

FUJITSU Software

Agile⁺ Relief C/C++ V1.1.1



Improvement Reference

Preface

Agile+ Relief C/C++ (referred to below as Agile+ Relief) and PGR relief C/C++ (referred to below as PGR relief) are applications that support program review by analyzing source programs and header files written in C and C++.

This manual describes the improvements and changes that have been made from "COReTOOL/PG-Relief C/C++ for Windows V1.0" to "Agile+ Relief C/C++ V1.1.1".

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

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.

All rights reserved, Copyright(C) 2002-2023 FUJITSU LIMITED.

Contents

1. Improvements for PGRelief V2.0	1
1.1 Addition of Message Indication	1
1.1.1 a Group.....	1
1.1.2 b Group.....	1
1.1.3 d Group.....	3
1.1.4 g Group.....	3
1.1.5 h Group.....	5
1.2 Extension and Improvement of Grammar	7
1.2.1 C99 Grammar Parse.....	7
1.2.2 gcc Special Grammar Parse.....	7
1.2.3 Suffix LL/LL within #if and #elif Expression.....	7
1.3 Improvements for Message Indication	7
1.3.1 Improvement for Control Flow of try catch Grammar (C++).....	7
1.3.2 Addition of Check for Comparison Expression in pgr0031 / pgr0062/ pgr0063 / pgr0065/ pgr0362	8
1.3.3 Improvements for pgr0011 and pgr0040 (= possibly mistaken as ==)	8
1.3.4 Improvement for pgr0519 (Return Value of Function Call is Ignored).....	8
1.3.5 Improvement for pgr1005	9
1.3.6 Improvement for pgr0680	9
1.3.7 Addition of Check When Function Format Macro Used as Function Call	9
1.3.8 Changes for Message Indications and Groups.....	10
1.4 Changes for GUI.....	12
1.4.1 C99 Grammar in Identifier Settings	12
1.4.2 Changes for Message Indication Group	12
1.5 Changes for Command	13
1.5.1 Addition of -C99 Option.....	13
1.5.2 Addition of -enable_gcc_keywords Option.....	13
1.5.3 Addition of -relaxed_escaped_newline Option.....	13
1.5.4 Addition of -K Option Varieties.....	13
1.5.5 Addition of Labels Recorded in Identifier File for -F Option.....	14
2. Improvements for PGRelief V3.0	16
2.1 Message and GUI in English	16

2.2	Optional Output of Message Indication	16
2.3	Check for IPA/SEC-C Coding Operation.....	16
3.	Improvements for PGRelief V3.1	17
3.1	Extension and Improvement of Grammar	17
3.1.1	g++ Special Grammar Parse	17
3.1.2	Addition of category of -K option.....	17
3.2	Improvement of GUI Parse log display.....	18
3.3	Improvements for Message Indication	18
3.3.1	Changes for Message Indications and Groups.....	18
3.3.2	Improvement for cast expression.....	18
3.3.3	Improvements for pgr0672.....	18
3.3.4	Improvements for pgr0695.....	18
3.3.5	Change of Message ID and Group for pgr0449.....	18
3.3.6	Division of Indication Message	19
4.	Improvements for PGRelief 2007.....	20
4.1	Addition of Indications	20
4.2	Division of Indication Messages	21
4.3	Addition of Naming Rule Check Function.....	21
4.4	Addition of Corresponding IPA/SEC-C Coding Rule.....	21
4.5	#pragma once Grammar Parse	21
4.6	Changes for command	22
4.6.1	Addition of Category of -K option	22
4.6.2	Addition of label that can be recorded in the identifier file of -F option.....	22
4.6.3	Addition of --temp-file option.....	23
4.6.4	Addition of -N option	23
4.7	Support for GB2312.....	23
5.	Improvements for PGRelief 2007 autumn	24
5.1	Changes for command	24
5.1.1	Addition of Category of -K option	24
6.	Improvements for PGRelief 2008.....	25
6.1	Addition of Indications	25
6.2	Improvements for Message Indication	26
6.3	Addition of support for Compiler	26
6.4	Support to UTF-8.....	26

6.5 Prohibition of the Indication Message Output.....	26
7. Improvements for PGRelief 2008 autumn	27
7.1 Addition of Indications	27
7.2 Improvements for Message Indication	28
7.3 Improvement of GUI analysis option setting.....	29
8. Improvements for PGRelief 2009.....	30
8.1 Addition of Indications	30
8.2 Extension of function for parse time reduction	30
8.3 Addition of new command (pgrfake).....	30
8.4 Link of Command execution result and GUI.....	30
8.5 Addition of support for Compiler	30
9. Improvements for PGRelief 2009 autumn	32
9.1 Addition of Indications	32
9.2 Addition of support for Compiler	32
10. Improvements for PGRelief 2010.....	33
10.1 Addition of Wide-ranging Detective function	33
10.2 Addition of Indications	33
11. Improvements for PGRelief 2010 autumn	35
11.1 Check for IPA/SEC-C++ Coding Operation.....	35
12. Improvements for PGRelief 2011	36
12.1 Addition of Indications	36
12.2 Improvements of Indications.....	36
13. Improvements for PGRelief 2011 autumn	38
13.1 Addition of Indications	38
14. Improvements for PGRelief 2012.....	39
14.1 Addition of Indications	39
14.2 Addition of --qm option to pgr5, pgrmisra and pgrsec commands	39
15. Improvements for PGRelief 2012 autumn	40
15.1 Addition of support for Compiler	40
16. Improvements for PGRelief 2013.....	41

16.1 Addition of Indications	41
16.2 Addition of support for Compiler	41
16.3 Addition of Evidence output function	41
16.4 Addition of Diagnosis report generation tool	42
17. Improvements for PGRelief 2013 autumn	43
17.1 Check for MISRA-C:2012 Coding Operation.....	43
18. Improvements for PGRelief 2014.....	44
18.1 Addition of support for Compiler	44
18.2 Addition of command multi-Executing function	44
18.3 Addition of 64 bit version pgr5 and pgrmetrics command	44
19. Improvements for PGRelief 2014 autumn	45
19.1 Addition of --include option to pgr5 command.....	45
20. Improvements for PGRelief 2015.....	46
20.1 Addition of Indications	46
20.2 Check for IPA/SEC-C V2 (ESCR-C V2) Coding Operation.....	46
21. Improvements for PGRelief 2015 autumn	47
21.1 Correspondence to the latest OS/MW environment.....	47
21.2 Update of GUI operating environment.....	47
22. Improvements for PGRelief 2016.....	48
22.1 Addition of Indications	48
22.2 Addition/Change of support for Compiler	48
22.3 Correspondence to the latest OS environment	49
22.4 Offer of CI tool (Jenkins) Plugin	49
23. Improvements for PGRelief 2016 autumn	50
23.1 Check for IPA/SEC-C++ V2 (ESCR-C++ V2) Coding Operation	50
23.2 Check for MISRA-C:2012 Amendment 1 Coding Operation	50
24. Improvements for PGRelief 2017.....	51
24.1 Addition of Indications	51
24.2 Correspondence rate improvement of MISRA-C:2012 Coding Operation	52
24.3 Addition of support for Compiler	52
24.4 Correspondence to the latest OS/MW environment.....	52

24.5	Operationality/convenience improvement of pgrfake command	52
25.	Improvements for PGRelief 2017 autumn	54
25.1	Addition of support for Compiler	54
25.2	Correspondence to 64 bits OS	54
25.3	Improvement for pgr1395	54
26.	Improvements for PGRelief 2018.....	55
26.1	Check for CERT C Coding Operation	55
26.2	Addition of Indications	55
26.3	Addition of support for Compiler.....	56
27.	Improvements for PGRelief 2018 autumn	57
27.1	Improvement of Indications	57
27.1.1	Improvement for pgr6000, pgr6005	57
27.1.2	Improvement for pgr6001	57
27.1.3	Improvement for pgr6002	57
27.2	Change the GUI's operating environment	57
27.3	Addition of CERT C definition files	57
28.	Improvements for PGRelief 2019.....	58
28.1	Addition of Indications	58
28.2	Improvement of Indications	59
28.2.1	Improvement for pgr2216	59
28.3	Check for IPA/SEC-C V3 (ESCR-C V3) Coding Operation.....	59
28.4	Addition of CERT C definition files	59
28.5	Correspondence to the latest OS/MW environment.....	59
28.6	Convenience improvement of pgrfake command.....	60
28.7	Change of CI Tool Collaboration Jenkins Plugin provision method	60
29.	Improvements for Agile+ Relief V1.1.1.....	61
29.1	Addition of --base and --diff option to pgr5 command	61
29.2	Added a function to GUI which suppress message by using diff of analyzation	61
29.3	Improvement of Indications	61
29.3.1	Improvement for pgr0431	61
29.3.2	Improvement for pgr0520	61
29.4	Correspondence to the latest OS/MW environment.....	61

1. Improvements for PGRelief V2.0

This section demonstrates the changes and improvements made from [COReTOOL/PG-Relief C/C++ for Windows V1.0] to [COReTOOL/PGRelief C/C++ for Windows V2.0]. The following improvements will be inherited in later versions.

1.1 Addition of Message Indication

133 message indications are added.

Meanwhile, new Group [h] (those for robust improvement) is added accordingly.

Note

If h Group has been defined by user in previous PGRelief version, a conflict might occur with the added h Group. Please execute as follows:

GUI

Start the previous PGRelief version, and modify h Group to other groups in the left side of [Project Option Settings] from [Settings] menu, and then reinstall the current PGRelief version.

Command

Please modify the letter h to other lowercase English letters within the Group Change File of -G option

1.1.1 a Group

12 message indications are added into a Group (those may be output for the programs with potential damage).

ID	Message Indication	Gr	C/C++
pgr0074	The variable @1, which may be 0, is used as a divisor.	a	C/C++
pgr0381	The initial value @1 exceeds the bit width of @3 of type @2.	a	C
pgr0382	The initial value @2 of type @1 and @4 of type @3 are of different types: pointer and non-pointer.	a	C
pgr0554	@1 might access an address which exceeds the space @2 (space: @3, size : @4).	a	C/C++
pgr0557	@1 gets the space size without considering the termination character.	a	C/C++
pgr0564	The @2 argument @3 of the function @1 is the address of an auto variable.	a	C/C++
pgr0568	The argument @2 of @1 needs at least @3 bytes of space.	a	C/C++
pgr0570	The function @1 assigns a value of @4 bytes to the variable @3 of @2 bytes.	a	C/C++
pgr0573	@1 resets the loop counter.	a	C/C++
pgr0575	The condition @1 will never be evaluated as true because the same condition exists in line @3 of @2.	a	C/C++
pgr0576	This else statement will never be executed.	a	C/C++
pgr0712	There is no case label in this switch body.	a	C

1.1.2 b Group

30 message indications are added into b Group (those applied for improving maintainability, portability

and readability).

ID	Message Indication	Gr	C/C++
pgr0258	The file @1 is included and should be confirmed.	b	C/C++
pgr0383	The initial value @2 of type @1 is greater than @4 of type @3, so the correct value might not be passed.	b	C
pgr0387	The initial value @2 of type @1 and @4 of type @3 are of different types.	b	C
pgr0571	The value of the @2 @3 is not set in the constructor @1 (declaration of the @2: line @5 of @4).	b	C++
pgr0572	A comma after @1 may have been omitted accidentally.	b	C/C++
pgr0574	Updating the auto variable @1 in a return statement is meaningless.	b	C/C++
pgr0577	The condition in this if statement will never be evaluated as false.	b	C/C++
pgr0578	A multithread function might update the @1 @2 (variable declaration: line @4 of @3).	b	C/C++
pgr0715	A function declaration @1 is made in a function.	b	C
pgr0721	The expression @1 compares two object addresses of different types (@2 , @3).	b	C
pgr0723	The file @1 cannot be included redundantly (location of redundancy: line @3 of @2).	b	C/C++
pgr0726	The typedef name @1 is the same as the typedef name defined in line @3 of @2.	b	C
pgr0729	@1 uses the confusing suffix "l".	b	C
pgr0733	Function @1 is defined without prior declaration.	b	C
pgr0734	The expression @2 appears in the return statement of the void type function @1.	b	C
pgr0735	The variable @2 is not checked immediately following the function call @1.	b	C
pgr0741	The identifier @1 resembles @4 in line @3 of @2, which may lead to confusion.	b	C
pgr0742	In ANSI, the operand @2 of the bit operation @1 must be of an integer type.	b	C
pgr0743	The cast operator in the cast expression @1 is unnecessary (expression type: @2).	b	C
pgr0745	A declaration is made with no explicit type specifier.	b	C/C++
pgr0746	Space is allocated in the header file (@1 @2).	b	C/C++
pgr0748	The const variable @1 is declared without being initialized.	b	C/C++
pgr0749	The bit width of signed bit field @1 is only 1 bit.	b	C/C++
pgr0753	The label @1 is not within the { } immediately following the switch.	b	C
pgr0755	@1 and the static @4 in line @3 of @2 have the same name	b	C
pgr0762	typedef can be used instead of macro @1.	b	C/C++
pgr0763	Macro @1 is operated by @2 in a block.	b	C
pgr0765	A preprocessing directive is used in the argument @1 of a macro function.	b	C/C++
pgr0768	An #if or #elif statement generates the string "defined" in the process of macro expansion.	b	C/C++
pgr0792	Floating-point comparison @1 might lead to an unexpected result.	b	C

1.1.3 d Group

3 message indications are added into d Group (those are output for type changes and side effect occurred during immigration)

ID	Message Indication	Gr	C/C++
pgr0384	The initial value @2 of type @1 and @4 of type @3 are of different types: signed and unsigned.	d	C
pgr0385	When char is an unsigned type, the initial value @2 of type @1 and @4 of type @3 are of different types: signed and unsigned.	d	C
pgr0386	When char is a signed type, the initial value @2 of type @1 and @4 of @3 type are of different types: signed and unsigned.	d	C

1.1.4 g Group

68 message indications are added into g Group (those are mainly applied for improving maintainability, portability and readability during firmware development).

