error.

The allowable file name length (including the path) that is specified by the -i option in the list of ao files @1, cannot exceed the smaller of the following values:

- 499(the maximum limit of the pgrmetrics command)

- the maximum possible file name length of each file system

Please check the length of each file name contained in the list of ao files @1.

PGRMETRICS_0014 --pgrdata_dir option is not specified.

--pgrdata_dir option is not specified. If --pgrdata_dir option has been selected from pgr5, please specify the --pgrdata_dir option.

PGRMETRICS_0015 --pgrdata_dir option is specified incorrectly.

--pgrdata_dir option is specified incorrectly. Please specify the --pgrdata_dir option correctly.

PGRMETRICS_9999 Error occurred during Agile$^+$ Relief execution.(code= @1)

Because an error occurred when executing Agile$^+$ Relief, the process was interrupted. Please contact the inquiry window our company about the message number and @1 code.

4.9 Error messages of pgrfake command

PGRFAKE_0001 No parameter has been specified in the command.

No parameter has been specified in the command. Please specify the command parameter of compiler as the parameter.

PGRFAKE_0101 Command parameter definition file @1 is not found.

Command parameter definition file @1 is not found. Please check if file @1 exists and whether you have access to it.

PGRFAKE_0102 Line @1 of the command parameter definition file is wrong.

Line @1 of the command parameter definition file is wrong. Please correct it.

PGRFAKE_0111 [PGRCOMMAND] section PGRLINKCHECK key is TRUE. It is necessary key @2 in @1 section.

Command parameter definition file is wrong. Check the settings of the PGRLINKCHECK key in the [PGRCOMMAND] section and the @2 key in the @1 section.

PGRFAKE_0112 [PGRCOMMAND] section PGRLINKCHECK key is TRUE. Set FALSE @2 key in @1 section.

Command parameter definition file is wrong. Check the settings of the PGRLINKCHECK key in the [PGRCOMMAND] section and the @2 key in the @1 section.

PGRFAKE_0113 [PGRCOMMAND] section PGRLINKCHECK key is TRUE. Set TRUE @2 key in @1 section.

Command parameter definition file is wrong. Check the settings of the PGRLINKCHECK key in the [PGRCOMMAND] section and the @2 key in the @1 section.

5. Before thoughts

5.1 Source File – Agile+ Relief Analysis Object (For Users)

The analysis object of Agile+ Relief is a C/C++ source program without compiler errors.

For a source file with compiling errors, Agile+ Relief will output a fatal error message to indicate the syntax error, but such behavior cannot be guaranteed. Please make sure that compiling error in the source file is removed before use.

In addition, after correcting the compiling error, a lack of analysis options may give rise to a fatal error. Please see [Message Indications Manual] to avoid the occurrence of such fatal errors.

5.2 Using the pgrfake Command to Execute All Agile+ Relief Analysis Commands (For Users)

When Agile+ Relief is used to check metrics, IPA/SEC-C/C++ coding manners, MISRA violations, or CERT violations, it is necessary to run the pgr5 command and various checking commands (pgrmetrics, pgrsec, pgrmisra, and pgrcert).

You can use the pgrfake command to run pgr5 and various test commands together.

A sample command parameter definition file required to execute the pgrfake command is stored in the following directory for reference.

(Agile+ Relief C/C++ installation directory)\Analyze\EPOM\FakeInfo\cmptable_guideline.ini

A MISRA option license is required to check for MISRA violations.

A CERT option license is required to check for CERT violations.

5.3 Linkage with Emacs editor (Except Windows)

When Emacs editor is used, the output of Agile+ Relief can be viewed in association with source file. According to M-x compile command,by executing pgr5 command, the output message of Agile+ Relief will be written into buffer of *compilation*. When a message is output, the lines of message can be checked in turn by executing C-x' command. For more details, please see also various Emacs books.

5.4 About Sequencing Methods of Output Message in Agile+ Relief (Except Windows)

Agile+ Relief is not designed to perform sequencing of output message.The combination of following commands may facilitate the comfort.

