

Contents

Chapter 1. Changes	2
1.1 License for the compiler	2
1.2 Support for RXv3 Instruction set architecture	2
1.3 Output of CRC operation results to binary files.....	2
1.4 Change of the first load address in a Hex file	2
1.5 Allocating the section to the unused vector area	2
1.6 Enhancement of the feature for detecting illegal indirect function calls [Professional]	3
1.7 Adding the option that can be specified for the library generator	3
1.8 Enhanced optimization	3
1.9 Other improvements.....	5
Chapter 2. Points for Caution	6
2.1 Note on a case of the W0523041 message [C/C++ Compiler].....	6
2.2 Note on using MVTC or POPC instructions [Assembler]	6
2.3 Note on the delete option for linkage [Optimizing linkage editor].....	6
Chapter 3. Restrictions	7
3.1 Restriction on usage of math.h functions (frexp, ldexp, scalbn and remquo) in C++ language (including EC++).....	7
3.2 Restriction of PIC/PID function (pic and pid options)	9
3.3 Eliminated options (for the C/C++ compiler)	9
3.4 Notes on C/C++ source-level debugging (for the C/C++ compiler)	9
3.5 Note on using sections that include the address 0xffffffff (in assembler).....	10
3.6 Note on using -form and -output at the same time (in the linkage editor).....	10
3.7 Note on using function names that begin with _builtin (for the C/C++ compiler).....	10
3.8 Note on using #pragma interrupt with functions for which save_acc is enabled and that have dummy arguments (for the C/C++ compiler)	10
3.9 Restriction of -merge_files	11
3.10 Restriction of -cfi_ignore_module	11
Chapter 4. Standard Libraries Included.....	12
4.1 Library files	12
4.2 Using the library files	13
4.2.1 Using the library files	13
4.2.2 Directory specifying a library file in the optimizing linkage editor	13

Chapter 1. Changes

This section describes changes on CC-RX from V2.08.00 to V3.00.00.

The features indicated as **[Professional]** can only be used if the compiler is registered under the professional license.

1.1 License for the compiler

To use CC-RX V3.00 or later, a license for the V3 version is necessary, which differs from the license for the V2 version that is necessary for CC-RX V2.00 to V2.08. Please be careful.

1.2 Support for RXv3 Instruction set architecture

RXv3 Instruction set architecture is supported.

The object code using RXv3 Instruction is generated by specifying the `-isa=rxv3` option for compiler or assembler.

For the detailed information of RXv3 Instruction set architecture, see Software manual.

1.3 Output of CRC operation results to binary files

The CRC operation results can be output to binary files in V3.00.00. In the version earlier than V3.00.00, the `-crc` option was only valid when `-form={hexadecimal | stype}` is specified. In V3.00.00, it is also valid when `-form=binary` is specified.

1.4 Change of the first load address in a Hex file

The optimizing linkage editor option `-output`, which specifies output files, is modified to accept `loadaddress` as its suboption. When `-output=/load-address` is specified, the first load address in the output file is changed to the value specified with `load-address` when an Intel Hex file or a Motorola S-record file is output. This option is useful when generating files with the PIC facility enabled.

1.5 Allocating the section to the unused vector area

Optimizing linkage editor option `"-split_vect"` is added.

Using the `-split_vect`, vector area section is generated per vector number, and the section for unused vector number is not generated.

This makes the unused vector area available for other purposes.

1.6 Enhancement of the feature for detecting illegal indirect function calls [Professional]

The feature for detecting illegal indirect function calls has been supported since V2.08.00.

In V3.00.00, to enhance this feature, the `-cfi_ignore_module` option is modified to accept library files (*.lib) as parameters.

1.7 Adding the option that can be specified for the library generator

In the previous versions, the library generator had an issue with some options that caused an error and library file was not generated. In V3.00.00, the error is eliminated and the following options are effective for the library generator.

- create_unfilled_area
- stack_protector [Professional]
- stack_protector_all [Professional]
- misra2012 [Professional]

1.8 Enhanced optimization

For V3.00.00, optimization is further enhanced on the point (a) listed and described below.