ID	Message Indication	Gr	C/C++
pgr0703	The length of the identifier @1 exceeds @2 characters.	g	C
pgr0704	The @1 @2 and the @5 in line @4 of @3 share the same name.	g	C
pgr0705	In this file, a basic type (int/short/char/long/double/float/_Bool) is used directly.	g	C
pgr0706	The octal constant @1 is used.	g	C/C++
pgr0707	The specifier "register" is used in the function @1.	g	C
pgr0708	The value of the unsigned expression @1 cannot be represented by an unsigned type.	g	C
pgr0709	L is used, denoting a wide character.	g	C
pgr0710	Multibyte character or kana(half-width) is used. (@1).	g	C
pgr0711	The keyword "continue" is used (location of relevant loop statement: line @2 of @1).	g	C
pgr0713	An expression @1 irrelevant to loop control exists in a for statement.	g	C
pgr0714	The loop counter @1 of a for statement is updated in the loop body (for statement position: line @3 of @2).	g	C
pgr0716	The function @1 is a recursive function.	g	C
pgr0717	The parameter name @1 is inconsistent with the parameter name in the function definition @4 in line @3 of @2.	g	C
pgr0718	There is a member in union @1 whose type is different from that of the float-type member @2.	g	C
pgr0719	The function @1 has 2 or more exits.	g	C
pgr0720	The expression @1 has more than 2 levels of pointers.	g	C
pgr0722	The assignment operator "=" is used in the conditional expression @1.	g	C/C++
pgr0724	Union @1 contains members of different sizes.	g	C
pgr0725	The pointer @1 exceeds 2 levels.	g	C
pgr0727	The expression @1 casts a function pointer of type @2 to a function pointer of a different type.	g	C
pgr0728	The subtraction @1 is performed with a signed and an unsigned operand.	g	C

pgr0730	volatile @1 has been used in sizeof, but the space will not be accessed.	g	C
pgr0731	@1 is not a bool value (a non-bool value is assigned in line @3 of @2).	g	C
pgr0732	The label default has been included before the label case in a switch statement.	g	C
pgr0736	@1 converts a pointer of type @2 into a function pointer.	g	C
pgr0737	The bitfield @1 is declared with type @2. The declaration should be made with signed int or unsigned int.	g	C/C++
pgr0738	No member name exists in the indicated bitfield declaration.	g	C
pgr0739	@1 @2 contains both a bit field member and a non-bit field member.	g	C
pgr0740	When char is treated as unsigned, the subtraction @1 is performed with the combination of a signed and an unsigned operand.	g	C
pgr0744	@1 @2 is not executed in all cases.	g	C/C++
pgr0747	A char is used in this file without specifying whether it is signed/unsigned.	g	C/C++
pgr0750	Function @1 is defined in K&R style or non-parameter form.	g	C/C++
pgr0751	In the declaration of the enumeration type @1, assigned members and unassigned members coexist.	g	C/C++
pgr0752	@1 uses a comma outside an initialization expression or update expression in a for statement.	g	C/C++
pgr0754	The result of @1 is used in another operation.	g	C
pgr0756	Assembly language @1 is used.	g	C/C++
pgr0757	A "/" style comment is used in the file.	g	C
pgr0758	Operator # or ## is used in a macro.	g	C/C++
pgr0759	@1, which is performed with unsigned types, results in a value that cannot be represented by an unsigned type.	g	C/C++
pgr0760	An unnecessary null statement may exist.	g	C/C++
pgr0761	Statements (in line @2 of @1) which are not preprocessing directives exist before this #include line.	g	C/C++
pgr0764	In a #define statement, parameter @2 is used two or more times in the replacement string of the macro function @1.	g	C/C++
pgr0766	The parameters @2 in the replacement string of the macro @1 are not enclosed with parentheses ().	g	C/C++
pgr0767	The replacement string of the macro function @1 is not enclosed as a whole with parentheses () or curly brackets { }.	g	C/C++
pgr0769	The char @1 not specified as signed or unsigned is not treated as a character value.	g	C
pgr0770	The signed/unsigned char @1 is not treated as a numeric value.	g	C
pgr0771	The unsigned value @1 is used without the suffix "U".	g	C
pgr0772	Two or more break statements appear in a loop.	g	C
pgr0773	@1 might exceed the bit width of the underlying type @3 of @2.	g	C
pgr0774	@1 negates the const/volatile qualifying the space pointed to by the pointer @3 of type @2.	g	C
pgr0775	The cast expression @1 casts an integer to a type that is not a smaller type of the same signed/unsigned status (cast type: @2, underlying type of the expression: @3).	g	C
pgr0776	The cast expression @1 casts an integer to a floating point type (cast	g	C

	type: @2, underlying type of the expression: @3).		
pgr0777	The cast expression @1 casts a floating point type to a type that is not a smaller type (cast type: @2, expression: @3).	g	C
pgr0778	The cast expression @1 casts a floating point type to an integer type (cast type: @2, expression: @3).	g	C
pgr0779	Pointer arithmetic operation @1 is not in the form of array indexing.	g	C
pgr0780	A union is used.	g	C
pgr0781	The cast expression @1 is a conversion involving a function pointer type and a non-integer type (cast type: @2, expression type: @3).	g	C
pgr0782	The cast expression @1 is a conversion involving an object pointer type and a type which is not an integer type, object pointer type or void* type.	g	C
pgr0783	@2 in the pointer arithmetic operation @1 is not a pointer pointing to an array.	g	C
pgr0784	@2 in the array @1 is not a pointer pointing to an array.	g	C
pgr0785	@2 and @3 of the pointer subtraction @1 do not point to the same array.	g	C
pgr0786	@1, @2 is used as macro.	g	C/C++
pgr0787	@2 in @1 is converted to a different type by an operation (underlying type before conversion: @3, underlying type after conversion: @4).	g	C
pgr0788	@2 in @1 is converted to a different type by an operation (underlying type before conversion: @3, underlying type after conversion: @4).	g	C
pgr0790	@2 in the @1 expression is not a primary expression.	g	C
pgr0791	@1 uses a comma in the initialization expression or update expression of a for statement.	g	C
pgr0793	A goto statement is used.	g	C
pgr0794	None of the parameters in the declaration of the function @1 have names.	g	C

1.1.5 h Group

20 message indications are added into h Group (those for robust improvement).

ID	Message Indication	Gr	C/C++
pgr0551	The function @1 contains a parameter of type array, but there are no unsigned parameters describing the array size.	h	C/C++
pgr0552	An address addition or subtraction operation is performed on the pointer type parameter @1 at @4 in line @3 of @2. The parameter type of the operated address might be set to array.	h	C/C++
pgr0553	The space indicated by parameter @2 of type @1 is not updated and can be qualified with const.	h	C/C++
pgr0555	The loop condition does not judge the value of the loop counter.	h	C/C++
pgr0556	%s is used in the format string of the function @1 without specifying string length.	h	C/C++
pgr0558	The operation @1 may omit the termination character of space @2 (space: @3, size @4).	h	C/C++
pgr0559	The size @2 used in the operation @1 does not include the termination character of @3.	h	C/C++
pgr0560	The @2 argument of the function @1 is a file operation in the directory	h	C/C++

	@3.		
pgr0561	Function @1 passes @3 to @2.	h	C/C++
pgr0562	The @2 argument of the function @1 is specified with a relative path.	h	C/C++
pgr0563	The function @1 is called in a loop.	h	C/C++
pgr0565	The path specified by the @2 argument of the function @1 might be a symbolic link.	h	C/C++
pgr0566	Before and after calling the function @1, privilege changes are not implemented with the function @2.	h	C/C++
pgr0567	The function @3 is not called @2 calling the function @1.	h	C/C++
pgr0568	The argument @2 of @1 needs at least @3 bytes of space.	a	C/C++
pgr0569	It cannot be guaranteed that file status will not change between calling the function @1 and calling the function @3 in line @2.	h	C/C++
pgr0570	The function @1 assigns a value of @4 bytes to the variable @3 of @2 bytes.	a	C/C++
pgr0571	The value of the @2 @3 is not set in the constructor @1 (declaration of the @2: line @5 of @4).	b	C++
pgr0572	A comma after @1 may have been omitted accidentally.	b	C/C++
pgr0573	@1 resets the loop counter.	a	C/C++

1.2 Extension and Improvement of Grammar

1.2.1 C99 Grammar Parse

During C parse, C99 grammar (ISO/IEC 9899:1999) parse now is available
'-C99'option now is available.

[Sample]

```
void func(){
    _Bool b;
    if( b){
        int *a;
        a = ( int []){ 1 , 2 , 3 };
    }
}
```

1.2.2 gcc Special Grammar Parse

During C parse, gcc (3.0) special grammar parse now is available.
'-KGNU/GCC'option now is available.

[Sample]

```
void func(){
    int a = ( { int y = 9; y; } );
    typeof (a) b;
    b = 1;
}
```

1.2.3 Suffix LL/LL within #if and #elif Expression

If suffix LL/ll found in #if and #elif expression, pgr0418 will be output and the parse is unperformable.
Now it is processed the same with integer, the parse can be done even if no option is specified.

[Sample]

```
#if 1LL
```

1.3 Improvements for Message Indication

1.3.1 Improvement for Control Flow of try catch Grammar (C++)

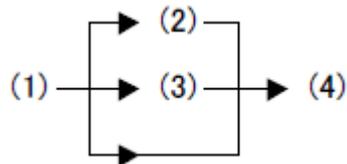
Improvements made for control flow of try ... catch are as follows. The related message indications are pgr0000, pgr0054, pgr0055, pgr0060, pgr0339, pgr0520, pgr0522, pgr0524, pgr0541, pgr0667, pgr0679 and pgr0689.

```
try { (1) }
catch( int a ){ (2) }
catch( char a ){ (3) }
(4)
```

Before:



Now:



1.3.2 Addition of Check for Comparison Expression in pgr0031 / pgr0062/ pgr0063 / pgr0065/ pgr0362

For the following message indications, the check is not only confined to the assignment expression, but also the comparison expression.

pgr0031 The expression @1 might not obtain the correct value. Use @2 instead.

pgr0062 The correct value of the @2 operation in @1 might not be able to be obtained.

pgr0063 The correct value of the @2 operation in @1 might not be able to be obtained.

pgr0065 The correct value of the << operation in @1 might not be able to be obtained.

pgr0362 The expression @1 might not obtain the correct value. Use @2 instead.

[Sample]

```
void func(  
long long x;  
int y;  
)  
{  
    if(x < ( y << 256)) /*pgr0065 indicated*/  
    {}  
}
```

1.3.3 Improvements for pgr0011 and pgr0040 (= possibly mistaken as ==)

When [expression = function call] occurred within the conditional expression, for the possibility of being mistaken as == is comparatively low, the message indication pgr0011 or pgr0040 (= possibly mistaken as ==) will not be output for a group.

[Sample]

```
int func(int);  
if ( x = func(0)) /*pgr0011 indicated*/
```

1.3.4 Improvement for pgr0519 (Return Value of Function Call is Ignored)

After Improvement: If the function name is registered under the label [RETURN_UNUSE_FUNCTION] of identifier file, the message will not be output. The standard library functions of ANSI, such as exit, free, memcpy, memset, printf, strcat, strcpy, etc. have been registered by default, thus no message will be

output.

