

Thank you for using the CS+ integrated development environment.

This document describes the restrictions and points for caution. Read this document before using the product.

## Contents

<b>Chapter 1. Target Devices.....</b>	<b>2</b>
<b>Chapter 2. User's Manuals .....</b>	<b>3</b>
<b>Chapter 3. Keywords When Uninstalling the Product .....</b>	<b>4</b>
<b>Chapter 4. Changes.....</b>	<b>5</b>
4.1 License for the compiler.....	5
4.2 Instruction set of the G4MH core [g4mh].....	5
4.3 Extended floating-point system registers [g4mh] .....	5
4.4 Register bank facility [g4mh] .....	6
4.5 Interrupt specifications in interrupt and exception handlers [g4mh].....	6
4.6 Intrinsic functions for the instruction set of the G4MH core [g4mh] .....	8
4.7 Intrinsic functions for debug instructions .....	8
4.8 C99 standard library functions .....	9
4.9 Inline expansion of standard library functions .....	9
4.10 Generation of floating-point operation codes that are efficient in execution .....	10
4.11 Generation of codes for floating-point reciprocal operations.....	11
4.12 Enhancement of the feature for detecting illegal indirect function calls [Professional edition]	12
4.13 Output of CRC operation results to binary files .....	12
4.14 Change of the first load address in a Hex file .....	12
4.15 Rectified points for caution.....	12
4.16 Other changes and improvements .....	13

---

## Chapter 1. Target Devices

The target devices supported by the CC-RH compiler are listed on the Website.

Please see the URL below.

CS+ Product Page:

<http://www.renesas.com/cs+>

## Chapter 2. User's Manuals

Please read the following user's manuals along with this document.

Manual Name	Document Number
CC-RH Compiler User's Manual	R20UT3516EJ0105
CS+ Integrated Development Environment User's Manual: CC-RH Build Tool Operation	R20UT3283EJ0106

---

## Chapter 3. Keywords When Uninstalling the Product

There are two ways to uninstall this product.

- Use the integrated uninstaller from Renesas (uninstalls all CS+ components)
- Use the Windows uninstaller (only uninstalls this product)

To use the Windows uninstaller, select "CS+ CC-RH V2.00.00" from "Programs and Features" of the control panel.

## Chapter 4. Changes

This chapter describes changes to the CC-RH compiler from V1.07.00 to V2.00.00.

Note that the features and changes that are only available to users holding a registered license for the Professional edition are indicated as **[Professional edition]**.

The features and changes applied only to the MCUs based on the G4MH core are indicated as **[g4mh]**.

### 4.1 License for the compiler

To use CC-RH V2.xx.xx, a license for the V2 version is necessary, which differs from the license for the V1 version that is necessary for CC-RH V1.xx.xx.

### 4.2 Instruction set of the G4MH core **[g4mh]**

**g4mh** has been added as a new argument for the `-Xcpu` option (compiler option and assembler option), which specifies the CPU core.

Specifying the `-Xcpu=g4mh` option generates a code that uses the instruction set of the G4MH at compilation. This option also enables the assembly of the G4MH instructions.

### 4.3 Extended floating-point system registers **[g4mh]**

A new option `-Xfxu` has been added to enable or disable the generation of codes for saving and restoring the extended floating-point system registers FXSR and FXXP in interrupt functions. This option should be specified in the form of `-Xfxu={on/off}`.

Note that V2.00.00 does not generate the extended floating-point calculation instruction (FXU instruction) during compilation of a C source file. Accordingly, this option should be specified when using the FXU instruction through the inline assembler directive in an interrupt function.

Argument	Description
<i>on</i>	Generates codes for saving and restoring the extended floating-point system registers FXSR and FXXP in interrupt functions.
<i>off</i>	Does not generate codes for saving and restoring the extended floating-point system registers FXSR and FXXP in interrupt functions.

