

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. 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 categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “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.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. 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.
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.

**= = = Be sure to read this note = = =**

## **M32R Family C/C++ compiler package (CC32R)**

### **V.5.00 Release 00**

## **Release Notes 1st Edition**

**Renesas Solutions Corp.**

**December 28, 2005**

**Welcome to M32R Family C/C++ compiler package (abbreviated as CC32R) V.5.00 Release 00. This document contains supplementary descriptions to the electronic User's Manual. Please read this release note while you refer to a corresponding item in electronic User's Manual.**

**Renesas Technology Corp. and Renesas Solutions Corp. reserve the right to change the contents of this release note without notice for improvements on characteristics, etc.**

Active X, Microsoft, MS-DOS, Visual Basic, Visual C++, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

HP-UX is a registered trademark of Hewlett-Packard Company.

Sun, Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries, and are used under license.

UNIX is a registered trademark of The Open Group in the United States and other countries.

IBM and AT are registered trademarks of International Business Machines Corporation.

HP9000 is a product name of Hewlett-Packard Company.

SPARC and SPARCstation are registered trademarks of SPARC International, Inc.

Intel and Pentium are registered trademarks of Intel Corporation.

i386, i486, and MMX are trademarks of Intel Corporation.

Adobe and Acrobat are registered trademarks of Adobe Systems Incorporated.

Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation in the U.S. and other countries.

All other brand and product names are trademarks, registered trademarks or service marks of their respective holders.

=====  
COPYRIGHT(C) 2005 **Renesas Technology Corp.** and **Renesas Solutions Corp.**

# 1. Changes from V.4.30 Release 00

---

The changes from the previous version are as follows.

## Corresponded to the C++ language (C/C++ compiler):

Supports a programming that used the C++ language.

Furthermore, the several C++ source code operations (compiling, linking, and library-generating) are different from the case of conventional C source codes.

See chapter 4 for details.

## Upgraded integrated development environment as HEW4:

Upgraded the HEW (High-performance Embedded Workshop), integrated development environment, to 4.00.01 from 3.01.04.

## Changed the installer:

Changed the installing program. See the chapter 3 for details.

- Installs CC32R and HEW in the same action.
- Bundles the Renesas Auto-Updater.
- Changed the start menu into "Renesas" from "RENESAS-TOOLS".
- Changed the install directory of default.

## Changed as an object module and assembly source file would be made in same time (C/C++ compiler):

Added new option "-cs" to the compiler.

This option means compiling and generating an object module and assembly source file in same time.

## Improved the -SEC option as for use many times (linker, C/C++ compiler):

Changed the compiler cc32R and the linker lnk32R so that the -SEC option was able to use 2 or more times.

### a) Conventional style -SEC option

```
% cc32R -o file.abs -SEC @D=1000,B,P=8000,C,D md1.mo md2.mo md3.mo
```

### b) New style -SEC option

```
% cc32R -o file.abs -SEC @D=1000,B -SEC P=8000 -SEC C,D md1.mo md2.mo md3.mo
```

## The three problems as follows have been fixed:

- On setting the address value of an object containing a structure, array or union using the #pragma ADDRESS directive, CC32R may generate incorrect code for this reference (C/C++ compiler):  
Published to RENESAS TOOL NEWS on February 16, 2005: RSO-M3T-CC32R-050216D
- On making a function call using a pointer pointing to a function whose type has been converted by a cast operator, CC32R may describe an error (C/C++ compiler):  
Published to RENESAS TOOL NEWS on December 1, 2004: RSO-M3T-CC32R-041201D
- On initializing a two-dimensional array of type char using an initializer, CC32R may generate incorrect code for this reference (C/C++ compiler):  
Published to RENESAS TOOL NEWS on September 16, 2004: RSO-M3T-CC32R-040916D

## 2. Precautions to be observed when installing

---

Please pay attention to the following point, in starting installation.

### About the license ID

You need to input a license ID in the intermediate step of installation. Before you start installing CC32R, check your license ID.