- Delete the repeated message

```
% pgr5  source filename... |  sort -u
```

- Sequence by filename or line number

```
% pgr5  source filename... |  sort   '-t('  -k 1,1 -k 2n,2
```

- Sequence by group or message ID

```
% pgr5  source filename... |  sort  -k 2,2 -k 3,3
```

- Output ! (Important) or a group only

```
% pgr5  source filename... |  sed  -n  -e'/! pgr/p'  -e'/a pgr/p'
```

- Suppress specified message only

```
% pgr5  source filename... |  sed  -e'/pgrxxxx/d'  ... continuously underlined part
```

Underlined part is the message ID to be suppressed. If want multiple message to be suppressed, please underline consecutively. About the output message ID, please refer to the supplementary [Message Indications Manual].

- Output specified message only

```
% pgr5  source filename... |  sed  -n  -e'/pgrxxxx/p'  ... continuously underlined part
```

Underlined part is the message ID to be output. If want multiple message to be output, please underline consecutively. About the output message ID, please refer to the supplementary [Message Indications Manual].

5.5 About Sequencing Methods of Output Message in Agile$^+$ Relief (For Windows)

It is convenient to display the result output by the pgr5 command or the pgrfake command in message display window (GUI). By using the message display window, messages can be displayed according to group or file unit. Also, the indicated object can be confirmed quickly by linking with source view or editor. For details of how to use the message display window, please refer to "Help".

6. Naming rule check

If -N option is specified, check the following naming rule according to the content defined in naming rule file that is transferred by -N option.

- Check the string corresponding to the prefix (Sample: add "g" to the beginning of the global variable)
- Check the string corresponding to the type (Sample: add "n" to the beginning of the int type variable)
- Check the use of uppercase letters, lowercase letters, numbers and underscore (Sample: All the macros should be in uppercase letters)

6.1 Object of naming rule check

Perform naming rule check to the following names.

- Definition of variables (global, member, and local)
- Definition or declaration of functions (Including the member function)
- Definition of class
- Definition or tag declaration of structure
- Definition or tag declaration of union
- Definition or tag declaration of enumeration
- Definition of macro

6.2 Correspondence between name structure and message ID

Names are considered to be composed of "scope prefix", "type prefix", and "name body" as below, and naming rule check should be performed respectively.

```
Name = "scope prefix" + "type prefix" + "name body"
```

The following table shows the composition of types of name and structures of name, and indicates whether to perform naming rule check. The one marked with (-) indicates that it is not the object for naming rule check. The one with message ID (pgrXXXX) indicates that it is the object for naming rule check, and the message ID will be output during violation.

Table 6 Check of Naming rule and output of indication messageID

Types of name	Structures of name		
	scope prefix	type prefix	name body
global variable	pgr1301	pgr1301	pgr1301(*)
member variable	pgr1301	pgr1301	pgr1301(*)
local variable	pgr1301	pgr1301	pgr1301(*)
function	-	pgr1304	pgr1304(*)
class	-	-	pgr1302
structure	-	-	pgr1302
union	-	-	pgr1302
enumeration	-	-	pgr1302
macro	-	-	pgr1303

* If the name is only composed of scope prefix and type prefix, without name body, pgr1305 will be output.

6.3 Naming rule file

Perform naming rule check, according to the content defined in naming rule file that is specified by -N option.

6.3.1 The initial pattern of naming rule file

The initial pattern of naming rule file is saved in NameInfo/NameRule.def in the same directory as PGR5 command.

6.3.2 Format

The format is as follows. Also, if ";"(semicolon) exists at the beginning of the line, the line is invalid.

```
[section name]
KEY = value
KEY = value
      :
[section name]
KEY = value
      :
```

6.3.3 Types of name and section names

Record section in the unit of name types.