## 4.4 Register bank facility [g4mh]

A new option **-Xresbank\_mode** has been added to enable or disable the generation of codes for saving and restoring the r20 to r29 registers and r31 register in the interrupt functions that use the RESBANK instruction. This option should be specified in the form of **-Xresbank\_mode=num**.

<i>num</i>	Description
0	Generates codes for saving and restoring the r20 to r29 registers and r31 register in the interrupt functions that use the RESBANK instruction.
1	Does not generate codes for saving and restoring the r20 to r29 registers and r31 register in the interrupt functions that use the RESBANK instruction.

Note that this option does not set up the save mode in the MCU. This only selects whether to generate codes for saving and restoring registers in accordance with the actual save mode specified in the MCU.

When the save mode in the MCU is set to 0 (RBCR0.MD = 0), specify **-Xresbank\_mode=0**.

When it is set to 1 (RBCR0.MD = 1), specify **-Xresbank\_mode=1**.

## 4.5 Interrupt specifications in interrupt and exception handlers [g4mh]

Interrupt and exception handlers can be specified through the following #pragma interrupt directive.

```
#pragma interrupt [(function-name[(interrupt-specification
[, interrupt-specification]...)] [,...] [ ]]
```

In V2.00.00, the parameters that can be used in the interrupt specifications have been changed as follows.

- **priority =**

**FPE** and **FXE** have been added as arguments that can be specified for the parameter.

When **FPE** or **FXE** is specified, codes for saving and restoring the registers are generated in accordance with the respective exception sources.

- **channel =**

The upper limit of the value that can be specified for the parameter has been extended from 511 to **2047**.

In V2.00.00, the following interrupt specifications have been added.

- **resbank**

This enables the use of the RESBANK instruction in EI-level interrupt functions.

When the RESBANK instruction is used, no codes are generated to save or restore the EIPC and EIPSW registers even in a multiplex interrupt function.

The error E0523005 will be output when this interrupt specification is used in the following cases.

- ✓ **-Xcpu=g4mh** is not specified.

- ✓ The `priority=` parameter is specified and an argument other than `EIINT_PRIORITY` is specified for the parameter.

The warning W0523116 will be output when this interrupt specification is used in the following cases. No codes are generated to save or restore respective registers in these cases, but when `resbank` is specified, the register values that have been modified in an interrupt function are restored to the previous values through the `RESBANK` instruction when execution returns from the interrupt function.

- ✓ `fpu=false` is specified together with `resbank` in the `#pragma` interrupt directive.
- ✓ The `-Xreg_mode` option is set to a value other than 32.
- ✓ The `-Xreserve_r2` option is specified.
- ✓ `-Xep=fix` is specified.

- **`fxu =`**

This enables or disables the generation of codes for saving and restoring the `FXSR` and `FXXP` registers in interrupt functions. `true`, `false`, or `auto` can be specified as an argument.

Argument	Description
<code>true</code>	Generates codes for saving and restoring the <code>FXSR</code> and <code>FXXP</code> registers in interrupt functions.
<code>false</code>	Does not generate codes for saving and restoring the <code>FXSR</code> and <code>FXXP</code> registers in interrupt functions.
<code>auto</code>	Generation of codes for saving and restoring the <code>FXSR</code> and <code>FXXP</code> registers depends on the setting of the <code>-Xfxu</code> option.

- **`param =`**

This specifies exception source registers whose values are stored in parameters at the beginning of an interrupt function.

The following six exception source registers can be specified. Up to four exception sources can be obtained through parameters in an interrupt function.

- `eiic*`, `feic*`, `fpsr*`, `fxsr`, `fxxc`, `fxxp`

[Example]

```
volatile unsigned long v;
#pragma interrupt handler (param=(eiic,fxsr,fxxc))
void handler (unsigned long a, unsigned long b, unsigned long c) {
    v = a; // The value of EIIC is stored in parameter a.
    v = b; // The value of FXSR is stored in parameter b.
    v = c; // The value of FXXC is stored in parameter c.
}
```

Note: \* These exception source registers can be specified even when `-Xcpu=g4mh` is not specified.