[Sample1] In command line, how to suppress the message for function aaa.

```
% pgr5 -F idt.txt file.c
```

```
<<<identifier file: idt.txt>>>
```

```
[RETURN_UNUSE_FUNCTION]
```

```
aaa
```

[Sample 2] In GUI, how to suppress the message for function aaa.

Please execute the following 1)~4) before parse.

1) Select [Identifier Settings] in the left side of [Project Option Settings] from [Settings] menu.

2) Add the following contents in the directory of [User Settings].

```
[RETURN_UNUSE_FUNCTION]
```

```
aaa
```

3) Click [OK]

4) Start to parse.

1.3.5 Improvement for pgr1005

Originally only the function declaration is the object to be checked, now, the function declarators without parameter are the objects for checking. Meanwhile, the message to output and source for checking are changed accordingly.

1.3.6 Improvement for pgr0680

Pgr0680 will be output when a defined or declared name is the same with the external name applied in system, such as macro and function in standard library, Pgr0786 similar to pgr0680 is added in PGRRelief V2.0. Pgr0786 will be output, if macro definition or macro deletion is performed upon the name, which is the same with the external name applied in system, such as macro and function in standard library.

For the contents to output pgr0680 and pgr0786, repeated registration is no longer required for pgr0680 to check the contents registered under [RESERVED_LIBRARY_IDENTIFIER] label. In this way, [RESERVED_LIBRARY_IDENTIFIER] is valid for both pgr0680 and pgr0786.

1.3.7 Addition of Check When Function Format Macro Used as Function Call

After the macro expansion of function format within standard header, such as getchar, the definition may result in a message output for unable to fix the faults. No more check is required once such message has been sure. In addition, the same check upon the macro of function format is required for function call.

Therefore, the macro of function format registered in [NOT_EXPAND_MACRO] can suppress the macro expansion, and a check for the macro being used as a function call will be added.

That is: If the #define of [macro function name] registered beneath the [NOT_EXPAND_MACRO] label of identifier file, the substitution with [Macro Definition Replacement String] will be performed to suppress the macro expansion even if the function macro is found during expansion.

```
[format]
```

```
[NOT_EXPAND_MACRO]
```

```
macro function name, macro definition replacement string
```

[Sample 1] In command line, how to suppress the message for the macro expansion of getchar and

isalphah.

```
% pgr5 -F idt.txt file.c
```

```
<<<identifier file: idt.txt>>>
```

```
[NOT_EXPAND_MACRO]
```

```
getchar,int getchar(void);
```

```
isalpha,int isalpha( int );
```

```
<<<file.c>>>
```

```
#include <stdio.h>
```

```
#include <ctype.h>
```

```
void func(int data)
```

```
{
```

```
    int c;
```

```
    while((c = getchar()) != EOF) ←pgr0646 not indicated (Note)
```

```
    { ~}
```

```
    if(isalpha(data)) ←pgr0645 not indicated (Note)
```

```
    { ~}
```

By writing the function declaration of int getchar(void);, getchar will be checked as the function without parameters while whose return value is of int type.

Note) The message indication ID and whether the message will be output are subjected to the applied standard library.

[Sample 2] In GUI, how to suppress the message for the macro expansion of getchar and isalphah. Please execute the following 1)~4) before parse.

1) Select [Identifier Settings] in the left side of [Project Option Settings] from [Settings] menu.

2) Add the following contents in the directory of [User Settings]:

```
[NOT_EXPAND_MACRO]
```

```
getchar,int getchar(void);
```

```
isalpha,int isalpha( int );
```

3) Click [OK]

4) Start to parse file.c

```
<<<file.c>>>
```

```
#include <stdio.h>
```

```
#include <ctype.h>
```

```
void func(int data)
```

```
{
```

```
    int c;
```

```
    while((c = getchar()) != EOF) ←pgr0646 not indicated (Note)
```

```
    { ~}
```

```
    if(isalpha(data)) ←pgr0645 not indicated (Note)
```

```
    { ~}
```

By writing the function declaration of int getchar(void);, getchar will be checked as the function without parameters while whose return value is of int type.

Note) The message indication ID and whether the message will be output are subjected to the applied standard library.

1.3.8 Changes for Message Indications and Groups

The following [Message Indication], [Gr] or [C/C++] have been changed. The part in red is identified for change.

Note

For GUI users, even only once Group change for message indications has been made in [Group] to the left side of [Project Option Settings] from [Settings] menu in previous versions, the following changes to [Gr] will be unperformable.

Described as follows, if want to classify pgr0657 to a Group, please do execution in [Group] to the left side of [Project Option Settings] from [Settings] menu.

ID	Message Indication	Gr	C/C++
pgr0657	The number of bits specified for the shift operation @1 exceeds the type width of the result.	a	C/C++
pgr0701	A break statement is used to end a loop.	g	C/C++
pgr1005	Parameters are not specified in the function declarator.	b	C/C++

1.4 Changes for GUI

1.4.1 C99 Grammar in Identifier Settings

As for the system settings of C language standard option panel in [Project Preference] -> [Identifier], the lowside combobox can be applied to switch between C90 and C99.

1.4.2 Changes for Message Indication Group

Add h group (for robust enhancement) in the tool bar of message indication display window.

Add h to the Group of statistics display.

In addition, the definitions of d, f and g in tool chip and statistics display of Group button on tool bar (for message indication display) have been changed as follows

d Group: Possibility for Effect by Types

f Group: EC++ Regulation Violation

g Group: Low Maintainability (firm)

Note

When the current version is installed with the previous versions already existed, h Group will not be displayed on tool bar. So please do execution as follows:

Click [Filtering Settings] in the left side of [System Option Settings] from [Settings] menu. Please click [Tool Bar Settings] on the right of display, then select the newly added Group name and click [Display].

1.5 Changes for Command

1.5.1 Addition of `-C99` Option

The option is specified during C99 grammar parse.

After being specified, the following applications of C99 grammar are available for C language parse. The normal execution cannot be guaranteed when this option and `-KGUN/GCC3_C90` option are specified concurrently.

1. Take `_Bool` as keyword.
2. Take `_Complex`, `_Imaginary` as keyword.
3. Take `inline` as keyword.
4. Take `restrict` as keyword.
5. Start `"/"` comment.
6. Apply the macro definition of variable parameters (`_VA_ARGS_`).
(Variable parameter marco for `gcc` is available.)
7. Start `_Pragma` preprocess directive.
- 8.Regard `"_STDC_HOSTED_"` and `"_STDC_VERSION_"` as the defined macro.

1.5.2 Addition of `-enable_gcc_keywords` Option

The option is specified during `gcc` special grammar parse.

After being specified, the following applications of `gcc` special grammar are available for C language parse.

1. Start `_label_`.
2. Start `typeof`.
3. Start `_real_` and `_imag_`.
4. Start `_alignof_`.

The option is also effective even though not specified for other `gcc` special grammar.

1.5.3 Addition of `-relaxed_escaped_newline` Option

The option is specified during `gcc` special grammar parse.

After being specified, the following applications of `gcc` special grammar are available for C language parse.

The space following `'\'` will be ignored during logical line construction.

1.5.4 Addition of `-K` Option Varieties

The following 6 kinds of `-K` option have been added.

`-K HITACHI/H8`

Corresponding to the reserve words of HITACHI's H8 compiler;

`-K TEXAS_INSTRUMENTS/TMS320C6000`

Corresponding to the reserve words of Texas Instruments' TMS320C6000 compiler;

`-K TEXAS_INSTRUMENTS/TI`

Equivalent with the original `-K TI` option. Please apply this option other than `-K TI` option;

`-K ARM_LTD/ARM`

Equivalent with the original `-K ARM` option. Please apply this option other than `-K ARM` option;

-K ARM_LTD/DEVELOPER_SUITE_V1_2

Corresponding to the reserve words of ARM's SUITE V1.2 compiler;

-K ARM_LTD/REALVIEW_V2_0

Corresponding to the reserve words of ARM's REALVIEW V1.2 compiler.

Note

For the users of Advanced RISC Machines' compiler and those of Texas Instruments' compiler, the changes are made upon [Compiler Settings] in GUI and -K option of pgr5 command.

GUI User

For the users of Advanced RISC Machines' compiler, please select [Compiler Settings] in the left side of [Project Option Settings] from [Settings], and select ARM beneath the directory of ARM_LTD from [Settings] of [Compiler Type] on the right, then click [OK] or [Apply].

For the users of Texas Instruments' compiler, please select [Compiler Settings] in the left side of [Project Option Settings] from [Settings], and select TI beneath the directory of TEXAS_INSTRUMENTS from [Settings] of [Compiler Type] on the right, then click [OK] or [Apply].

Command User

When -KARM option (for Advanced RISC Machines' Compiler) of previous versions is specified, please change into -KARM_LTD/ARM option. When -KTI option (corresponding to Texas Instruments' compiler) in the previous versions is specified, please change into -KTEXAS_INSTRUMENTS/TI.

1.5.5 Addition of Labels Recorded in Identifier File for -F Option

The following 26 labels can be recorded in the identifier file of -F option.

label name	Description	Message ID
AMBIGUOUS_CHARACTERS	Possibly Confused Characters	pgr0741
AUTO_VAR_ACCESS_FUNCTION	Auto variable cannot be used as function parameter.	pgr0564
BUFFER_FORMATTED_IO_FUNCTIONS	Functions to output format information.	pgr0570
CHANGE_PROCESS_PERMISSION_FUNC TION	Functions for root privilege management and those for sub process generation	pgr0566
CHARACTER_INPUT_FUNCTION	Name of functions for character entry.	pgr0563
CHECK_FILES_ACCESSIBILITY_FUNC TION	Functions recommended and not for file state check before opening file.	pgr0569
CHECK_FILES_STATE_FUNCTION	Functions recommended for file state check before opening file.	pgr0567
CHECK_HEADER	Head files to be checked whether being processed in #include.	pgr0258
CHECK_OPERATIONS_ON_FILES_FUNC TION	Functions to expand Symbolink and those to input file.	pgr0565
COMMAND_EXECUTE_FUNCTION	Functions for program execution and strings cannot be passed to program.	pgr0561 pgr0582
DIRECT_INPUT_OUTPUT_FUNCTION	Functions to access the space of [Data Length x Data Number].	pgr0554
FORMAT_CHECK_FUNCTION	Formatted Functions	pgr0556 pgr0580

IDENTIFIER_LENGTH	Valid Identifier Length	pgr0703
MACRO_RESERVED_IDENTIFIER	External Names used in System.	pgr0786
MULTI_THREAD_FUNCTION	Name of Multithread Functions.	pgr0578
NOT_EXPAND_MACRO	Macro of function format not desired to be expanded.	All
PARA_SIZE_CHECK_FUNCTION	Functions with upperlimit in processing field.	pgr0568
RELATIVE_PATH_FUNCTION	Functions to execute the program and relative information for path designation of program.	pgr0562 pgr0583
RESERVED_LIBRARY_IDENTIFIER	External Names used in System.	pgr0680 pgr0786
RETURN_UNUSE_FUNCTION	Functions not requiring return value check.	pgr0519
SET_VARIABLE_FUNCTION	Function Name and External variable Name to be checked after function call.	
STRING_BUFFER_CHECK_FUNCTION	Functions to set copy size of strings.	pgr0558 pgr0559
STRING_BUFFER_MALLOC_CHECK_FUNCTION	Functions to obtain string space.	pgr0557
STRING_LENGTH_FUNCTION	Functions to obtain the string length without termination character.	pgr0557 pgr0559
TEMPDIR_FILE_OPEN_FUNCTION	Open files under the temporary directory.	pgr0560 pgr0581
WORSE_FUNCTION_BETTER_FUNCTION	Unrecompensed and recommended Function Names.	pgr0579

2. Improvements for PGRelief V3.0

This section demonstrates the changes and improvements made from [COrEToOL/PG-Relief C/C++ for Windows V1.0] to [COrEToOL/PGRelief C/C++ for Windows V2.0].

In addition, the improvements in previous versions will be inherited in later versions. About the improved items in last version please refer to [1.Improvements for PGRelief V2.0].

2.1 Message and GUI in English

By choosing the desired language (Japanese/English) to display during the installation, the message indication and the results of command execution, also GUI will be displayed in English.

2.2 Optional Output of Message Indication

By specifying the desired message indication and MISRA-C rule file in GUI, the selected message indication and MISRA-C rules will be output.

In addition, the suppression of the header file related message indications is available now.

2.3 Check for IPA/SEC-C Coding Operation

The check for IPA/SEC-C coding operation is available now.

3. Improvements for PGRelief V3.1

This section describes the improvements and changes that have been made from

"COReTOOL/PGRelief C/C++ for Windows V3.0" to "COReTOOL/PGRelief C/C++ for Windows V3.1".