Table 7 section name of naming rule file

Types of name	section names
global variable	GLOBAL_VAR
member variable	MEMBER_VAR
local variable	LOCAL_VAR
function	FUNC
class	CLASS
structure	STRUCT
union	UNION
enumeration	ENUM
macro	MACRO

6.3.4 Content and KEY of naming rule check

Record the KEY of check for scope prefix, type prefix, and name body in the unit of types of name.

```
;Naming rules of global variable
[GLOBAL_VAR]
;KEY specification of scope prefix
   PREFIX = g_
;KEY specification of type prefix
   int = n
;KEY specification of name body
  UPPERLOWER = FIRST_UPPER
;   Sample:
;   int  g_nCounter     <-OK
;   int  nCounter       <-NG    scope prefix does not begin with "g_".
;   int  g_iCounter      <-NG    type prefix is recorded as "i", which should be "n".
;   int  g_ncounter  <-NG    beginning of the name body is in the lowercase letter. It should
  be in the uppercase letter at the beginning.


;Naming rules of functions
[FUNC]
   Specification of KEY of type prefix
   Specification of KEY of name body


;Naming rules of class
[CLASS]
   Specification of KEY of name body
              :
```

The approach to specify the naming rule check for scope prefix is shown in the following table. Also, it can be specified when the section name is [GLOBAL_VAR], [MEMBER_VAR], and [LOCAL_VAR].

Table 8 Specification approach of naming rule check (scope prefix)

KEY	Value
PREFIX	Specify the scope prefix that corresponds to the variable names other than static variable or const variable. When it is not specified, the scope prefix will not be checked. Sample: PREFIX=g_ Check if the variable name begins with "g_".
PREFIX_STATIC	Specify the scope prefix that corresponds to the static variable. When it is not specified, the scope prefix will not be checked. Sample: PREFIX_STATIC=gs_ Check if the variable name begins with "gs_".
PREFIX_CONST	Specify the scope prefix that corresponds to the const variable. When it is not specified, the scope prefix will not be checked. Sample: PREFIX_CONST=gc_ Check if the variable name begins with "gc_".
PREFIX_STATIC_CONST	Specify the scope prefix that corresponds to the static const variable. When it is not specified, the scope prefix will not be checked. Sample: PREFIX_STATIC_CONST=gsc_ Check if the variable name begins with "gsc_".
PREFIX_CHECK	Specify whether to check the scope prefix or not. TRUE :Check (Omitted value) FALSE :Do not check

The approach to specify the naming rule check for type prefix is shown in the following table. Also, it can be specified when the section name is [GLOBAL_VAR], [MEMBER_VAR], [LOCAL_VAR], and [FUNC].

Table 9 Specification approach of naming rule check (Type prefix)

KEY	Value
char signed char unsigned char short int signed short int unsigned short int int signed int unsigned int long int signed long int unsigned long int long long int signed long long int unsigned long long int bool float double void	Specify the type prefix that corresponds to the basic type. When it is not specified, the type prefix of the basic type will not be checked. Sample: int =n Check if the type prefix is "n" in the name of variable or function of int type
pointer	Specify the type prefix that corresponds to the pointer type. When it is not specified, the type prefix of the pointer type will not be checked. Sample: pointer =p Check if the type prefix is "p" in the name of variable or function of pointer type

Typedef type name (*changeable) * KEY specification of typedef type name	Specify the type prefix that corresponds to the typedef type. When it is not specified, the type prefix of the typedef type will not be checked. Sample: ; typedef int INT32; INT32 = n32 Check if the type prefix is "n32" in the name of variable or function of INT32 type
Class name/structure name/union name/enumeration name/ structure tag name /union tag name /enumeration tag name (Variable)	Specify the type prefix that corresponds to the following: class name, structure name, union name, enumeration name, structure tag name, union tag name, and enumeration tag name. When it is not specified, the type prefix will not be checked. Sample: ; struct S {...} ; S = s Check if the type prefix is "s" in the name of variable or function of "S" type
TYPE_CHECK	Specify whether to check the type prefix or not. TRUE :Check (Omitted value) FALSE :Do not check

The approach to specify naming rule check of the name body is shown in the following table. It can be specified for all sections.

