CubeSuite+ RX Compiler CC-RX V2.00.00

Release Notes

R20UT2524EJ0100
Rev.1.00
Mar 25, 2013

Contents

Chapter 1. Target Devices

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

Please see this URL.

CubeSuite+ Product Page:

http://www.renesas.com/cubesuite+

Chapter 2. Key Points for Selecting Uninstallation Method

There are two ways to uninstall this product.

- Use the integrated uninstaller (uninstalls CubeSuite+)
- Use separate uninstaller (uninstalls this product only)

To use the separate uninstaller, select the following from the Control Panel:

- Add/Remove Programs (Windows XP)
- Programs and Features (Windows Vista, Windows 7)

Then select "CubeSuite+ CC-RX V2.00.00".

Chapter 3. Changes

This section describes changes for using CC-RX.

3.1 New optimization engine for enhance optimization [C/C++ Compiler]

3.2 Stronger optimization over greater ranges during the process of compilation [C/C++ Compiler]

(a) **-merge_files** option to allow the compiler to compile multiple source files and output the results in a single object file (except in the case of C++ and EC++ files).

(b) **-whole_program** option to make the compiler perform optimization on the assumption that the input files form a complete program.

(c) **-ip_optimize** option to apply global optimization (i.e. optimization spanning functions).

3.3 Support for **big5** (traditional characters) and **gb2312** (simplified characters) Chinese character sets [C/C++ Compiler]

3.4 Changes to assembly source code output [C/C++ Compiler]

Assembly source code output when **-output=source** or **-listfile** is specified differs to that output by version 1.xx of the compiler in the following ways.

(a) Tabs are inserted before and after each RX instruction and assembler directive.

(b) **.LINE** directives are not output unless **-debug** has been specified.

(c) Information on some labels used by the compiler in its internal processing may be output as comments.

(d) When **-show=source** is specified, **#include** statements are displayed without expansion.

(e) When **-show=source** is specified, lines that follow a **#line** directive may be shown differently from those in the actual source code.

3.5 Types of global variables in the debugging environment [C/C++ Compiler]

When two or more declarations include the definition of a global variable and the types do not match in all cases, the defined types are used in the display of the variable in the debugging environment.

3.6 Corrections to faults

(a) The following problems with the linkage editor have been corrected.

L1320 was output even if there were no duplicate symbols between files.

L1320 was not output even if there were duplicate symbols between files.

(b) When the **-rom** option was used to allocate a section of size zero that had a boundary alignment value of two or more in the location counter for the RAM, symbols defined in ROM sections were not correctly allocated to RAM sections.

3.7 Destination of output messages (all commands)

The destination of output messages has been changed from standard output to standard error.

Chapter 4. Cautions

This section describes cautions for using CC-RX.

4.1 Note on a Case of the W052304 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 W052304 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.

4.2 Note on using MVTC or POPC instructions [Assembler]

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

4.3 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 at the function entrance, set the breakpoint via the Label window or at the prologue code of the function.

Chapter 5. Restrictions

This chapter describes the restrictions of the CC-RX V.2.00.00.

5.1 Restriction of PID function (nouse_pid_register Option)

When using PID function, if the nouse_pid_register option is effective for the all program files of master program, the following errors will occur.

(1) A program of setting address into PID register

(2) Standard library (Setting it to the library generator 'lbgrx')

[For restriction of (1):]

First, take out a function of setting PID register and put it into an independent C or assembler file.

Next, compile or assemble this file without the nouse_pid_register option.

Finally, link it and other master files together.

[For restriction of (2):]

Please cope by following (a) or (b) method.

(a)To include a standard library in application not a master

* Using jump-table is not necessary. (see also chapter 8.4.2(2).
* Generate standard library with setting the pid option for the library generator 'lbgrx'.

(b)To include a standard library in a master

(i) Generate standard library without setting the pid option for the library generator 'lbgrx'.

(ii) For following examples, change each JMP R14 statements of the all entries in the jump table, and use the changed one.