(a) Reducing the number of load executions

The compiler optimizes and reduces the number of load executed, by utilizing the loaded array elements in the loop statements.

```
// Source code
void func(unsigned char* p, int n) {
    unsigned int i, j;
    unsigned char temp;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - i; j++) {
            if (p[j] != 0 && p[j] > p[j + 1]) {
                temp = p[j];
                p[j] = p[j + 1];
                p[j + 1] = temp;
            }
        }
    }
}
```

```

// V2.08.00 generate code
_func:
    CMP #00H, R2
    PUSH.L R6
    BEQ L19
L11:
    MOV.L #00000000H, R14
    MOV.L R2, R15
L12:
    CMP R2, R14
    BEQ L18
L13:
    MOV.L R15, R5
    MOV.L R1, R3
L14:
    MOVU.B [R3], R4
    CMP #00H, R4
    BEQ L17
L15:
    MOVU.B 01H[R3], R6
    CMP R6, R4
    BLEU L17
L16:
    MOV.B R6, [R3]
    MOV.B R4, 01H[R3]
L17:
    ADD #01H, R3
    SUB #01H, R5
    BNE L14
L18:
    ADD #01H, R14
    SUB #01H, R15
    CMP R2, R14
    BNE L12
L19:
    RTSD #04H, R6-R6
    
```

```

// V3.00.00 generate code
_func:
    CMP #00H, R2
    PUSH.L R6
    BEQ L19
L11:
    MOV.L #00000000H, R14
    MOV.L R2, R15
L12:
    CMP R2, R14
    BEQ L18
L13:
    MOVU.B [R1], R3 ; move to the outside loop
    MOV.L R15, R5
    MOV.L R1, R4
L14:
    MOVU.B 01H[R4], R6
    CMP R6, R3
    BLEU L17
L15:
    TST #0FFH, R3
    BEQ L17
L16:
    MOV.B R6, [R4]
    MOV.L R3, R6
    MOV.B R3, 01H[R4]
L17:
    ADD #01H, R4
    SUB #01H, R5
    MOV.L R6, R3
    BNE L14
L18:
    ADD #01H, R14
    SUB #01H, R15
    CMP R2, R14
    BNE L12
L19:
    RTSD #04H, R6-R6
    
```

1.9 Other improvements

Other major improvements are described below.

(a) Improved prevention of internal errors

Problem with an internal error during building has been rectified.

Chapter 2. Points for Caution

This section describes points for caution regarding CC-RX.

2.1 Note on a case of the W0523041 message [C/C++ Compiler]

When the `int_to_short` option is specified and a file including a C standard header is compiled as C++ or EC++, the compiler may show the W0523041 message. In this case, simply ignore the message because there are no problems.

[NOTE]

In compilation of C++ or EC++, the `int_to_short` option will be invalid.

Data that are shared between C and C++ (EC++) program must be declared as the long or short type rather than as the int type.

2.2 Note on using MVTC or POPC instructions [Assembler]

In the assembly language, the program counter (PC) cannot be specified for MVTC or POPC instructions.

2.3 Note on the delete option for linkage [Optimizing linkage editor]

When a function symbol is removed by the delete option, its following function in the source program is not allowed to have a breakpoint at its function name on the editor in your debugging. If you would like to set a breakpoint via the Label window at the function entrance, set the breakpoint via the Label window or at the program code of the function.

Chapter 3. Restrictions

This chapter describes restrictions on CC-RX.

3.1 Restriction on usage of math.h functions (frexp, ldexp, scalbn and remquo) in C++ language (including EC++)

An object is generated which will be an infinite-loop at execution when the actual argument of some function (frexp, ldexp, scalbn or remquo) of math.h is int-type, at compiling C++ or EC++ program.

Conditions:

This problem occurs when both (1) and (2) are satisfied.

(1) This program is in C++ or the lang=cpp option is effective.

(2) math.h is included and any of the following functions is called.

- (a) frexp(double, long*) with 'int *' type second argument (except when the first argument is float-type and the dbl_size=8 option is effective).
- (b) ldexp(double, long) with 'int *' type second argument (except when the first argument is float-type and the dbl_size=8 option is effective).
- (c) scalbn(double, long) with 'int *' type second argument (except when the first argument is float-type and the dbl_size=8 option is effective).
- (d) remquo(double, double, long*) with 'int *' type third argument (except when the both the first and second arguments are float-type and the dbl_size=8 option is effective).