Table 10 Specification approach of naming rule check (Name body)

KEY	Value
UPPERLOWER	Specify whether the uppercase letters and lowercase letters can be used for name body.Please make sure to specify. FIRST_UPPER:Use uppercase letters at the beginning. FIRST_LOWER:Use lowercase letters at the beginning. ALL_UPPER: Use uppercase letter for all. ALL_LOWER: Use lowercase letters for all ALL_UPPERLOWER:Use either uppercase letters or owercase letters

UNDERLINE	Specify whether the underscore "_" can be used for name body. TRUE :Can be used (Omitted value) FALSE :Cannot be used
NUMBER	Specify whether numbers can be used for name body. TRUE :Can be used (Omitted value) FALSE :Cannot be used
NAME_CHECK	Specify whether to check the name body or not. TRUE : Check (Omitted value) FALSE :Do not check

6.4 Supplementary items

- While recording the naming rule file, the same key cannot be recorded in the same section.
  Sample:int = i
  ```
  int = n
  ```
- For variables or functions of pointer type, type prefix will be checked for KEY"pointer", and other type prefix will not be checked.
  Sample: (Naming rule file)
  ```
   pointer = p
   int = i
  (source program)
   int * pnCnt ;              <- Check "pointer",Do not check "int".
  ```
- Do not check the type prefix that corresponds to the type before typedef.
  Sample: (Naming rule file)
  ```
  INT32 = n32
  int    =  i
  (source program)
  typedef  int  INT32 ;
  INT32  n32Cnt;
  INT32  iCnt;               <-Do not check "int".
  ```
- When the variable is array type, the scope prefix will be checked, but the type prefix and the name body will not be checked.
  Sample: (Naming rule file)
  ```
  PREFIX = g_
  char    =  c
  (source program)
  char   g_szBuff[256];    <- Check "g_", but do not check "sz" and "Buff".
  ```
- Do not check the variables or functions of reference type.
  Sample: (source program)
  ```
  int&  refVal;              <-Do not check "refVal".
  ```
- Do not check the macro function.
  Sample:(source program)
  ```
  #define FUNC(arg) {}    <-Do not check "FUNC".
  ```

Appendix A: Notes for Agile$^+$ Relief Application in Different Compilers

Notes for inputting C/C++ source files to Agile$^+$ Relief in different compilers are illustrated as follows.

A.1 DIAB DATA Compilers

For the users of DIAB DATA compilers the options listed in following table are required for pgr5.

Table 11 The Relation Between DIAB DATA Compilers and pgr5 Options

Compiler	Compiler Options	pgr5 Options
D-cc Compiler	-Xansi	**-KDIABDATA/DCCANSI**
	-Xk-and-r	**-KDIABDATA/DCCKANDR**
	-Xstrict-ansi	**-KDIABDATA/DCCSTRICT**
	Above options are not specified.	**-KDIABDATA/DCC**

A.2 FUJITSU Compilers

For the users of FUJITSU compilers the options listed in following table are required for pgr5.

Table 12 The Relation Between FUJITSU Compilers and pgr5 Options

Compiler	Compiler Options	pgr5 Options
FUJITSU C Development Package 4.0.1	Without -X, -Xa, or -Xt	**-KFUJITSU/FCC401**
	-Xc	**-KFUJITSU/FCC401C**
SOFTUNE F2MC-16 C Compiler	Without -B	**-KFUJITSU/SOFTUNEF2MC16**
	-B	**-KFUJITSU/SOFTUNEF2MC16B**
SOFTUNE FR C Compiler	Without -B	**-KFUJITSU/SOFTUNEFR**
	-B	**-KFUJITSU/SOFTUNEFRB**

When version and options not documented here are used, please inspect predefined macro definitions and assertions in the compiler manual and specify the corresponding options in pgr5.

A.3 GNU project C Compiler, GNU Compiler Collection

For the users of GNU project C Compiler, GNU Compiler Collection the options listed in following table are required for pgr5.

Table 13 The Relation Between GNU project C Compiler and pgr5 Options

Compiler	Compiler Options	pgr5 Options

GNU project C Compiler 2.X before	All	**-KGNU/GCC**
GNU project C Compiler 3.0	-ansi and -std= for C90 specification	**-KGNU/GCC3_C90**
	Options other than above	**-KGNU/GCC3**
C, C++ Compiler 4.0-4.8 that GNU Compiler Collection offers	-ansi and -std= for C90 specification	**-KGNU/GCC4_C90**
	-ansi and -std= for C11 specification	**-KGNU/GCC4_C11**
	Options other than above	**-KGNU/GCC4**

When version and options not documented here are used, please inspect predefined macro definitions and assertions in the compiler manual and specify the corresponding options in pgr5.