Yet, a fee is necessary to the changing from previous version to V.5.00 Release 00. Therefore, this version of CC32R cannot be installed by license ID of V.4.30 Release 00 or earlier.

### Restrictions and Precautions

The Restrictions or Precautions sometimes are being published to this document. Before using CC32R please read attentively these contents.

These contents include several precautions that were found after the manual was made.

### Necessary memory and HDD capacity

In order that CC32R operates comfortably, it requires at least 64 Mbytes of memory and a hard disk drive having 500 Mbytes or more of space.

### Operating environment

The CC32R does not run under Windows 3.1 and Windows NT3.5x or earlier.

### Precautions about the file name and directory name

The file name and directory name of the source program must follow the precautions described below:

- Directory and file names that contain multi-byte character (ex. Japanese-Kanji) cannot be used.
- Only one period (.) can be used in a file name.
- Network path names cannot be used. Assign the path to a drive name.
- Shortcuts cannot be used.
- Directory and file names that contain a space character cannot be used.
- The ". . ." symbol cannot be used as a means of specifying two or more directories.
- A file name in length of 128 characters or more including path specification cannot be used.

### Precautions about virus check programs

If this software is started while a virus check program is resident in memory, it may not operate properly. In such a case, remove the virus check program from memory before you start the software.

### When introducing the SQMLint (MISRA C Rule Checker).

Please be sure to install the SQMLint after the CC32R.

If the CC32R is installed after the SQMLint, that cannot use the function of the SQMLint.

For details, refer to the manual or release note that are attached to the SQMLint.

### Settings of environment variables

The environment variables listed below must always be set. If one of these variables remains unset or an invalid path is specified, an error may occur when running the software. (These variables do not need to be set when running the CC32R from TM.)

M32RLIB  
M32RBIN  
M32RINC  
M32RTMP

## Entering user registration

To be eligible for upgrade information, technical support, and other services, you must be registered as a user with Renesas. Unless you are a registered user, the said services cannot be received.

When you've installed the CC32R, the following file is created.

```
C:\Renesas\CC32R\V500R00\support\cc32r\regist.txt
```

(It is in the case the directory was chosen in installation supposed as C:\Renesas\CC32R\V500R00.)

Copy all contents of the `regist.txt` file and paste them into a file, then send it to the electronic mail address given below.

```
regist_tool@renesas.com
```

If you are not using an electronic mail, output the contents of the `regist.txt` file to a printer and send the printout to Renesas by facsimile. See the License ID Certificate included with your product to find the address to be sent.

(If you use facsimile, a some number of days may be required until completion of user registration. Please use e-mail, as much as possible.)

For information on our policy concerning the protection of personal information, please refer to the Renesas Technology Homepage.

URL: <http://www.renesas.com/fmwk.jsp?cnt=privacy.htm&fp=/privacy/&site=i>

The information we receive via the User Registration Form aids us greatly in our customer support activities. We provide Renesas Technology and related companies, distributors, etc., with essential user information (electronically or on paper) that will further help them provide customer support.

If you do not wish to have your user information provided to other related companies, please contact us to let us know. Note, however, this will limit our ability to provide complete product support.

## How to get the latest information of CC32R

Renesas has an internet home page in the URL shown below. The latest information on Renesas Development Environmen are published here.

```
http://www.renesas.com/en/tools
```

## 3. Installing CC32R

### 3.1 How to install CC32R

(1) Execute the installer by the method of items as follows (A or B).

A. In the case of installing from the included CD:

Execute each installation program as shown in Table 1, in accordance with OS and language that you are using.

(The drive letter of CD-ROM (Q:) differs each in PC. Please confirm yours. )

**Table 1 CC32R file name of installer** (The drive letter of CD-ROM is supposed with "Q:".)

Supported OS	Languages	Installer names
Windows 98 / SE Windows Me Windows NT 4.0	English	Q:\CC32R\W95E\SETUP.EXE
Windows 2000 Windows XP	Japanese	Q:\CC32R\W95J\SETUP.EXE

B. In the case of utilizing online version up]:

Download installer files and install by the method that is written to the tool homepage.