Examples:

file.cpp:

```
// Example of compiling C++ source that generates an infinity-loop
#include <math.h>
double d1,d2;
int i;
void func(void)
{
    d2 = frexp(d1, &i);
}
```

Command Line:

```
ccrx -cpu=rx600 -output=src file.cpp
```

file.src: Example of the generated assembly program

```
_func:
    ; ... (Omitted)
    ; Calling substitute function of frexp
    BSR __$frexp__tm__2_f__FZ1ZPi_Q2_21_Real_type__tm__4_Z1Z5_Type
    ; ... (Omitted)

__$frexp__tm__2_f__FZ1ZPi_Q2_21_Real_type__tm__4_Z1Z5_Type:
L11:
    BRA L11 ; Calls itself ==> infinity-loop
```

Countermeasures:

Select one of the following ways to avoid the problem.

- (1) Compile the program with the lang=c or lang=c99 option.
- (2) Change int or int * into long or long *.
- (3) Append the following declarations to each function that is being used.

```
/* For the frexp function */
static inline double frexp(double x, int *y)
{ long v = *y; double d = frexp(x,&v); *y = v; return (d); }
/* For the ldexp function */
static inline double ldexp(double x, int y)
{ long v = y; double d = ldexp(x,v); return (d); }
/* For the scalbn function */
static inline double scalbn(double x, int y)
{ long v = y; double d = scalbn(x,v); return (d); }
/* For the remquo function */
static inline double remquo(double x, double y, int *z)
{ long v = *z; double d = remquo(x,y,&v); *z = v; return (d); }
```

Example of (2):

Change in file.cpp:

```
#include <math.h>
double d1,d2;
int i;
void func(void)
{
    long x = i; /* Accept as long type temporary */
    d2 = frexp(d1, &x); /* Call with long type argument */
    i = x; /* Set the result for variable 'i' */
}
```


Example of (3):

Change in file.cpp:

```
#include <math.h>
/* Append declaration */
static inline double frexp(double x, int *y)
{ long v = *y; double d = frexp(x,&v); *y = v; return (d); }
double d1,d2;
int i;
void func(void)
{
    d2 = frexp(d1, &i);
}
```

3.2 Restriction of PIC/PID function (pic and pid options)

When a standard library is created by the library generator (lbgrx) with the pic or pid option specified, the following warning may appear once or more.

```
W0591301:"-pic" option ignored (When the pic option has been specified)
```

```
W0591301:"-pid" option ignored (When the pid option has been specified)
```

Despite the warning, the created standard library has no problems.

3.3 Eliminated options (for the C/C++ compiler)

(a) `-file_inline`, `-file_inline_path`

Specifying these options has no effect and the compiler will output a warning. Instead of **`-file_inline`** or **`-file_inline_path`**, write **`#include`** in the source code. In case of C and C99, `merge_files` can be used instead.

(b) `-enable_register`

This option is simply ignored and does not affect the generated code.

3.4 Notes on C/C++ source-level debugging (for the C/C++ compiler)

(1) Even when **`-debug`** is specified, you may not be able to set a breakpoint or stop stepped execution on lines that

contain a dynamic initialization expression for a global variable (in C++),

are the first lines of functions that begin with a loop statement (e.g. **`do`** or **`while`**) and do not have an **`auto`** variable or of functions for which **`#pragma inline_asm`** has been specified, or

contain the control section and body of a loop statement (e.g. **`for`**, **`while`**, or **`do`**) written as a single line.

(2) The values of members of union type and of dummy variables that are to be passed via registers may be displayed incorrectly (e.g. in the [Watch] window).

3.5 Note on using sections that include the address 0xffffffff (in assembler)

If two or more **.section** directives in the assembly source code contain **.org** directives, the sections have the same name, and the sections overlap at 0xffffffff, the assembler outputs an internal error message (C0554098).

Example)

```
.section SS,ROMDATA
.org 0fffffffh
.byte 1
.byte 2 ; 0xffffffff
.section SS,ROMDATA
.org 0fffffffh
.byte 3; ; 0xffffffff
.end
```

3.6 Note on using **-form** and **-output** at the same time (in the linkage editor)

When **-form=rel** and **-output=<filename>** are specified for the linkage editor (**rlink**) at the same time, the filename extension given as **<filename>** is ignored and replaced with **.rel**.

Example)

```
rlink -form=relocate -output=DefaultBuild\lib_test.lib
```

The filename specified for output, **test.lib**, is changed to **test.rel**.