Also, the improvements in the previous version are also included in this version. For the improvements in the previous version, please refer to "1. Improvements for PGRelief V2.0" and "Improvements for 2. PGRelief V3.0".

3.1 Extension and Improvement of Grammar

3.1.1 g++ Special Grammar Parse

During C++ parse, g++(3.0)special grammar parse now is available. Please specify the '-KGNU/GCC3' option.

[Sample]

```
__complex__ float a = 5.1I;
int minget(int a, int b) {
    return a <? b;
}
int chfunc(int a){
    switch(a) {
        case 'A'...'Z' :
            return 10;
        ~
    }
    return 0;
}
```

3.1.2 Addition of category of -K option

The following -K option is added.

-K ARM_LTD/REALVIEW_V3_0

The reserved word of ARM REALVIEW V3.0 compiler is supported

3.2 Improvement of GUI Parse log display

While performing "Whole parse", "Differences parse", "Selected parse", and "Unified parse", all the indication messages will be displayed in the parse log, so the display will be time-consuming. This is also relating to the number of indication messages, but the parse time can be reduced through the following operations.

- 1) Select [System Option Settings] from the [Settings] Menu
- 2) Select [Parse Action Settings] on the left
- 3) Select [Only output the group messages to the log] on the right
- 4) Click "OK"

3.3 Improvements for Message Indication

3.3.1 Changes for Message Indications and Groups

The [Message Indication], or [C/C++] in the following have been changed. The part in red is identified for change.

ID	Message Indication	Gr	C/C++
pgr0705	The basic type (int/short/char/long/double/float/_Bool) is used in this file.	g	C
pgr0710	Multi-byte characters or half-angle kana characters are used. (@1)	g	C
pgr0722	Compound operator "=" is used in condition expression @1.	g	C/C++

3.3.2 Improvement for cast expression

The cast expression of integer type constant now can be checked as the constant.

[Sample]

```
short s = (int)1000000; /*Message pgr0381*/
```

3.3.3 Improvements for pgr0672

In the expression of "Return expression", In order to explain the message output caused by the recording of function type or the pointer that points to the function, the keyword of return has been embedded in the output message.

3.3.4 Improvements for pgr0695

The operation for the operator of dynamic_cast, const_cast, static_cast, and reinterpret_cast can be checked as cast.

3.3.5 Change of Message ID and Group for pgr0449

The following problems originally existed in pgr0449: Indication occurred in the standard header.

"/include/clusapi.h"(1497) ! pgr0449 member name is not recorded.

pgr0449 is changed to pgr0259, and its group is changed from ! to b, so that the above indication will not be output.

	ID	Indication message	Gr	C/C++
Before change	pgr0449	member name is not recorded.	!	C/C++
After change	pgr0259	member name is not recorded	b	C/C++

3.3.6 Division of Indication Message

pgr0374 is divided into two pieces, and pgr0455 is made.

When the #error statement is recorded in the file of the standard include directory, it will change to pgr0455(!group) and the processing will be cancelled because of a high probability of parse failure of PGRelief.

If it is not under the standard include directory, it is the same as before, which will indicate pgr0374(bgroup), and continue with the following processing.

ID	Message	Gr	C/C++
pgr0374	#error @1	b	C/C++
pgr0455	#error @1	!	C/C++

pgr0309 is divided into two pieces, and pgr0342 is made.

For the main function, it is changed to pgr0342(agroup) because of its high probability of problem occurrence.

For the non-main function, it will indicate pgr0309(bgroup) as before.

ID	Message	Gr	C/C++
pgr0309	The function @1 has no return value and should be declared as void.	b	C/C++
pgr0342	No return value generated in Function @1.	a	C/C++

pgr0038 is divided into two pieces, and pgr0058 is made.

When the outside of == is return statement or the list of argument, it is changed to pgr0058 because of its low probability of problems comparing to others.

If the outside of = is the content other than return statement or list of argument, then it will indicate pgr0038 as before.

[Sample]

```
void func(int);  
:  
func( x == 0 ); /*It indicates pgr0038 before, and changed to pgr0058 now*/
```

ID	Message	Gr	C/C++
pgr0038	In @1, == may have been mistakenly used for =.	a	C/C++
pgr0058	Because == other than = is used in @1, return value or argument is a bool.	b	C/C++

4. Improvements for PGRelief 2007

This section describes the improvements and changes that have been made from "COReTOOL/PGRelief C/C++ for Windows V3.1" to "PGRelief 2007".

Also, improvements in the previous version are also included in this version. For the improvements in the previous version, please refer to "1. Improvement for PGRelief V2.0", "2. Improvements for PGRelief V3.0", and "3. Improvements for PGRelief V.3.1".

4.1 Addition of Indications

The following 18 pieces of indication have been added.

With the addition of indication, new group "n" (message that indicates the record that violates the naming rule) is also added.

Attention

In the PGRelief of old version, when the n group is defined at the user side, it will be duplicated with the n group added in this version. Please execute the following operations.

Users who use GUI

Launch the PGRelief of previous version, in the [group] on the left of [Fixed Project Option Settings] of the [Settings] menu, change the n group to another group, and then follow the PGRelief of this version.

Users who use command

Please change the n in the group change file specified with -G option to other lowercase English letters.

ID	Message	Gr	C/C++
pgr0080	The result of the constant expression "@1" exceeds the width of type @2.	a	C/C++
pgr0801	Comma is used in the declaration.	g	C/C++
pgr0802	Bit-field member is declared in struct "@2".	g	C/C++
pgr0806	@1 @2 is declared when declaring the @3 tag.	g	C
pgr0811	An error may exist in the controlling expression "@1" of the switch statement.	b	C/C++
pgr0812	Integer constant expression is used as the expression of the case label in this switch statement.	g	C
pgr0821	Because of type @3, "@2" of cast expression "@1" is converted into the type specified in the cast expression with sign conversion.	a	C/C++
pgr0822	The type of corresponding argument "@3" is not matched with "@2" in the format string of function "@1". (parameter : @4, expected type : @5)	a	C/C++
pgr0823	The number of corresponding arguments is different from the number of necessary arguments in the format string of function "@1". (number of % conversion instruction introduced : 1, number of corresponding arguments : @2, number of necessary arguments : @3)	a	C/C++
pgr0831	This case label may be of no effect. (constant expression of the case label : @1, control expression of the switch statement : @2, enum type declaration in line @4 of "@3")	a	C
pgr0832	The terminal character of field "@2" may not be set in the function call "@1". (filed : @3, size : @4)	a	C/C++
pgr0833	The return value "@3" of function "@2" at line @1 point to a static field, it	a	C/C++

	is an error to free it by @4 "@5".		
pgr0834	An error may exist in "@1" which trying to get the size of a pointer type.	a	C/C++
pgr1301	The @1 variable @2 is not following the naming rule (@3).	n	C/C++
pgr1302	The @1 @2 is not following the naming rule (@3).	n	C/C++
pgr1303	The macro @1 is not following the naming rule (@2).	n	C/C++
pgr1304	The function @1 is not following the naming rule (@2).	n	C/C++
pgr1305	The @1 @2's name is composed of nothing besides the prefix.	n	C/C++

4.2 Division of Indication Messages

pgr0752 is divided into two pieces, and pgr0801 is made.

Initially, if using comma in the initialization expression and update expression, then pgr0791 will be indicated, and pgr752 will be indicated if using comma at other places. Not it is changed to: if using comma in the initialization expression and update expression, then pgr0791 will be indicated, if using comma in declaration statement, pgr0801(New) will be indicated, and pgr752 will be indicated if using comma at other places.

ID	Message	Gr	C/C++
pgr0752	@1 uses a comma outside an initialization expression or update expression in a for statement.	g	C/C++
pgr0801	Comma is used in the declaration.	g	C/C++

4.3 Addition of Naming Rule Check Function

The check of whether the variable name and function name has been recorded according to the naming rule is now available.

4.4 Addition of Corresponding IPA/SEC-C Coding Rule

The check of the following IPA/SEC-Ccoding rule is now available.

M2.2.4

M4.3.1

M4.5.2

P1.3.3(1),P1.3.3(2),P1.3.3(3)

4.5 #pragma once Grammar Parse

"#pragma once" special grammar of Microsoft compiler is not supported.

4.6 Changes for command

4.6.1 Addition of Category of –K option

The –K option that supports the following compilers has been added.

Compiler of Microsoft

compiler		- K option
Microsoft Visual	Unused MFC	-KMS/VC2003
C++ .NET 2002	Used MFC	-KMS/VC2003MFC
Microsoft Visual	Unused MFC	-KMS/VC2003
C++ .NET 2003	Used MFC	-KMS/VC2003MFC

Compiler of RENESAS

compiler	-K option
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

Compiler of NEC

compiler	-K option
78K0	-KNEC/78K0
78K0S	-KNEC/78K0S
78K4	-KNEC/78K4
V850	-KNEC/V850

4.6.2 Addition of label that can be recorded in the identifier file of –F option

In the identifier file of –F option, the following 3 labels is now available to be recorded.

Label name	Meaning	Message
FORMATTED_OUTPUT_CONVERSION_FUNCTION	Information of output function with format	pgr0822 pgr0823
MEMORY_COPY_CHECK_FUNCTION	Information of area copy function that specifies the copy size.	pgr0832
POINT_STATIC_AREA_FUNCTION	Information of function that returns to the static area.	pgr0833

4.6.3 Addition of --temp-file option

The option for specifying the saving target location of parse results (ao file and fd file etc.) has been added.

4.6.4 Addition of -N option

The option for specifying the naming rule file has been added.

4.7 Support for GB2312

GB2312 has been added in the code category of input source and output indication messages.

5. Improvements for PGRelief 2007 autumn

This section describes the improvements and changes that have been made from "PGRelief 2007" to "PGRelief 2007 autumn".

Also, improvements in the previous version are also included in this version. For the improvements in the previous version, please refer to "1. Improvement for PGRelief V2.0", "2. Improvements for PGRelief V3.0", "3. Improvements for PGRelief V.3.1", and "4. Improvements for PGRelief 2007".

5.1 Changes for command

5.1.1 Addition of Category of -K option

The -K option that supports the following compilers has been added.

Compiler of Microsoft

compiler		- K option
Microsoft Visual C++ 2005	Unused MFC	-KMS/VC2005
	Used MFC	-KMS/VC2005MFC

6. Improvements for PGRelief 2008

This section describes the improvements and changes that have been made from "PGRelief 2007 autumn" to "PGRelief 2008".

Also, improvements in the previous version are also included in this version. For the improvements in the previous version, please refer to "1. Improvement for PGRelief V2.0", "2. Improvements for PGRelief V3.0", "3. Improvementa for PGRelief V.3.1", "4. Improvements for PGRelief 2007" and "5. Improvements for PGRelief 2007 autumn".

6.1 Addition of Indications

The following 29 pieces of indication have been added.

ID	Message	Gr	C/C++
pgr2201	The object state of the member function @3 of @1 @2 is not updated and can be qualified with const.	b	C++
pgr2202	The virtual function @1 (defined in line @3 of @2) is called in @4 @5.	b	C++
pgr2203	The value of the non-static member variable @1 is assigned in the constructor of @2 @3. The initialization should be used in the constructor initializer.	c	C++
pgr2204	@1 @2 including the data member of pointer type does not declare the copy constructor or copy assignment operator explicitly.	b	C++
pgr2205	@1 @2 defines the destructor explicitly, but does not define the copy constructor or copy assignment operator explicitly.	b	C++
pgr2206	The destructor of the base class @1 containing the virtual function must be specified as virtual.	a	C++
pgr2207	The operator @2 constituting a pair with the operator @1 is not overloaded.	b	C++
pgr2208	The base class @1 is not an abstract class.	g	C++
pgr2209	The non-virtual function @3 of @1 @2 is redefined in @4 @5 (in line @7 of @6).	b	C++
pgr2210	The function @3 of @1 @2 changes the value of the default parameter in the virtual function @6 of @4 @5 (in line @8 of @7).	a	C++
pgr2211	Overload of the @1 operator might change the evaluation order.	a	C++
pgr2212	The cast expression @1 in the C style should be avoided in C++.	g	C++
pgr2213	The operator @2 constituting a pair with the operator @1 is not overloaded.	a	C++
pgr2215	@3 is executed for @2 of the pointer type (@1).	b	C++
pgr2216	The macro function @1 can be replaced by the inline function or the template function.	b	C++
pgr2217	The macro @1 can be replaced by a const or an enum constant.	b	C++
pgr2218	The declaration or definition should be written in the namespace to prevent the name conflict.	b	C++
pgr2219	The exception @1 should be thrown in the form of a value.	b	C++
pgr2220	The exception should be caught by reference.	b	C++
pgr2221	The argument @1 is used when the exception is thrown again.	a	C++
pgr2222	The copy constructor of @1 @2 does not copy all the data members.	a	C++

pgr2223	The copy assignment operator of @1 @2 does not copy all the data members.	a	C++
pgr2224	The member variable @3 allocated in @1 @2 should be deleted in the destructor.	a	C++
pgr2225	The variable @2 generated in the form of @1 is destroyed in the form of @3.	a	C++
pgr2226	@2 containing new exists in the return statement of the function @1.	g	C++
pgr2227	The argument @1 that is not updated can be specified with const reference type. (argument : No. @2 , type: @3)	b	C++
pgr2228	The new operators overloaded in @1 @2 do not declare all the forms.	g	C++
pgr2229	A using directive or declaration is used in a header. (file name: @1, description: @2)	b	C++
pgr2230	The operator() function of @1 @2 has multiple overload functions.	g	C++

6.2 Improvements for Message Indication

Expand the following objects to be pointed out from the source file in the C language to that in the C/C++ language.