(2) Follow the messages displayed on the screen as you install CC32R.

[About entering user information]

The data you input in the intermediate of installation is necessary to create a file for user registration.

\* Please confirm "Entering user registration" of chapter 2, for information on our policy concerning the protection of personal information.

[About selecting contents of installation]

It is not possible selective installation. (That means only compiler or only hew, for example.)

If any components of them are not necessity, please do uninstallation each of them after installation in accordance with the method that was described to "3.2 How to uninstall".

[About target directory of installation]

CC32R (cross tool group: C/C++ compiler, etc.) and the integrated development environment HEW are able to choose separately the target directories. Edit them if necessary.

\* If the integrated development environment HEW is already installed the target directory of HEW is not able to change.

**Table 2 The default target directory of installation**

Objects	The default target directory of initial installation (Able to change)	Already installed	
		Directory	Can be changed ?
CC32R C/C++ compiler, assembler, etc.	C:\Renesas\CC32R\V500R00	same as the left	Yes
The integrated development environment HEW	C:\Program Files Renesas\HEW	Directory of installed the HEW	No

[About the AutoUpdater]

The AutoUpdater will start and station into PC automatically.

The AutoUpdater is an utility that watch the Renesas HomePage periodically and detects the renewal of the installed development tools.

[Constitution of start menu]

After installation, the folders and shortcuts that showed them below will be registered to the [start]->[Programs]->[Renesas].

High-performance Embedded Workshop  
M3T-CC32R V.x.xx Release xx (Numbers or versions are displayed to "x").  
User Registration  
Renesas AutoUpdate  
Renesas Tools HomePage

### 3.2 How to uninstall

The program that became unnecessary can be eliminated from PC by the following method. Furthermore, please uninstall the integrated development environment and CC32R separately.

1. If the integrated development environment is need to uninstall, confirm whether or not the AutoUpdate icon is on the Windows task tray.  
If its icon is exist, choose by the right click, and select "(E)nd" to finish the AutoUpdate.
2. Open the "Add and remove programs" (or "Add and remove applications" [\*\*]) from the Windows "Control panel"
3. Select one item of the table 3 from "Add and remove programs" list and click the "Modify and remove". (This button may be not the "Modify and remove" but the "Remove" or "Add and remove"[\*\*].)

[\*\*] These differ by Windows.

**Table 3 The index names of uninstall items**

Objects	Index names	Remarks
CC32R C/C++ compiler, assembler, etc.	Renesas M32R Family C/C++ Compiler Package M3T-CC32R V.x.xx Release xx (xx-xx-xxxx xx:xx:xx)	Versions, numbers, dates and times are displayed to "x".
The integrated development environment HEW	High-performance Embedded Workshop	Will be uninstalled with the other HEW in a mass.

4. Uninstallation will start. Then wait until "Completed" is displayed and push "OK".

### 3.3 Setting PC environment

If you use on DOS prompt, please set environment variables like Table 4.

The environment variables marked by "Auto" in Table 4, you do not need to be set because the installer automatically rewrites AUTOEXEC.BAT.

**Table 4 Example of setting PC environment variables**

(The directory name of the installation is supposed with "C:\Renesas\CC32R\V500R00" .)

Environment names	Example of settings
M32RBIN	Auto (SET M32RBIN=C:\Renesas\CC32R\V500R00\bin32R)
M32RLIB	Auto (SET M32RLIB=C:\Renesas\CC32R\V500R00\lib32R)
M32RINC	Auto (SET M32RINC=C:\Renesas\CC32R\V500R00\inc32R)
M32RTMP	Auto (SET M32RTMP=C:\Renesas\CC32R\V500R00\TMP)
M32RKIN	SET M32RKIN=euc
M32RKOUT	SET M32RKOUT=euc
Command path	Auto (PATH %M32RBIN%; %PATH%)

#### [NOTES]

- In this setup example, products are installed by "C:\Renesas\CC32R\V500R00" of the installer. If you wish to install products in a different directory, change the setup contents to the one you want.
- If you use from HEW (High-performance Embedded Workshop) you do not need to be set these environment variables.