3.7 Note on using function names that begin with **_builtin** (for the C/C++ compiler)

Declaration of a function with a name that begins with **_builtin** and for which the definition is in **machine.h** in the **include** directory may lead to an internal error. In general, do not use any names that begin with an underscore (**_**) in your source code, since such names are reserved.

3.8 Note on using **#pragma interrupt** with functions for which **save_acc** is enabled and that have dummy arguments (for the C/C++ compiler)

When **#pragma interrupt** is specified for a function and the **save_acc** flag is enabled (including where this is done by using the **-save_acc** compiler option), the compiler may not output code that reflects the correct values of dummy arguments which are passed via R4.

Note: In general, we do not recommend defining arguments for functions with the **#pragma interrupt** specification.

3.9 Restriction of `-merge_files`

Under certain conditions, compilation with `-merge_files` or `-whole_program` specified as the translation unit of code that includes union-type variables will produce error code F0530800 or warning code W0530811.

[Conditions]

If all the following conditions are satisfied, error code F0530800 or warning code W0530811 will be produced.

- (1) `-merge_files` or `-whole_program` is specified.
- (2) A union-type external variable having two or more members has been initialized outside any function, and, other than the member that have been initialized, a member has an alignment and size larger than the other member or members.
- (3) The variable described in (2) above is declared as extern for reference by either of the following.
 - (3-1) Source files other than the definition of external variable described in (2) exists.
 - (3-2) Header files included directly or indirectly by the source files other than the definition of external variable described in (2) exists.

[Workarounds]

Take any of the following steps.

- (1) Specify neither of the options in condition (1).
- (2) Initialize the union-type external variable described in condition (2) within a function.
- (3) Refer to the variables corresponding to condition (2) only in the source file that includes the definition of the external variable.

3.10 Restriction of `-cfi_ignore_module`

When compiling C/C++ source file using `-output=abs`, it isn't possible to specify the intermediately generated object files by `-cfi_ignore_module`.

Please generate an object file using `-output=obj` and indicate a generated object file by `-cfi_ignore_module`.

Chapter 4. Standard Libraries Included

This chapter describes restrictions on standard libraries included in RX Family C/C++ Compiler.

This compiler package includes four library files (*.lib) for the RX600. You can use any of the library files if they correspond to the options that you wish to specify. Using these files shortens the time required for building.

4.1 Library files

Table 4.1 shows the standard library files and compiler options.

Note:

The compiler options you specify should be the same as the microcontroller options defined for each of the library files listed in table 4.1. Otherwise these library files are not usable, so specify your compiler options in the library generator to generate your own library file.

Library File	Purposes	Optimize ^{*2} Options	Microcontroller Options ^{*1 *2}		
			-endian	-cpu -rtti -exception -noexception	Others ^{*3}
rx600lq.lib	For the RX600 Optimization type:Speed Little endian	-speed -goptimize	-endian=little	-cpu=rx600	-round=nearest -denormalize=off -dbl_size=4 -unsigned_char -unsigned_bitfield -bit_order=right -unpack -fint_register=0 -branch=24
rx600ls.lib	For the RX600 Optimization type:Size Little endian	-size -goptimize			
rx600bq.lib	For the RX600 Optimization type: Speed Big endian	-speed -goptimize	-endian=big	-rtti=on -exception	
rx600bs.lib	For the RX600 Optimization type: Size Big endian	-size -goptimize			

Table 4.1 Library Files

*Notes:

*1 For details on microcontroller options, please see the “Microcontroller Options” columns of the “(1) Compile Options” of “section A.1.3 Options”, in the Integrated Development Environment User’s Manual:RX Build.

*2 These option selections are same from the each default of them.

4.2 Using the library files

The library files included in the compiler package must be linked in either of the ways given in sections 4.2.1 and 4.2.2.

4.2.1 Using the library files

When the e² studio has been installed in C:\Renesas\e2_studio, the library files are stored in the following location:

```
C:\Program Files\Renesas\RX\V3_0_0\lib
```

("V3.00.00" indicates the version and revision number of the compiler package.)

4.2.1 Directory specifying a library file in the optimizing linkage editor

Copy the library file(s) included in the package (stored in the location given in section 4.2.1) into a desired directory. Then specify one of the copied library files for the Library option and start the linkage processing.

All trademarks and registered trademarks are the property of their respective owners.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.

No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.

17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338