ID	Message	Gr	C/C++
pgr0016	Return value @1 exceeds the range of the return type of the function @2 (return value : @3, return type: @4).	a	C/C++
pgr0019	Assignment of @2 of type @1 with the constant @3 that exceeds the bit width of type @1 is a mistake.	a	C/C++
pgr0066	The @2 argument @3 of the function @1 exceeds the bit width of the corresponding parameter @4 in the function definition in line @6 of @5 (argument: @7, parameter: @8).	a	C/C++
pgr0067	The @2 argument @3 of the function @1 exceeds the bit width of the corresponding parameter @4 in the function declaration in line @6 of @5 (argument: @7, parameter: @8).	a	C/C++
pgr0246	The cast expression @1 casts a pointer type to a smaller type (cast type: @2 , expression: @3).	a	C/C++
pgr0381	The initial value @1 exceeds the bit width of @3 of type @2.	a	C/C++
pgr0712	There is no case label in this switch body.	a	C/C++

6.3 Addition of support for Compiler

The -K option that supports the following compilers has been added.

-K ARM_LTD/REALVIEW_V3_1

The reserved word of ARM REALVIEW V3.1 compiler is supported.

6.4 Support to UTF-8

UTF-8 has been added in the encode category of input source and output indication messages.

6.5 Prohibition of the Indication Message Output

Append the function of prohibiting the output to the check-item definition file defining the indication of an output object. The function is not appended in the check-item definition files corresponding to the

MISRA-C and IPA/SEC-C rules.

7. Improvements for PGRelief 2008 autumn

This section describes the improvements and changes that have been made from "PGRelief 2008 " to "PGRelief 2008 autumn".

Also, improvements in the previous version are also included in this version. For the improvements in the previous version, please refer to "1. Improvement for PGRelief V2.0", "2. Improvements for PGRelief V3.0", "3. Improvements for PGRelief V.3.1", "4. Improvements for PGRelief 2007", "5. Improvements for PGRelief 2007 autumn" and "6. Improvements for PGRelief 2008".

7.1 Addition of Indications

The following 27 pieces of indication have been added.

ID	Message	Gr	C/C++
pgr0805	In this file, a basic arithmetic type (int/short/char/long/double/float) is used directly.	g	C++
pgr0835	The size of the copy source of the copy expression @1 might exceed that of the copy target @2 (size of copy target: @3, size of copy source: @4).	a	C/C++
pgr0836	@1 is compared with irrelative type.(enum type : @2, compared type : @3)	b	C/C++
pgr0837	@1 is compared with irrelative type. (enum type:@2 , compared type: @3)	b	C/C++
pgr0838	@1 is compared with irrelative value.(enum type: @2 , compared value: 0)	a	C/C++
pgr0839	@1 is compared with irrelative value.(enum type: @2 , compared value: @3)	a	C/C++
pgr0840	Resources allocated with the operator "new" in line @1 may not have been deallocated.	a	C++
pgr2231	The namespace @1 exceeds 2 levels.	g	C++
pgr2232	@1 @2 has the conversion function.	g	C++
pgr2233	@1 @2 has the single-argument constructor.	g	C++
pgr2236	The function operator= does not return "**this".	g	C++
pgr2237	The format of allocation for variable @1 does not conform to the format of deallocation. (allocation : @2, deallocation : @3)	a	C++
pgr2238	The @2 type has been passed to the argument of function @1.	a	C++
pgr2239	The catch handler for base class @1 exists before the catch handler (in line @4 of @3) for derived class @2.	a	C++
pgr2240	The end of catch handler is not catch(...).	b	C++
pgr2241	The end of catch handler of the function @1 is not catch(...).	a	C++
pgr2242	The postfix operator @1 was declared.	c	C++
pgr2243	The member function was declared in struct @1.	g	C++
pgr2244	The anonymous namespace was defined in the header file @1.	c	C++
pgr2248	The @1 @2 should catch exceptions.	b	C++
pgr2249	The array of derived class is passed to the base class @1. (function: @2; argument: No. @3; type of argument: @4; type of parameter: @5; declaration : line @6 of @7)	a	C++

pgr2250	The derived class that was ensured by array form is passed to the base class @1. (function: @2; argument: No. @3; type of argument: @4; type of parameter: @5; declaration: line @6 of @7)	a	C++
pgr2251	dynamic_cast should be used for the cast @1 to derived class.(after conversion : @2, before conversion : @3)	b	C++
pgr2252	The non-virtual function group @5 of @3 @4 has been hidden by @1 @2.	b	C++
pgr2253	A member without set with a value has been referenced in the member initialization list.	a	C++
pgr2254	The exception specification is used.	g	C++
pgr2255	NULL is thrown.	b	C++

7.2 Improvements for Message Indication

Expand the following objects to be pointed out from the source file in the C language to that in the C/C++ language.

ID	Message	Gr	C/C++
pgr0351	In the assignment expression @1, the size of the expression on the right is larger than that of the expression on the left, so the correct value might not be assigned (left : @2 , right : @3).	b	C/C++
pgr0369	The correct value might not be returned because the type size of the return value @1 is larger than that of the function @2 (return value: @3 , function type: @4).	b	C/C++
pgr0383	The initial value @2 of type @1 is greater than @4 of type @3, so the correct value might not be passed.	b	C/C++
pgr0397	The size of the @2 argument @3 of the function @1 is larger than that of the corresponding parameter @4 in the function definition in line @6 of @5, so the correct value might not be passed (argument: @7, parameter: @8).	b	C/C++
pgr0398	The size of the @2 argument @3 of the function @1 is larger than that of the corresponding parameter @4 in the function declaration in line @6 of @5, so the correct value might not be passed (argument: @7, parameter: @8).	b	C/C++
pgr0643	Initialization of the array/structure/union @1 is inconsistent with its makeup.	g	C/C++
pgr0661	The cast expression @1 casts a non-pointer type to a pointer type (cast type: @2, expression type: @3).	g	C/C++
pgr0662	The cast expression @1 casts a pointer type to a non-pointer type (cast type: @2, expression type: @3).	g	C/C++
pgr0663	The cast expression @1 casts a pointer type to a different pointer type (cast type: @2, expression type: @3) .	g	C/C++
pgr0707	The specifier "register" is used in the function @1.	g	C/C++
pgr0711	The keyword "continue" is used (location of relevant loop statement: line @2 of @1).	g	C/C++
pgr0713	An expression @1 irrelevant to loop control exists in a for statement.	g	C/C++
pgr0714	The loop counter @1 of a for statement is updated in the loop body (for statement position: line @3 of @2).	g	C/C++

pgr0716	The function @1 is a recursive function.	g	C/C++
pgr0717	The parameter name @1 is inconsistent with the parameter name in the function definition @4 in line @3 of @2.	g	C/C++
pgr0718	There is a member in union @1 whose type is different from that of the float-type member @2.	g	C/C++
pgr0719	The function @1 has 2 or more exits.	g	C/C++
pgr0721	The expression @1 compares two object addresses of different types (@2 , @3).	b	C/C++
pgr0725	The pointer @1 exceeds 2 levels.	g	C/C++
pgr0729	@1 uses the confusing suffix "!".	b	C/C++
pgr0730	volatile @1 has been used in sizeof, but the space will not be accessed.	g	C/C++
pgr0731	@1 is not a bool value (a non-bool value is assigned in line @3 of @2).	g	C/C++
pgr0732	The label default has been included before the label case in a switch statement.	g	C/C++
pgr0733	Function @1 is defined without prior declaration.	b	C/C++
pgr0735	The variable @2 is not checked immediately following the function call @1.	b	C/C++
pgr0742	In ANSI, the operand @2 of the bit operation @1 must be of an integer type.	b	C/C++
pgr0753	The label @1 is not within the { } immediately following the switch.	b	C/C++
pgr0754	The result of @1 is used in another operation.	g	C/C++
pgr0763	Macro @1 is operated by @2 in a block.	b	C/C++
pgr0771	The unsigned value @1 is used without the suffix "U".	g	C/C++
pgr0772	Two or more break statements appear in a loop.	g	C/C++
pgr0773	@1 might exceed the bit width of the underlying type @3 of @2.	g	C/C++
pgr0774	@1 negates the const/volatile qualifying the space pointed to by the pointer @3 of type @2.	g	C/C++
pgr0779	Pointer arithmetic operation @1 is not in the form of array indexing.	g	C/C++
pgr0780	A union is used.	g	C/C++
pgr0781	The cast expression @1 is a conversion involving a function pointer type and a non-integer type (cast type: @2, expression type: @3).	g	C/C++
pgr0783	@2 in the pointer arithmetic operation @1 is not a pointer pointing to an array.	g	C/C++
pgr0784	@2 in the array @1 is not a pointer pointing to an array.	g	C/C++
pgr0785	@2 and @3 of the pointer subtraction @1 do not point to the same array.	g	C/C++
pgr0790	@2 in the @1 expression is not a primary expression.	g	C/C++
pgr0791	@1 uses a comma in the initialization expression or update expression of a for statement.	g	C/C++
pgr0792	Floating-point comparison @1 might lead to an unexpected result.	b	C/C++
pgr0793	A goto statement is used.	g	C/C++
pgr0812	Integer constant expression is used as the expression of the case label in this switch statement.	g	C/C++

7.3 Improvement of GUI analysis option setting

The object file can only be used in pgr5 command previously, and now can be used in GUI.

8. Improvements for PGRelief 2009

This section demonstrates the changes and improvements made from [PGRelief 2008 autumn] to [PGRelief C/C++ 2009].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to "1. Improvements for PGRelief V2.0", "2. Improvements for PGRelief V3.0", "3. Improvements for PGRelief V3.1", "4. Improvements for PGRelief 2007", "5. Improvements for PGRelief 2007 autumn", "6. Improvements for PGRelief 2008", and "7. Improvements for PGRelief 2008 autumn".

8.1 Addition of Indications

The following 9 indications are added.

ID	Message Indication	Gr	C/C++
pgr0841	The initialization of variable @1 which is declared in the switch statement is not implemented.	a	C/C++
pgr0842	The statement in the switch statement is not implemented.	a	C/C++
pgr0843	The variable @1 whose value is not set is referred.	a	C/C++
pgr0844	The variable @1 whose value is not set is used as argument of function. (function name : @2, function declaration : line @4 of @3)	g	C/C++
pgr0845	The body of if statement or else statement is not implemented.	b	C/C++
pgr0846	There's preprocessing directive @1 which is not defined with ANSI in the place that becomes invalid because of a conditional taking.	g	C/C++
pgr0847	The format of #pragma @1 is wrong.	b	C/C++
pgr0850	Even if argument is assigned in the function, parameter can not be updated in function call.(expression : @1, function : @2, argument : @3)	b	C
pgr0854	The sizeof() operation that is operated for argument @1 of array type may be unreasonable.	a	C

8.2 Extension of function for parse time reduction

The pre-compiling header function that can reduce parse time has been extended.

The old pre-compiling header function can only be used on the compilers of "Microsoft Visual C++ Compiler Products" through GUI. Now, the constraint of compiler use has been removed, and it can be used through GUI and pgr5 command.

8.3 Addition of new command (pgrfake)

The pgrfake command has been added. This command can automatically select the options required for parsing from the command parameters of compiler. It can also execute the pgr5 command.

If this command is used in the environment where a set of compiling work such as defined make file or batch file has been defined, it is no need to set options by the pgr5 command.

8.4 Link of Command execution result and GUI

The execution result of the pgr5 command or pgrfake command can be displayed in the indication message display window (GUI).