### 3.4 How to browse the electronic manual

To see these electronic manuals, use a PDF file displaying program, like a "Adobe Acrobat Reader". So install it in your computer as necessary.

- About the Acrobat Reader

Download "Adobe Acrobat Reader" from following URL of Adobe Systems Incorporated. The latest Acrobat Reader can be downloaded from this home page. Please refer to the following URL about the environment can be use and the latest information for Acrobat Reader.

<http://www.adobe.com/>

When using CC32R, there are following two methods for displaying an electronic manual:

1. The electronic manuals are registered in the start menu when installing by default.

Choose the necessary electronic manual file from menu

[Start] → [Programs] → [RENESAS-TOOLS] → [CC32R V.x.xx Release x]

(x expresses version number etc.)

2. Open by double-clicking an electronic manual file

The electronic manuals are installed in the below

C:\Renesas\CC32R\V500R00\manual by installation directory of default

(C:\Renesas\CC32R\V500R00). Double-click these files to open an electronic manual by Acrobat Reader. It is possible to read manuals, that you open by double-clicking these files (extension .pdf).

The following Table 5 lists the electronic manual files.

Table 5 Electronic manual file names

Languages	Manual names	PDF file names
English	M32R Family C/C++ Compiler Package C/C++ Compiler User's Manual	cc32rue.pdf
	M32R Family C/C++ Compiler Package Assembler User's Manual	as32rue.pdf
	MAP Viewer User's Manual	mapue.pdf
Japanese	M32R Family C/C++ Compiler Package C/C++ Compiler User's Manual	cc32ruj.pdf
	M32R Family C/C++ Compiler Package Assembler User's Manual	as32ruj.pdf
	MAP Viewer User's Manual	mapuj.pdf

[NOTE] (About User's Manual of MAP Viewer)

MAP Viewer can be used even if it combines with CC32R. (In this manual, MAP Viewer is used with NC30WA.)

## 4. Restrictions on Usage

There are restrictions of the CC32R.

For other precautions of only this version, see the 'Precautions on using' of the next chapter.

### ■ About the CPU Selection and the start-up file in the HEW (Integrated Environment)

In the case of making a new workspace of the integrated environment HEW (High-performance Embedded Workshop), the "CPU Series" item has no effect to the start-up file. Add the control register setting program needed for the generated start-up file. These settings are written to the user's manuals of each CPU.

Furthermore, the "CPU Series" and "CPU Types" items at making a new workspace will be used to the initial setting of the compiling options.

### ■ About changing CTOR, VTBL and COMMON sections (C/C++ Compiler)

The CTOR, VTBL and COMMON sections that was made on a C++ compiling cannot be changed even if one of #pragma SECTION and even -R option was used.

That is because these sections need distinction especially by the standard library and the linker, also if these section names were changed an obstruction on their processing will appear. Therefore, those names can not be changed.

### ■ Restriction of #pragma INTERRUPT (INTF) on the C++ language (C/C++ Compiler)

The #pragma INTERRUPT (and the #pragma INTF) designates to an interrupt handling function can be written on a C++ language. But, the functions that were designated shall be of the C language.

Therefore, on the C++ language, the function that is designated with the #pragma INTERRUPT shall be declared by extern "C" description.

### ■ On the case of the very long identifier definition on the C++ language

In the C++ language the result of the 241 or more characters length identifier definition differs from the C language.

Table 6 The compilation results of identifier that has 241 characters or more

Identifiers	In the C language	In the C++ language
global objects	Using only the first 240 characters, and discarding from 241st the characters of the back.	Using only the first 240 characters, and discarding from 241st the characters of the back.
Other identifiers (function, class, structure, union, member, argument)	Using only the first 240 characters, and discarding from 241st the characters of the back.	The compiler recognizes and distinguishes from 241st the characters of the back, but these names are symbolized for the recognizing. Therefore, for example, a function name is not displayed by the original name in the linking error, etc.