Example: Changing _printf entry

Before:

```
_printf:
   MOV.L #0ffff90cfH,R14 ; Address 0ffff90cfH is an example
   JMP R14
```

After:

```
_printf:
   MOV.L  #0ffff90cfH,R14
   PUSH.L R13 ; The case of PID register is R13
   JSR    R14
   POP    R13 ; The case of PID register is R13
   RTS
```

5.2 Restriction on usage the some math.h functions of frexp, ldex, scalbn and remquo on C++ language (including EC++)

Compiling C++ and EC++ program that uses a certain math.h function (frexp, ldexp, scalbn or remquo) with an int-type argument generates an infinity-loop object.

Conditions:

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

(1) This program 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 (excpet 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):

Changed of 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):

Changed of 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);
}
```

5.3 Using the Standard Library (for Customers Who Have Upgraded the Compiler from V1.02.00 to V1.02.01)

If you were using the standard library with V1.02.00 and then upgraded the compiler to V1.02.01 or newer, change environment variable TMP_RX to a different directory from that used with V1.02.00. This is necessary because the library generator, lbgrx, for V1.02.00 Release 00 and later versions stores the intermediate results of creating a library in the directory indicated by TMP_RX and reuses these on the next occasion a library is created.

If you do not change the directory, the codes created by the older compiler remain in the standard libraries generated by lbgrx.

Follow the procedure below to use the standard library in the High-performance Embedded Workshop.

[Procedure to be Taken to Use the Standard Library in the CubeSuite+]

Go through steps (1) to (3) once after upgrading the compiler to V1.02.01.

(1) Open the command prompt (the following steps require operations at the command prompt).

(2) Execute dir %TEMP%*.pgl at the command prompt and check that one or more files with a name that is a combination of numbers and letters and with the extension .pgl appears.

Result of executing dir %TEMP%*.pgl (example):

```
del %TEMP%\80000400801000002 25a40409694ab0200000000.pgl
```

(3) Use the del command to delete each of the .pgl files that appeared in step (2). Using the del command to delete one file (example):

[Remark]

If there are no .pgl files other than those that appeared in step (2), you can delete all of the .pgl files at once.

Deleting all .pgl files by using a command (example):

```
del %TEMP%\*.pgl
```

5.4 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.

```
C1301 (W) "-pic" option ignored (When the pic option has been specified)
C1301 (W) "-pid" option ignored (When the pid option has been specified)
```

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

5.5 Building Projects Converted from V1.00.xx to V1.01.00 or V1.02.00 -- (Warning W0561120 and Error W0563100 on Section L)

If you convert a High-performance Embedded Workshop project created with compiler package V.1.00 Release 00 or an early version to make it compatible with V1.02.00, the following warning and error message may appear.

```
W0561120 Section address is not assigned to "L"
W0563100 Section address overflow out of range : "L"
```

In such a case, select (a) or (b) given in section 5, “9.4.1 Compatibility with V.1.00 / (2) Adding Section L (section Option and Start Option)” in this document.

Remark:

If you have used the High-performance Embedded Workshop included in this compiler package to convert a project created with compiler package V.1.00 Release 02 or an early version (RX Toolchain 1.0.0.0 to 1.0.0.2) to make it compatible with V1.02.00 or later, this problem does not occur because (b) is automatically done by the CubeSuite+.

5.6 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

(b) -enable_register
This option is simply ignored and does not affect the generated code.

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

(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).

5.8 Note on using two or more files with the same name other than the filename extension (in the C/C++ compiler)

An internal error occurs if two or more files specified for the ccrx command in conjunction with **-output=abs** have the same name except for the filename extension.

Example) ccrx -output=abs file.c file.src

E0511200: Internal error(505)

5.9 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 0fffffffeh
.byte 1
.byte 2 ; 0xffffffff
.section SS,ROMDATA
.org 0ffffffffh
.byte 3; ; 0xffffffff
.end
```

5.10 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**.

5.11 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.

5.12 Note on using functions for which **instalign** or #**pragma instalign8** has been specified (for the C/C++ compiler)

If the definition of function A is followed by that of function B, and function B is called from within the final line of function A, the assembler may output an error message when either of the following conditions is satisfied.

- The source code is compiled with **-instalign=8**.

- **#pragma instalign8** is specified for function B.

5.13 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.

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be 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, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

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: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, 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 906, 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 Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2