A.4 HITACHI Compilers

For the users of HITACHI compilers the options listed in following table are required in pgr5.

Table 14 The Relation Between HITACHI Compilers and pgr5 Options

Compiler	Runtime Environment	pgr5 Options
HITACHI	SH	**-KHITACHI/SH**
	H8	**-KHITACHI/H8**

A.5 Mentor Graphics Japan Co., Ltd Compilers

For the users of Mentor Graphics Japan Co. Ltd compilers the options listed in following table are required in pgr5.

Table 15 The Relation Between Mentor Graphics Japan Co., Ltd Compilers and pgr5 Options

Compiler	Runtime Environment	pgr5 Options
MRI Compiler	PC version	**-KMGJ/MRIPC**
	Sun version	**-KMGJ/MRISUN**

A.6 Oracle Compilers

For the users of Oracle compilers the options listed in following table are required in pgr5.

Table 16 The Relation Between Oracle Compilers and pgr5 Options

Compiler	Compiler Options	pgr5 Options
SPARCompiler2.0.1	Without -X, -Xt, or -Xa	**-KSUN/SPARC201**

SPARCompiler3.0	Without -X, -Xt, or -Xa	**-KSUN/SPARC3**
SPARCompiler3.0.1	Without -X, -Xt, or -Xa	**-KSUN/SPARC301**
	-Xs	**-KSUN/SPARC301S**
	-Xc	**-KSUN/SPARC301C**
SPARCompiler4.0	Without -X, -Xt, or -Xa	**-KSUN/SPARC4**
	-Xs	**-KSUN/SPARC4S**
	-Xc	**-KSUN/SPARC4C**
SPARCompiler4.2	Without -X, -Xt, or -Xa	**-KSUN/SPARC42**
	-Xs	**-KSUN/SPARC42S**
	-Xc	**-KSUN/SPARC42C**
SPARCompiler5.0	Without -X, -Xt, or -Xa	**-KSUN/SPARC5**
	-Xs	**-KSUN/SPARC5S**
	-Xc	**-KSUN/SPARC5C**

When version and options not documented here are used, please inspect predefined macro definitions and assertions in the compiler manual and specify the corresponding options in pgr5.

Other points deserving attention:

Identifier

ASNI: Available characters for identifier: The initial character is an English letter or and underscore, string of English letters, digit or an underscore string from the second character. The identifier not specified in ANSI is treated as a syntax error and cannot be analyzed in Agile^{+} Relief.

A.7 Microsoft Compilers

For the users of Microsoft compilers the options listed in following table are required in pgr5.

Table 17 The Relation Between Microsoft Compilers and pgr5 Options

Compiler		pgr5 Options
Microsoft Visual C++V1.x before		**-KMS/VC1**
Microsoft Visual C++V4.0	MFC unused	**-KMS/VC4**
	MFC used	**-KMS/VC4MFC**
Microsoft Visual C++V4.2	MFC unused	**-KMS/VC42**
	MFC used	**-KMS/VC42MFC**
Microsoft Visual C++V5.0	MFC unused	**-KMS/VC5**
	MFC used	**-KMS/VC5MFC**
Microsoft Visual C++V6.0	MFC unused	**-KMS/VC6**
	MFC used	**-KMS/VC6MFC**
	MFC is not used in API of Win64bit.	**-KMS/VC6WIN64**
Microsoft Visual C++ .NET 2002	MFC not used	**-KMS/VC2003**
	For MFC	**-KMS/VC2003MFC**
Microsoft Visual C++ .NET 2003	MFC not used	**-KMS/VC2003**
	For MFC	**-KMS/VC2003MFC**
Microsoft Visual C++ 2005	MFC not used	**-KMS/VC2005**
	For MFC	**-KMS/VC2005MFC**

Microsoft Visual C++ 2008	MFC not used	**-KMS/VC2008**
	For MFC	**-KMS/VC2008MFC**
Microsoft Visual C++ 2010	MFC not used	**-KMS/VC2010**
	For MFC	**-KMS/VC2010MFC**
Microsoft Visual C++ 2012	MFC not used	**-KMS/VC2012**
	For MFC	**-KMS/VC2012MFC**
Microsoft Visual C++ 2013	MFC not used	**-KMS/VC2013**
	For MFC	**-KMS/VC2013MFC**
Microsoft Visual C++ 2015	MFC not used	**-KMS/VC2015**
	For MFC	

MFC means Microsoft Foundation Class Library.