8.5 Addition of support for Compiler

The -K option that supports the following compilers has been added.

-K ARM_LTD/REALVIEW_V4_0

The reserved word of ARM REALVIEW V4.0 compiler is supported.

9. Improvements for PGRelief 2009 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2009] to [PGRelief C/C++ 2009 autumn].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to “1. Improvements for PGRelief V2.0”, “2. Improvements for PGRelief V3.0”, “3. Improvements for PGRelief V3.1”, “4. Improvements for PGRelief 2007”, “5. Improvements for PGRelief 2007 autumn”, “6. Improvements for PGRelief 2008”, “7. Improvements for PGRelief 2008 autumn”, and “8. Improvements for PGRelief 2009”.

9.1 Addition of Indications

The following 7 indications are added.

ID	Message	Gr	C/C++
pgr0860	The array @1 exceeds the range (array declaration: line @3 of @2, declaration pattern: @4).	a	C/C++
pgr0861	The array @1 exceeds the range (array declaration: line @3 of @2, declaration pattern: @4).	a	C/C++
pgr0862	The array @1 at line @2, whose @3 is assigned with @4, exceeds the range of @5 (array declaration: line @7 of @6, declaration pattern: @8).	a	C/C++
pgr0863	The array @1 at line @2, whose @3 is assigned with @4, exceeds the range of @5 (array declaration: line @7 of @6, declaration pattern: @8).	a	C/C++
pgr0864	Because of type char, specified char @2 of cast expression @1 is converted into the type specified in the cast expression with sign conversion.	b	C/C++
pgr0865	Resources point @4 allocated with the function @3 in line @2 of function @1 is returned.	b	C/C++
pgr0866	Global point @1 to resources which was allocated with the function @3 in line @2 of function @4 may not have been deallocated.	b	C/C++

Notes:

- The added pgr0860 and pgr0861 are created by dividing the detection condition of existing pgr0012. Therefore, in the source where pgr0012 is detected, pgr0860 or pgr0861 will also be detected.
- The added pgr0862 and pgr0863 are created by dividing the detection condition of existing pgr0013. Therefore, in the source where pgr0013 is detected, pgr0862 or pgr0863 will also be detected.

9.2 Addition of support for Compiler

The following options have been added to the -K option that designates the compiler specification.

- Compiler of Microsoft Corporation

Compiler		-K option
Microsoft Visual	MFC is not used	-KMS/VC2008
C++ 2008	MFC is used	-KMS/VC2008MFC

10. Improvements for PGRelief 2010

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2009 autumn] to [PGRelief C/C++ 2010].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to "1. Improvements for PGRelief V2.0", "2. Improvements for PGRelief V3.0", "3. Improvements for PGRelief V3.1", "4. Improvements for PGRelief 2007", "5. Improvements for PGRelief 2007 autumn", "6. Improvements for PGRelief 2008", "7. Improvements for PGRelief 2008 autumn", "8. Improvements for PGRelief 2009" and "9. Improvements for PGRelief 2009 autumn".

10.1 Addition of Wide-ranging Detective function

The addition of Wide-ranging Detective is that tracking the route of function invocation and indicate the position where failures may occur. In addition, when using the Wide-ranging Detective function, the license of Wide-ranging Detective Option is needed.

About the Wide-ranging Detective indication in "PGRelief C/C++ 2010" please refer to [10.2 Addition of Indications].

10.2 Addition of Indications

The following 11 indications are added.

ID	Message	Gr	C/C++	Remark
pgr0083	The variable @2, assigned with the 0 address in line @1, might access the 0 address.	a	C	member
pgr0084	The variable @1, which may be 0, is used as a divisor.	a	C	member
pgr0510	The 0 address might be referenced because the variable @2 might have been assigned with the 0 address in line @1.	a	C	member
pgr0699	The variable @2 compared with the 0 address in line @1 might access the 0 address.	a	C	member
pgr0868	The index @2 of array @1 may exceed the range of array. (array declaration: line @4 of @3)	b	C/C++	
pgr0871	The index @2 of array @1 in loop may exceed the array boundary. (array declaration: line @4 of @3)	b	C/C++	
pgr0873	The character pointer that renewed in loop doesn't check if it is the terminal character.	b	C/C++	
pgr5011	The argument @3 at the line @2 of @1 accessed to 0 address. (Route: @4)	a	C	Wide-ranging Detective
pgr5021	The argument @3 at the line @2 of @1 accessed outside range of array. (Route: @4)	a	C	Wide-ranging Detective
pgr5031	The resource allocated by @1 is not freed. (Route: @2)	a	C	Wide-ranging Detective
pgr5041	The resource allocated by @1 has been freed repeatedly. (Route: @2)	a	C	Wide-ranging Detective

The meaning of keywords in the Remark Field:

Wide-ranging Detective: message output when the license of Wide-ranging Detective Option exists.

member : message assigned a new ID after performing member variable extension for the original indication target.

11. Improvements for PGRelief 2010 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2010] to [PGRelief C/C++ 2010 autumn].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to “1. Improvements for PGRelief V2.0”, “2. Improvements for PGRelief V3.0”, “3. Improvements for PGRelief V3.1”, “4. Improvements for PGRelief 2007”, “5. Improvements for PGRelief 2007 autumn”, “6. Improvements for PGRelief 2008”, “7. Improvements for PGRelief 2008 autumn”, “8. Improvements for PGRelief 2009”, “9. Improvements for PGRelief 2009 autumn”, and “10. Improvements for PGRelief 2010”.

11.1 Check for IPA/SEC-C++ Coding Operation

The check for IPA/SEC-C++ coding operation is available now.

12. Improvements for PGRelief 2011

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2010 autumn] to [PGRelief C/C++ 2011].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to "1. Improvements for PGRelief V2.0", "2. Improvements for PGRelief V3.0", "3. Improvements for PGRelief V3.1", "4. Improvements for PGRelief 2007", "5. Improvements for PGRelief 2007 autumn", "6. Improvements for PGRelief 2008", "7. Improvements for PGRelief 2008 autumn", "8. Improvements for PGRelief 2009", "9. Improvements for PGRelief 2009 autumn", "10. Improvements for PGRelief 2010", and "11. Improvements for PGRelief 2010 autumn".

12.1 Addition of Indications

The following 16 indications are added.

ID	Message	Gr	C/C++
pgr0365	The left operand and right operand of the conditional expression @1 are of different types : pointer and non-pointer(left : @2, right : @3).	b	C
pgr0872	Variable @1 may be referenced before it has been set with a value.	a	C
pgr0875	Switch statement has multiple default labels.	a	C/C++
pgr0876	The size of the copy source of the copy expression @1 might exceed that of the copy target @2.(size of copy target: @3, size of copy source: @4)	a	C/C++
pgr0877	There may be a mistake if unsigned value compared with the condition that less than or equal to 0.	g	C/C++
pgr0878	If char is unsigned, the condition expression @1 is meaningless.	g	C/C++
pgr2256	The macro @1 can be replaced by a string literal.	g	C++
pgr2257	Overload of the "," operator might change the evaluation order.	g	C++
pgr2259	The pointer "this" is deleted.	a	C++
pgr2260	Operator "delete []" has described the size inside.	g	C++
pgr2261	The cast expression @1 performs a cast to a smaller type. (cast type : @2, expression : @3)	d	C++
pgr2264	The data member @1 share the same name with one in base class @2 (at line @4 of @3).	b	C++
pgr2265	The pointer "this" has been used in the member initialization list.	a	C++
pgr2266	Not all data members has been initialized in constructor.	b	C++
pgr2267	The instance is transferred from derived class to the base class. (function : @1; argument : No.@2; type of argument : @3; type of parameter : @4; declaration : line @6 of @5)	b	C++
pgr2269	Not all data members has been initialized in constructor.	b	C++

12.2 Improvements of Indications

The following indications are extended from C source file to C/C++ source file.

ID	Message	Gr	C/C++
pgr0350	The two sides of the assignment expression @1 are of different types: signed and unsigned (left : @2 , right : @3).	d	C/C++

pgr0368	The return value @1 and the function @2 are of different types: one is signed and the other is unsigned (return value :@3, function type : @4).	d	C/C++
pgr0384	The initial value @2 of type @1 and @4 of type @3 are of different types: signed and unsigned.	d	C/C++
pgr0395	The @2 argument @3 of the function @1 and the corresponding parameter @4 in the function definition in line @6 of @5 are of different types: signed and unsigned (argument: @7, parameter: @8).	d	C/C++
pgr0396	The @2 argument @3 of the function @1 and the corresponding parameter @4 in the function declaration in line @6 of @5 are of different types: signed and unsigned (argument: @7, parameter: @8).	d	C/C++
pgr0708	The value of the unsigned expression @1 cannot be represented by an unsigned type.	g	C/C++
pgr0728	The subtraction @1 is performed with a signed and an unsigned operand.	g	C/C++
pgr0740	When char is treated as unsigned, the subtraction @1 is performed with the combination of a signed and an unsigned operand.	g	C/C++

13. Improvements for PGRelief 2011 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2011] to [PGRelief C/C++ 2011 autumn].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to “1. Improvements for PGRelief V2.0”, “2. Improvements for PGRelief V3.0”, “3. Improvements for PGRelief V3.1”, “4. Improvements for PGRelief 2007”, “5. Improvements for PGRelief 2007 autumn”, “6. Improvements for PGRelief 2008”, “7. Improvements for PGRelief 2008 autumn”, “8. Improvements for PGRelief 2009”, “9. Improvements for PGRelief 2009 autumn”, “10. Improvements for PGRelief 2010”, “11. Improvements for PGRelief 2010 autumn”, and “12. Improvements for PGRelief 2011”.

13.1 Addition of Indications

The following 1 indications are added.

ID	Message	Gr	C/C++
pgr0824	The No. @2 argument @3 passed to the function @1 is incorrect.	a	C/C++

14. Improvements for PGRelief 2012

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2011 autumn] to [PGRelief C/C++ 2012].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to "1. Improvements for PGRelief V2.0", "2. Improvements for PGRelief V3.0", "3. Improvements for PGRelief V3.1", "4. Improvements for PGRelief 2007", "5. Improvements for PGRelief 2007 autumn", "6. Improvements for PGRelief 2008", "7. Improvements for PGRelief 2008 autumn", "8. Improvements for PGRelief 2009", "9. Improvements for PGRelief 2009 autumn", "10. Improvements for PGRelief 2010", "11. Improvements for PGRelief 2010 autumn", "12. Improvements for PGRelief 2011", and "13. Improvements for PGRelief 2011 autumn".

14.1 Addition of Indications

The following 5 indications are added.

ID	Message	Gr	C/C++
pgr0388	The initial value @1 exceeds the max value @4 of @3 of type @2.	b	C/C++
pgr0584	In @1, the range of space @3 is less than the size of @2.(space: @4, size: @5)	a	C/C++
pgr0585	@1 might access an address which exceeds the range of space @2. (space: @3, size: @4)	a	C/C++
pgr0697	An infinite loop might result because type width of the loop counter @1 (of type @2) is less than the type width of the expression @3 (of type @4) with which it is compared.	a	C/C++
pgr0869	The index @2 of array @1 may exceed the range of array.(array declaration: line @4 of @3)	a	C/C++

14.2 Addition of --qm option to pgr5, pgrmisra and pgrsec commands

In response to the "analysis result consolidation mode" added in the "PGRelief 2012 Quality Management Option", the --qm option has been added to the pgr5, pgrmisra and pgrsec commands. For the analysis result consolidation mode of the Quality Management Option, please refer to "2.3.2 Procedure of using under analysis result consolidation mode" in "Quality Management Option Manual".

15. Improvements for PGRelief 2012 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2012] to [PGRelief C/C++ 2012 autumn].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to “1. Improvements for PGRelief V2.0”, “2. Improvements for PGRelief V3.0”, “3. Improvements for PGRelief V3.1”, “4. Improvements for PGRelief 2007”, “5. Improvements for PGRelief 2007 autumn”, “6. Improvements for PGRelief 2008”, “7. Improvements for PGRelief 2008 autumn”, “8. Improvements for PGRelief 2009”, “9. Improvements for PGRelief 2009 autumn”, “10. Improvements for PGRelief 2010”, “11. Improvements for PGRelief 2010 autumn”, “12. Improvements for PGRelief 2011”, “13. Improvements for PGRelief 2011 autumn”, and “14. Improvements for PGRelief 2012”.

15.1 Addition of support for Compiler

The `-K` option that supports the following compilers has been added.

`-K RENESAS/78K0R`

The reserved word of renesas 78K0R compiler is supported.

`-K RENESAS/RX`

The reserved word of renesas RX compiler is supported.