## 4.6 Intrinsic functions for the instruction set of the G4MH core [g4mh]

The following intrinsic functions have been added, which are available when the `-Xcpu=g4mh` option is specified.

Assembler Instruction	Format of Intrinsic Function
<code>clip.b</code>	<code>long __clipb(long a);</code>
<code>clip.bu</code>	<code>unsigned long __clipbu(unsigned long a);</code>
<code>clip.h</code>	<code>long __cliph(long a);</code>
<code>clip.hu</code>	<code>unsigned long __cliphu(unsigned long a);</code>
<code>ldl.bu</code>	<code>long __ldlbu(unsigned char* a);</code>
<code>ldl.hu</code>	<code>long __ldlhu(unsigned short* a);</code>
<code>stc.b</code>	<code>long __stcb(unsigned char* a, unsigned char b);</code>
<code>stc.h</code>	<code>long __stch(unsigned short* a, unsigned short b);</code>

Note that when the `-Xcpu=g4mh` option is specified, the intrinsic function `__set_il_rh()`, which controls the interrupt level, cannot be used. If attempted, the error F0523073 will be output.

Directly access and manipulate the interrupt control registers when the `-Xcpu=g4mh` option is specified.

## 4.7 Intrinsic functions for debug instructions

The following intrinsic functions have been added.

Assembler Instruction	Format of Intrinsic Function
<code>dbcp</code>	<code>void __dbcp(void);</code>
<code>dbpush</code>	<code>void __dbpush(long regID1, long regID2);</code>
<code>dbtag</code>	<code>void __dbtag(long a);</code>

## 4.8 C99 standard library functions

The following C99 standard library functions have been supported.

- `fmax()`, `fmaxf()`
- `fmin()`, `fminf()`
- `copysign()`, `copysignf()`

## 4.9 Inline expansion of standard library functions

A new option\* **-library** has been added to select whether to use function calls or to expand functions inline in the generation of codes for calling the following standard library functions. This option should be specified in the form of `-library={function|intrinsic}`.

- `abs()`, `labs()`, `llabs()`
- `fabs()`, `fabsf()`
- `sqrt()`, `sqrtf()`
- `fmax()`, `fmaxf()`
- `fmin()`, `fminf()`
- `copysign()`, `copysignf()`

Argument	Description
<i>function</i>	Always uses function calls in code generation.
<i>intrinsic</i>	Expands functions inline to improve the efficiency of code execution. Functions may not be expanded inline under some calling conditions.

Note: \* This is a compiler option and differs from the linker option `-library`.

## 4.10 Generation of floating-point operation codes that are efficient in execution

A new option **-relaxed\_math** has been added to generate floating-point operation codes that are efficient in execution. When this option is specified, codes are generated with the following rules in the C-language and IEEE754 standards not applied strictly. Accordingly, the operation results may differ from those obtained by strictly applying the rules in the C-language and IEEE754 standards.

- Operation precision
- Exception occurrence
- Not-a-numbers
- Sign of 0.0

[Example 1] The operation precision is not ensured.

```
float func1(float a) {
    return (a * 3.0f) * 0.3f;
}
```

### When the option is not specified

```
_func1:
    movhi 0x00004040, r0, r2
    mulf.s r2, r6, r2
    mov 0x3E99999A, r5
    mulf.s r5, r2, r10
    jmp [r31]
```

### When the option is specified

```
_fun1:
    mov 0x3F666667, r2
    mulf.s r2, r6, r10
    jmp [r31]
```

When the option is not specified: After calculation of (a \* 3.0f), the result is multiplied with 0.3f.

When the option is specified: "3.0f \* 0.3f = 0.899...f" is calculated during compilation and the entire calculation is converted to a single multiplication of "a \* 0.899...f".