Source(*) created through Managed Extensions for C++, Common Language Runtime or C++ Component Extensions is not supported.

(* When #using < mscorlib.dll > exists in the code,

When Keyword= "ManagedCProj" exists in projectfile of .NET, and when the /clr option or the /ZW option needs to be specified during compilation etc.)

Other points deserving attention:

Identifier

ASNI: Available characters for identifier: The initial character is an English letter or and underscore, string of English letters, digit or an underscore string from the second character. The identifier not specified in ANSI is treated as a syntax error and cannot be analyzed in Agile^{+} Relief.

A.8 GAIO Compilers

For the users of GAIO compilers **-KGAIO/XASS** options are required in pgr5.

A.9 Texas Instruments Compilers

For the users of Texas Instruments compilers the options listed in following table are required for pgr5.

Table 18 The Relation Between Texas Instruments Compilers and pgr5 Options

Compiler	Compiler Options	pgr5 Options
TEXAS_INSTRUMENTS	TI	**-KTEXAS_INSTRUMENTS/TI**
	TMS320C6000	**-KTEXAS_INSTRUMENTS/TMS320C6000**

-KTI is scheduled for deletion, please use -KTEXAS_INSTRUMENTS/TI.

A.10 ARM Compilers

For the users of ARM compilers the options listed in following table are required for pgr5.

Table 19 The Relation Between ARM Compilers and pgr5 Options

Compiler	Compiler Options	pgr5 Options
ARM_LTD	ARM	**-KARM_LTD/ARM**
	DEVELOPER_SUITE_V1_2	**-KARM_LTD/DEVELOPER_SUITE_V1_2**
	REALVIEW_V2_0	**-KARM_LTD/REALVIEW_V2_0**
	REALVIEW_V2_2	**-KARM_LTD/REALVIEW_V2_2**
	REALVIEW_V3_0	**-KARM_LTD/REALVIEW_V3_0**
	REALVIEW_V3_1	**-KARM_LTD/REALVIEW_V3_1**
	REALVIEW_V4_0	**-KARM_LTD/REALVIEW_V4_0**

-KARM is scheduled to be deleted, please use -KARM_LTD/ARM.

A.11 Renesas compiler

For the users of RENESAS compilers the options listed in following table are required for pgr5.

Table 20 The Relation Between RENESAS Compilers and pgr5 Options

Compiler		pgr5option
RENESAS	M16C	**-KRENESAS/M16C**
	M16C80	**-KRENESAS/M16C80**
	M32C	**-KRENESAS/M32C**
	M32R	**-KRENESAS/M32R**
	740	**-KRENESAS/740**
	7770	**-KRENESAS/7770**
	SuperH	**-KRENESAS/SuperH**
	H8S	**-KRENESAS/H8S**
	H8_300	**-KRENESAS/H8_300**
	78K0R	**-KRENESAS/78K0R**
	RX	**-KRENESAS/RX**

A.12 NEC compiler

For the users of NEC compilers the options listed in following table are required for pgr5.

Table 21 The Relation Between NEC Compilers and pgr5 Options

Compiler		pgr5option
NEC	78K0	**-KNEC/78K0**
	78K0S	**-KNEC/78K0S**
	78K4	**-KNEC/78K4**
	V850	**-KNEC/V850**

A.13 IAR SYSTEMS compiler

For the users of IAR SYSTEMS compilers the options listed in following table are required for pgr5.

Table 22 The Relation Between IAR SYSTEMS Compilers and pgr5 Options

Compiler		pgr5option
IAR	EWARM7.4	**-KIAR/EWARM7.4**
	EWARM8.1	**-KIAR/EWARM8.1**

Appendix B: Command Line Option Changes from Previous Products

The changes made to command line options from previous products (EYDS/PG-Relief) are illustrated below. For the differences in input files and output files, please see [1. Command Startup Method] in [Command Manual].