16. Improvements for PGRelief 2013

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2012 autumn] to [PGRelief C/C++ 2013].

In addition, improvements in the previous version are also contained in this version. For improvements in the previous version, please refer to "1. Improvements for PGRelief V2.0", "2. Improvements for PGRelief V3.0", "3. Improvements for PGRelief V3.1", "4. Improvements for PGRelief 2007", "5. Improvements for PGRelief 2007 autumn", "6. Improvements for PGRelief 2008", "7. Improvements for PGRelief 2008 autumn", "8. Improvements for PGRelief 2009", "9. Improvements for PGRelief 2009 autumn", "10. Improvements for PGRelief 2010", "11. Improvements for PGRelief 2010 autumn", "12. Improvements for PGRelief 2011", "13. Improvements for PGRelief 2011 autumn", "14. Improvements for PGRelief 2012", and "15. Improvements for PGRelief 2012 autumn".

16.1 Addition of Indications

The following 9 indications are added.

ID	Message	Gr	C/C++
pgr0881	Not specified by const but referred string literal.	g	C/C++
pgr0882	There is no static specification in the inline function @1.	g	C
pgr0883	There is a pointer that modified by restrict to the parameter of the function @1	g	C
pgr0884	Jump out of the scope of goto statement.(jump position : line @1)	b	C/C++
pgr0885	It jumps from the goto statement to the previous line.(jump position : line @1)	b	C/C++
pgr0886	There is a parameter of function @1 which is array described by keyword static.	g	C
pgr0887	File @1 maybe has already been opened with @3 mode in line @2.	a	C/C++
pgr0888	Maybe write into the file @1 which is opened with read-only mode.(file open line : line @2)	a	C/C++
pgr0889	@1 is an expression of FILE pointer which is back referred.	g	C/C++

16.2 Addition of support for Compiler

The -K option that supports the following compilers has been added.

-K MS/VC2010

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

-K MS/VC2010MFC

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

16.3 Addition of Evidence output function

The function to output the evidence of which it input the analytical result of the source program has been added.

Please refer to "PGRelief Usage" - "Output Evidence" of PGRelief help for details.

16.4 Addition of Diagnosis report generation tool

The diagnosis report generation tool that generated the source diagnosing report from the analytical result of PGRelief has been added. In PGRelief, these evidence file is input, and Diagnosis report generation tool that outputs the result of analyzing the information as a report of the Book-type of Microsoft(R) Office Excel(R) is being offered. Please use "Diagnosis report generation tool" from the start menu or the start screen (since Windows(R) 8, Windows Server(R) 2012) of Windows(R).

17. Improvements for PGRelief 2013 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2013] to [PGRelief C/C++ 2013 autumn].

In addition, improvements in the previous version are also contained in this version.

17.1 Check for MISRA-C:2012 Coding Operation

The check for MISRA-C:2012 coding operation is available now.

18. Improvements for PGRelief 2014

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2013 autumn] to [PGRelief C/C++ 2014].

In addition, improvements in the previous version are also contained in this version.

18.1 Addition of support for Compiler

The `-K` option that supports the following compilers has been added.

`-K GNU/GCC4`

The reserved word of C, C++Compiler 4.0-4.8 that GNU Compiler Collection has to offer is supported.

`-K GNU/GCC4_C90`

The reserved word of C, C++Compiler 4.0-4.8 that GNU Compiler Collection has to offer is supported. (When you specify the C90 syntax by "`-ansi`" and "`-std=`", etc.)

18.2 Addition of command multi-Executing function

The function that `pgr5`, `pgrmisra`, and the `pgrsec` commands were able to be executed up to four at the same time by one C/C++ analysis license on one terminal was added.

Please refer to "1.1.6 Environmental Variable" of 'Command Manual' for details.

18.3 Addition of 64 bit version `pgr5` and `pgrmetrics` command

`Pgr5` and `pgrmetrics` command were added and the Module to operate as a native application was added on 64 bit OS.

19. Improvements for PGRelief 2014 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2014] to [PGRelief C/C++ 2014 autumn].

In addition, improvements in the previous version are also contained in this version.

19.1 Addition of --include option to pgr5 command

The --include option has been added to the pgr5 command.

Please refer to "1.1.3. Options Instruction" of 'Command Manual' for details.

20. Improvements for PGRelief 2015

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2014 autumn] to [PGRelief C/C++ 2015].

In addition, improvements in the previous version are also contained in this version.

20.1 Addition of Indications

The following 11 indications are added.

ID	Message	Gr	C/C++
pgr0856	Compare pointer with an enumeration constant in the conditional expression @1. (pointer type : @2)	b	C
pgr0857	Compare pointer with an integer constant in the conditional expression @1. (pointer type : @2)	b	C
pgr0858	Compare pointer with an integer constant in the conditional expression @1. (pointer type : @2)	g	C
pgr1151	Before calling the function @1, the function @2 should be called.	h	C/C++
pgr1152	The @2 argument of the function @1 should not be a constant value.	h	C/C++
pgr1153	File may not be closed before spawning processes. (file open line : line @1)	h	C/C++
pgr1154	The length of variable @2 read by function @1 may be 0.	h	C/C++
pgr1155	The @2 argument of the function @1 is not specified.	h	C/C++
pgr1156	There is no checking before unsigned integer operation "@1" so that it may wrap.	h	C/C++
pgr1157	There is no checking before signed integer operation "@1" so that it may overflow.	h	C/C++
pgr1158	There is no checking of the arguments @2 and @3 to function @1 so that it may wrap when multiplied.	h	C/C++

20.2 Check for IPA/SEC-C V2 (ESCR-C V2) Coding Operation

The check for IPA/SEC-C V2 (ESCR-C V2) coding operation is available now.

21. Improvements for PGRelief 2015 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2015] to [PGRelief C/C++ 2015 autumn].

In addition, improvements in the previous version are also contained in this version.

21.1 Correspondence to the latest OS/MW environment

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

[OS Environment]

- Windows(R) 10

[Server virtualization software] * Only the floating license version corresponds.

- Microsoft(R) Windows Server(R) 2012 R2 Hyper-V(R)
- VMware vSphere(R) 5.0

21.2 Update of GUI operating environment

The Java SE Runtime Environment that GUI used was updated to Java SE Runtime Environment 8.

22. Improvements for PGRelief 2016

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2015 autumn] to [PGRelief C/C++ 2016].

In addition, improvements in the previous version are also contained in this version.

22.1 Addition of Indications

The following 10 indications of security vulnerability are added.

ID	Message	Gr	C/C++
pgr1160	The variable @1 which used in multithread function is not in exclusive control.	h	C/C++
pgr1161	No privilege change is implemented before @4 while @3 is previously called in line @2 of @1.	h	C
pgr1162	The string literal @1 may be sensitive data.	h	C/C++
pgr1163	The variable @2 which gets resource from @1 is not initialized.	h	C/C++
pgr1164	Copying environment variable into @2 in @1 may access addresses beyond array boundary.	h	C/C++
pgr1170	The @1 @2 is used. @3 should be used instead of @2.	h	C/C++
pgr1171	The @1 @2 is used. The @2 that depends on filename is weaker in dealing with TOCTOU conflict. @3	h	C/C++
pgr1172	The @1 @2 is used. The @2 is weaker in dealing with command injection.	h	C/C++
pgr1173	The @1 @2 is used. Please check whether the @2 runs with root privilege or not.	h	C/C++
pgr1174	The @1 @2 is used. If the root privilege is set before, please check whether observes correct revocation order while relinquishing privileges or not.	h	C/C++

22.2 Addition/Change of support for Compiler

The -K option that supports the following compilers has been added.

-K GNU/GCC4_C11

The reserved word of C, C++Compiler 4.0-4.8 that GNU Compiler Collection and C11 Grammar has to offer is supported.

The identifier of the following compilers (non-recommended function / recommended function etc.) was reviewed along with the addition of the indication the security vulnerability.

-K MS/VC2005

-K MS/VC2005MFC

-K MS/VC2008

-K MS/VC2008MFC

-K MS/VC2010

-K MS/VC2010MFC

22.3 Correspondence to the latest OS environment

The following has been added as the OS environment that PGRelief C/C++ operates.

[OS Environment]

- Red Hat(R) Enterprise Linux(R) 7

* Only the floating license of 64 bit version corresponds.

22.4 Offer of CI tool (Jenkins) Plugin

Plugin that cooperated with Jenkins of the CI(Continuous Integration) tool was added. The execution result of the PGRelief C/C++ came to be able to be consolidated on Jenkins, and to be displayed by applying plugin.

Please refer to 'CI Tool Collaboration Jenkins Plugin Manual'

("Etc\ci-plugin\pgrelief-jenkins-plugin_en.pdf") stored in the installation medium for details.

* Only the floating license version corresponds.

23. Improvements for PGRelief 2016 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2016] to [PGRelief C/C++ 2016 autumn].

In addition, improvements in the previous version are also contained in this version.

23.1 Check for IPA/SEC-C++ V2 (ESCR-C++ V2) Coding Operation

The check for IPA/SEC-C++ V2 (ESCR-C++ V2.0) coding operation is available now.

23.2 Check for MISRA-C:2012 Amendment 1 Coding Operation

The check for MISRA-C:2012 coding operation to contain MISRA-C:2012 Amendment 1 is available now.

24. Improvements for PGRelief 2017

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2016 autumn] to [PGRelief C/C++ 2017].

In addition, improvements in the previous version are also contained in this version.

24.1 Addition of Indications

The following 25 indications are added.

ID	Message	Gr	C/C++
pgr0803	Variable-length array @1 is declared.	g	C
pgr0804	The macro name @1 is a keyword.	g	C
pgr0807	The type @1 is not used.	g	C
pgr0808	The tag @1 is not used.	g	C
pgr0809	The macro @1 is not used.	g	C
pgr0810	Line-splicing is used in "// " comments.	g	C
pgr0813	The octal and hexadecimal escape sequences are mixed with universal characters.	g	C
pgr0814	The length of the external identifier @1 exceeds @2 characters.	b	C
pgr0815	The name of @1 and @4 at the line @3 of file @2 are only different in case.	b	C
pgr0816	The length of the identifier @1 exceeds @2 characters.	g	C
pgr0817	The length of the macro name @1 exceeds @2 characters.	g	C
pgr0818	An element of object @1 is initialized more than once.	b	C
pgr0819	The size of the array @1 is not specified explicitly when designated initializers are used to initialize its elements.	g	C
pgr0820	The type of the @2 argument @3 of the function @1 should be @4. (argument type : @5)	g	C
pgr0825	The cast expression @1 is casted between two inappropriate essential types. (cast type: @2, expression type: @3)	g	C
pgr0826	The assignment expression @1 assigns a void * pointer type to @2 type.	g	C
pgr0827	@1 is the macro function which defined in the line @3 of @2 was not replaced.	b	C
pgr0828	The initializer lists of @1 contains an expression @2 which occurred side effects.	b	C
pgr0829	The array size of the @2 argument @3 of the function @1 is different from the corresponding parameter @4 in the function declaration in line @6 of @5. (argument: @7, parameter: @8)	g	C
pgr0830	The array size of the parameter @4 in the function declaration in line @6 of @5 which corresponds to the @2 argument @3 of the function @1 is not specified. (argument: @7, parameter: @8)	g	C
pgr0848	The expression of #if or #elif is not evaluated to 0 or 1.	g	C
pgr0849	The source code may be commented out.	g	C

pgr1237	It is not desirable to try to compare string by function @1.	b	C
pgr1239	The type of the @2 argument @3 and the @4 argument @5 of the function @1 are different. (@2 argument : @6, @4 argument : @7)	b	C
pgr1395	The literal @1 is used. (literal type @2)	g	C

24.2 Correspondence rate improvement of MISRA-C:2012 Coding Operation

It corresponded to all rules of MISRA-C:2012(contain MISRA-C:2012 Amendment 1). Please refer to "3.3 MISRA-C V3" of 'MISRA Option Manual' for details.

24.3 Addition of support for Compiler

The -K option that supports the following compilers has been added.

-K MS/VC2012

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

-K MS/VC2012MFC

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

-K MS/VC2013

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

-K MS/VC2013MFC

The reserved word of Microsoft Visual C++ 2013(MFC used in Win32bit) is supported.

24.4 Correspondence to the latest OS/MW environment

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

[OS Environment] * Only the floating license version corresponds.

- Windows Server(R) 2016

[Server virtualization software] * Only the floating license version corresponds.

- Microsoft(R) Windows Server(R) 2016 Hyper-V(R)

24.5 Operability/convenience improvement of pgrfake command

The pgrfake command has been improved as follows.

- Environmental variables "PGRFAKE_INI_FILE" is added

It came to be able to specify the arbitrary Command parameter definition file, when the pgrfake command was executed.