[Example 2] Input of not-a-numbers is not assumed.

```
float func2(float a, float b) {
    return (a > b) ? a : b;
}
```

**When the option is not specified**

```
_func2:
    cmpf.s 0x00000006, r7, r6
    cmovf.s 0, r6, r7, r10
    jmp [r31]
```

**When the option is specified**

```
_func2:
    maxf.s r6, r7, r10
    jmp [r31]
```

When the option is not specified: If a or b is a not-a-number, the result of the `cmpf.s` instruction is "false" and b is returned.

When the option is specified: If a or b is a not-a-number, the `maxf.s` instruction returns the value that is not a not-a-number.

[Example 3] The sign of 0.0 is ignored.

```
float func3(float a, float b) {
    return -(a - b);
}
```

**When the option is not specified**

```
_func3:
    subf.s r7, r6, r2
    negf.s r2, r10
    jmp [r31]
```

**When the option is specified**

```
_func3:
    subf.s r6, r7, r10
    jmp [r31]
```

When the option is not specified: If a and b are  $-0.0f$ ,  $-0.0f$  is returned.

When the option is specified: If a and b are  $-0.0f$ ,  $+0.0f$  is returned because  $-(a - b)$  is converted to  $(b - a)$ .

## 4.11 Generation of codes for floating-point reciprocal operations

A new option `-use_recipf` has been added to generate the `recipf` instruction (`recipf.d` or `recipf.s`) for floating-point reciprocal operations.

In V2.00.00 or earlier versions, the `recipf` instruction was generated by default to improve the efficiency of execution. However, the `recipf` instruction always causes an inexact operation exception in the FPU. To prevent this situation, the `divf` instruction is generated by default in V2.00.00. To generate the `recipf` instruction, specify the `-use_recipf` option.

## 4.12 Enhancement of the feature for detecting illegal indirect function calls [Professional edition]

The feature for detecting indirect function calls to illegal addresses was newly supported in V1.07.00. To enhance this feature, the `-cfi_ignore_module` option has been modified to accept library files (\*.lib) as parameters.

## 4.13 Output of CRC operation results to binary files

The CRC operation results can be output to binary files in V2.00.00.

In V2.00.00 or earlier versions, the `-crc` option was only valid when `-form={hexadecimal | stype}` is specified. In V2.00.00, it is also valid when `-form=binary` is specified.

## 4.14 Change of the first load address in a Hex file

The linker option `-output=suboption`, which specifies output files, has been modified to accept `load-address` as a 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 effective when generating files with the PIC facility enabled.

## 4.15 Rectified points for caution

The following four points for caution no longer apply. For details, refer to Tool News.

- Static declaration of a structure, an array, or a union that has an initializer (No. 19)
- Assembly-language code using reserved symbol (No. 20)
- Section where the initializers of auto variables are allocated when the `-Xmulti_level` option is specified (No. 21)
- Compiler option “`-store_reg`” (No. 22)

## 4.16 Other changes and improvements

Other major changes and improvements are described below.

(a) Assembly list file (\*.prn)

The information under "Command Line Parameter" at the end of the \*.prn files may differ from those output in previous versions, but this does not affect the generated codes.

(b) Setting of invalid options

When an invalid option is specified, an error message or a warning message may be output.

[Example 1] The `-Xpreprocess` option is specified while the `-P` option is not specified.

V2.00.00 or earlier versions:

No error or warning is output and this option is ignored.

V2.00.00 or later versions:

The warning "W0511151: The "-Xpreprocess" option is ignored when the "-P" option is not specified." is output and this option is ignored.

[Example 2] The `-V` option, which cannot have any argument, is specified together with the `-VV` option.

V2.00.00 or earlier versions:

The error "E0511109: The "-V" option can not have an argument." is output.

V2.00.00 or later versions:

The error "E0511108: The "-VV" option is not recognized." is output.

(c) Enhancement of optimization

The execution speed has been increased mainly by improving the analysis of loop statements. Other optimization processes have also been enhanced.

(d) Correction of internal errors

Internal errors sometimes occurred in the build process in previous versions. These errors have been corrected.

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 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