Table 23 List of Option Changes from Previous Products

Options in Previous Products	Modifications
-H, B	Size unit specified by parameter was changed from [byte number] into [bit number].
-K	Compiler version reconfirmation. (please see table-15)
-O	Compiler versions originally specified by the parameter are incorporated into the -K option. (see table-16)
-F	Please make sure label is written in the identifier file specified by the parameter.
-l	Incorporated into -F option.
-o	Moved to pgrmetrics command.
-i,r,s,t,n,g	With the introduction of Message groups these option are no longer supported.

Table 24 List of Changes to -K Option

Options in Previous Products	Corresponding Option
-K ASM	No longer need to specify.
-K ASMC	--slashcomment
-K COM	--slashcomment
-K DCC	Select one: -K DIABDATA/DCC -K DIABDATA/DCCANSI -K DIABDATA/DCCANSIDCCKANDR -K DIABDATA/DCCSTRICT
-K FOR	--for_scope_disable
-K GCC	-K GNU/GCC
-K HITACHISH	-K HITACHI/SH
-K MRI	Select one: -K MGJ/MRIPC -K MGJ/MRISUN
-K SOFTUNE	Select one: -K FUJITSU/SOFTUNEF2MC16 -K FUJITSU/SOFTUNEFR
-K SOFTUNEB	Select one: -K FUJITSU/SOFTUNEF2MC16B -K FUJITSU/SOFTUNEFRB
-K VC1	-K MS/VC1

-K VC4	Select one: -K MS/VC4 -K MS/VC4MFC -K MS/VC42 -K MS/VC42MFC -K MS/VC5 -K MS/VC5MFC -K MS/VC6 -K MS/VC6MFC -K MS/VC6WIN64
-K XASS	-K GAIO/XASS

Table 25 List of Changes to -O Option

Options in Previous Products	Corresponding Option
-O ARM	-K ARM
-O DCC	-K DIABDATA/DCC
-O DCCANSI	-K DIABDATA/DCCANSI
-O DCCKANDR	-K DIABDATA/DCCKANDR
-O DCCSTRICT	-K DIABDATA/DCCSTRICT
-O FCC401	-K FUJITSU/FCC401
-O FCC401C	-K FUJITSU/FCC401C
-O GCC	-K GNU/GCC
-O HITACHISH	-K HITACHISH/SH
-O MRIPC	-K MGJ/MRIPC
-O MRISUN	-K MGJ/MRISUN
-O SOFTUNEF2MC16	-K FUJITSU/SOFTUNEF2MC16
-O SOFTUNEF2MC16B	-K FUJITSU/SOFTUNEF2MC16B
-O SOFTUNEFR	-K FUJITSU/SOFTUNEFR
-O SOFTUNEFRB	-K FUJITSU/SOFTUNEFRB
-O SPARC201	-K SUN/SPARC201
-O SPARC3	-K SUN/SPARC3
-O SPARC301	-K SUN/SPARC301
-O SPARC301C	-K SUN/SPARC301C
-O SPARC301S	-K SUN/SPARC301S
-O SPARC4	-K SUN/SPARC4
-O SPARC4C	-K SUN/SPARC4C
-O SPARC4S	-K SUN/SPARC4S
-O SPARC42	-K SUN/SPARC42
-O SPARC42C	-K SUN/SPARC42C
-O SPARC42S	-K SUN/SPARC42S
-O SPARC5	-K SUN/SPARC5
-O SPARC5C	-K SUN/SPARC5C
-O SPARC5S	-K SUN/SPARC5S
-O VC1	-K MS/VC1
-O VC4	-K MS/VC4
-O VC4MFC	-K MS/VC4MFC
-O VC42	-K MS/VC42
-O VC42MFC	-K MS/VC42MFC
-O VC5	-K MS/VC5
-O VC5MFC	-K MS/VC5MFC

-O VC6	-K MS/VC6
-O VC6MFC	-K MS/VC6MFC
-O VC6WIN64	-K MS/VC6WIN64
-O XASS	-K GAIO/XASS