- "NOANALYZEKEY" Key is added to the Command parameter definition file

It came to be able to control the execution of PGRelief by specifying the option (optional "-M" of gcc, etc.) that controlled compilation of compiler.

- The phrase method of "OUTPUTDIR" Key specified for the Command parameter definition file is improved

It came to be able to specify "~"(home directory), when the directory that output the result of the pgrfake command was specified.

* It is effective only when operating on "Red Hat Enterprise Linux (R)".

Please refer to "1.3 pgrfake command" of 'Command Manual' for details.

25. Improvements for PGRelief 2017 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2017] to [PGRelief C/C++ 2017 autumn].

In addition, improvements in the previous version are also contained in this version.

25.1 Addition of support for Compiler

The -K option that supports the following compilers has been added.

-K IAR/EWARM7.4

The reserved word of IAR SYSTEMS IAR Embedded Workbench for ARM 7.4 compiler is supported.

-K IAR/EWARM8.1

The reserved word of IAR SYSTEMS IAR Embedded Workbench for ARM 8.1 compiler is supported.

25.2 Correspondence to 64 bits OS

The PGRelief C/C++ (x64 edition) was added. All functions of the PGRelief C/C++ operate as 64-bit native application.

The large-scale source program written in the C/C++ language can be analyzed by GUI and the pgrfake command.

25.3 Improvement for pgr1395

Initialization concerning the message fan-out of pgr1395 was changed in consideration of the case where analytical time becomes long by a large amount of message output.

Before:

Initialization) All the detection parts are output.

Now:

Initialization) Only one place is output by the source file (Each literal type).

Please refer to "2. Detailed Explanation of Message Indications" - pgr1395 of 'Message Indications Manual' for details.

26. Improvements for PGRelief 2018

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2017 autumn] to [PGRelief C/C++ 2018].

In addition, improvements in the previous version are also contained in this version.

26.1 Check for CERT C Coding Operation

The check for CERT C coding operation is available now.

In addition, when checking for CERT C coding operation, the license of CERT Option is needed.

Please refer to 'CERT Option Manual' and "4. CERT Option" of 'Message Indications Manual' for details.

26.2 Addition of Indications

The following 16 indications are added.

ID	Message	Gr	C/C++	Remark
pgr1166	The identifier @1 that begin with an underscore is reserved for ANSI, it is possible to lead to undefined behavior.	b	C/C++	
pgr6000	It may be wrong to pass unlocked @1 to the function @2.	h	C/C++	CERT
pgr6001	You have discarded @1 by function @2 that is not locked or unlocked. Make sure that you have not disposed of objects that you want to use on other threads.	h	C/C++	CERT
pgr6002	The @2 argument @3 of the function @1 is not global or static, so it may not be valid when the thread is executed.	h	C/C++	CERT
pgr6003	The variable @2 whose type is @1 is not a global or static variable, so it may not be valid when the thread is executed.	h	C/C++	CERT
pgr6004	Since the sequence of function @1 is different, a deadlock may occur if function @2 is processed in parallel.(places of comparison: line @3 and line @4, places of indication: line @5 and line @6)	h	C/C++	CERT
pgr6005	There may be a deadlock if variable @2 locked by function @1 has not been unlocked. (locked times : @3, unlocked times : @4)	h	C/C++	CERT
pgr6006	Calling a function @1 that may be blocked by another operation during exclusive control could result in a deadlock.	h	C/C++	CERT
pgr6007	The function @1 should not be used to compare data that contains floating-point type.	b	C/C++	CERT
pgr6008	The thread-unsafe function @1 is used in multithread function.	h	C/C++	CERT
pgr6009	The parameter @1 in main function may not reflect the result of @2.	b	C/C++	CERT
pgr6010	The @2 argument of function @1 does not convert byte order by function @3.	h	C/C++	CERT

pgr6011	If the argument @3 is passed to the @2 parameter of function @1, the thread may be canceled asynchronously. Make sure that a correct argument is passed.	h	C/C++	CERT
pgr6012	The argument is different from the function @2 in the line @1. (condition variable : @3, mutex : @4)	h	C/C++	CERT
pgr6013	If the operand or expression @1 is signed char and does not cast characters to unsigned char before converting to larger integer sizes, then a symbolic extension may appear and the bit pattern may change.	h	C/C++	CERT
pgr6015	Before calling the function @1, the variable @2 is not set to @3. The variable @2 is not initialized in the function @1. Please set to @3 before calling.	h	C/C++	CERT

The meaning of keywords in the Remark Field:

CERT: message output when the license of CERT Option exists.

26.3 Addition of support for Compiler

The -K option that supports the following compilers has been added.

-K MS/VC2015

The reserved word of Microsoft Visual C++ 2015(Win32bit) is supported.

27. Improvements for PGRelief 2018 autumn

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2018] to [PGRelief C/C++ 2018 autumn].

In addition, improvements in the previous version are also contained in this version.

27.1 Improvement of Indications

27.1.1 Improvement for pgr6000, pgr6005

The label of identifier file was added, to control the error message of the function that wraps the lock function and the unlock function.

Please refer to "2. Detailed Explanation of Message Indications" - pgr6000, pgr6005 of 'Message Indications Manual' for details.

27.1.2 Improvement for pgr6001

It was improved, to not the output message when the same mutex has been initialized.

Please refer to "2. Detailed Explanation of Message Indications" - pgr6001 of 'Message Indications Manual' for details.

27.1.3 Improvement for pgr6002

It was improved, to not output message when the substitution immediately before for the pointer type argument is an external variable.

Please refer to "2. Detailed Explanation of Message Indications" - pgr6002 of 'Message Indications Manual' for details.

27.2 Change the GUI's operating environment

The Java of the GUI's operating environment was changed to AdoptOpenJDK 8.

27.3 Addition of CERT C definition files

The CERT C definition files of the following versions that was used to detect the CERT C Coding Standard violations was added.

-C_20180725_pgr2018

28. Improvements for PGRelief 2019

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2018 autumn] to [PGRelief C/C++ 2019].

In addition, improvements in the previous version are also contained in this version.

28.1 Addition of Indications

The following 22 indications are added.

ID	Message	Gr	C/C++	Remark
pgr1242	Expression @2 that uses @1 as operand may contain unnecessary operation.	b	C/C++	
pgr1243	The result of assignment @1 is used in the other expression @2.	g	C/C++	
pgr1250	'\0' is used for pointer type.	b	C/C++	
pgr1251	(void*)0 is used for char type.	b	C/C++	
pgr1252	0 is used for char type.	g	C/C++	
pgr1253	The initialization indicator @2 for array @1 is greater than the array size.	a	C/C++	
pgr1254	The length of the array using the initializer is omitted.	b	C/C++	
pgr1255	The end of the array using the initialization indicator has not been initialized.	g	C/C++	
pgr1256	The bit field @2 which used in multithread function @1 is not in exclusive control.	h	C/C++	
pgr1257	The variable @2 qualified by volatile which used in multithread function @1 is not in exclusive control.	h	C/C++	
pgr1807	The type @1 is not used.	g	C	
pgr1808	The tag @1 is not used.	g	C	
pgr6016	If the function @2 is called in the program exit handler @1, it may cause undefined behavior.(exit handler registration: line @4 of @3)	h	C	CERT
pgr6020	There is a risk of being attacked, if the format specification argument @2 in function @1 uses external information.	h	C/C++	CERT
pgr6021	If you assign the result of function @1 to type @2, you may not be able to correctly determine the end of the file in EOF or WEOF. It would be better to use the same type @3 as the function.	h	C/C++	CERT
pgr6022	You may not be able to correctly determine the state of stream @2 by checking the return value of the character input function @1. It is better to use the function feof or ferror to check the state of the stream.	h	C/C++	CERT
pgr6030	@1 might access an address which exceeds the range of field @2.(field size: @3, copy size: @4)	h	C/C++	CERT
pgr6031	@1 might access an address which exceeds the range of field @2.(field size: @3, copy size: @4)	h	C/C++	CERT

pgr6050	Signal processing is not safe because it calls the asynchronous unsafe function @2 from signal handler @1.(signal registration: line @4 of @3)	h	C	CERT
pgr6051	Signal processing is not safe because it references the asynchronous unsafe variable @2 from signal handler @1.(signal registration: line @4 of @3, variable declaration: line @6 of @5)	b	C	CERT
pgr6052	Signal processing is not safe because it calls the asynchronous unsafe function @5 at line @4 of @3 in the function @2 that is called within signal handler @1.(signal registration: line @7 of @6)	a	C	CERT
pgr6053	Signal processing is not safe because it references the asynchronous unsafe variable @5 at line @4 of @3 in the function @2 that is called within signal handler @1.(signal registration: line @7 of @6, variable declaration: line @9 of @8)	b	C	CERT

The meaning of keywords in the Remark Field:

CERT: message output when the license of CERT Option exists.

28.2 Improvement of Indications

28.2.1 Improvement for pgr2216

Indication of pgr2216 was extended from C++ source file to C/C++ source file.

The label of identifier file was added, to avoid indicate macro functions that are difficult to change to inline or template functions.

Please refer to "2. Detailed Explanation of Message Indications" – pgr2216 of 'Message Indications Manual' for details.

28.3 Check for IPA/SEC-C V3 (ESCR-C V3) Coding Operation

The check for IPA/SEC-C V3 (ESCR-C V3.0) coding operation is available now.

28.4 Addition of CERT C definition files

The CERT C definition files of the following versions that was used to detect the CERT C Coding Standard violations was added.

-C_20190708_pgr2019

28.5 Correspondence to the latest OS/MW environment

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

[OS Environment] * Only the floating license version corresponds.

- Windows Server(R) 2019

[Server virtualization software] * Only the floating license version corresponds.

- Microsoft(R) Windows Server(R) 2019 Hyper-V(R)

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

- Microsoft(R) Office Excel(R) 2016

- Microsoft(R) Office Excel(R) 2019

28.6 Convenience improvement of pgrfake command

You can use the pgrfake command to execute all PGRelief Analysis Commands.

Please refer to "1.3 pgrfake command" and "5.2 Using the pgrfake Command to Execute All PGRelief Analysis Commands (For Users)" of 'Command Manual' for details.

28.7 Change of CI Tool Collaboration Jenkins Plugin provision method

The "CI Tool Collaboration Jenkins Plugin" that was provided in the floating license version installation media has been changed to be available as a download from the product HP. For details on the plug-in, please refer to "CI Tool Collaboration Jenkins Plugin" on the following download site.

<https://www.fujitsu.com/jp/products/software/resources/technical/agilerelief/search/>

* When problems such as EOL (End-of-life) and security vulnerabilities occur in the open source software used in "CI Tool Collaboration Jenkins Plugin", the provision method has been changed so that it can be supported as needed.

29. Improvements for Agile+ Relief V1.1.1

This section demonstrates the changes and improvements made from [PGRelief C/C++ 2019] to [Agile+ Relief V1.1.1].

In addition, improvements in the previous version are also contained in this version.

29.1 Addition of --base and --diff option to pgr5 command

Added --base, --diff options to the pgr5 command. These options are a function to the pgr5 command to suppress messages that were checked in the past or that were checked before the reference point.

Please refer to "1.1.3. Options Instruction" in "Command Manual".

29.2 Added a function to GUI which suppress message by using diff of analyzation

Added a function to GUI which suppress message by using diff of analyzation. Please refer to Agile+ Relief help "Analysis Option Settings" and "Message Viewer".

29.3 Improvement of Indications

29.3.1 Improvement for pgr0431

The message content when parsing of a C++ source file is interrupted has been changed to make it easier to identify the line where the interruption occurred.

Please refer to "2. Detailed Explanation of Message Indications" – pgr0431 of 'Message Indications Manual' for details.

29.3.2 Improvement for pgr0520

If the return type of a function (function with the same name as the identifier registered under the [NULL_RETURN_FUNCTION] label in the identifier file) that may return a pointer containing NULL is unknown, pgr0520 can be detected.

Please refer to "2. Detailed Explanation of Message Indications" – pgr0520 of 'Message Indications Manual' for details.

29.4 Correspondence to the latest OS/MW environment

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

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

- Microsoft(R) Excel(R) for Office 365

- Microsoft(R) Office Excel(R) 2021

[OS Environment] * Only the x64 edition corresponds.

- Windows(R) 11

[OS Environment] * Only the floating license version for 64bit OS corresponds.

- Windows Server(R) 2022

- Red Hat(R) Enterprise Linux(R) 8

[Server virtualization software] * Only the floating license version corresponds.

- Microsoft(R) Windows Server(R) 2022 Hyper-V(R)

- VMware vSphere® 7.0