■ On the case of describing outside range value to the `#line` directive in the C++ language

In the C++ language the result of describing outside range value to the `#line` directive differs from the C language.

*Caution:*

Even if the language which is C or C++ in CC32R, the 32,768 or more value cannot be described to the `#line` directive. Actually, even without defending this regulation, the generated M32R machine codes will work. However, be careful as for that an compiling error may occur, and the source file may not be displayed normally when the generated object is used for debugging.

1 ~ 32,767:

It is processed normally.

32,768 ~ 4,294,967,295:

An internal error may happen as follows.

```
internal error: BAD Location  
internal error: unreachable logic
```

4,294,967,296 or more:

It is processed as an error as follows

```
error: invalid line number
```

## 5. Precautions on using of V.5.00 Release 00

---

There are precautions on usage about this version.

- On declaring the template variable as an array, an error may occur at the linking (C/C++ Compiler)

When a template variable as an array is declared from a class template, this object's instance is not defined.

Therefore, this array object will be undefined symbol and an error may occur at the linking.

### Conditions:

This problem occurs if the following conditions are satisfied:

- (1) The `-strict_standard` option is not selected.
- (2) An array object is defined from a class template.
- (3) The object in (2) is external defined.  
(That means it is global variable with no static declaration.)
- (4) The object in (2) has no initializer.
- (5) The object in (2) is referenced or assigned in a function.

### Examples:

[Source file]

[sample.c]

```
-----  
template <class T> class X {  
public:  
    T data;  
};  
  
X<int> A[10];          /* Conditions(2)(3)(4) */  
  
void  
func(void)  
{  
    A[3].data = 0;    /* Condition(5) */  
}  
-----
```

### Solutions:

This problem will be circumvented in any of the following ways:

- (1) Make the `-strict_standard` option effective when compiling this program.
- (2) Add an initializer at the array object.

[Avoidance example for sample.c]

```
-----  
template <class T> class X {  
public:  
    T data;  
};  
  
X<int> A[10] = {0};    /* Adding the initializer "={0}" */  
  
void  
func(void)  
{  
    A[3].data = 0;  
}  
-----
```

- On dividing the minimum value of the long type by minus one, CC32R will show an error. (C/C++ Compiler)

When the program of dividing the minimum value of the long type by minus one is compiled by CC32R with the level-4 optimization, the opt.exe will cause an application error, and the compiling will abort.

**Conditions:**

This problem occurs if the following conditions are satisfied:

- (1) Any of the optimizing options -O7, -O6, -O5, and -O4, or -Ospace only or -Otime only is selected.
- (2) Using one of the following dividing operations with expression A and B:
  - (a) A / B
  - (b) A % B
- (3) The expression A in (2) satisfies one of the following conditions:
  - (a) Constant -2,147,483,648
  - (b) Constant LONG\_MIN
  - (c) It is a variable of long type, and dynamically initialized with the value which is one of (3)(a) or (b) above inside this function, furthermore this initializing is kept until the place (2).
- (4) The expression B in (2) satisfies one of the following conditions:
  - (a) Constant -1
  - (b) It is a variable of long type, and dynamically initialized with the value which is (4)(a) above inside this function, furthermore this initializing is kept until the place (2).

**Examples:**

[Source file]

[sample.c]

```
-----  
#include <limits.h>  
long a, b, c;  
void func(void)  
{  
    a = LONG_MIN;          /* Condition (3)(c) */  
    b = -1;                /* Condition (4)(b) */  
  
    /* Both a and b are not changed in this place. */  
  
    c = a % b;            /* Condition (2)(b), (3)(c) */  
}
```

**Solutions:**

This problem will be circumvented in any of the following ways:

- (1) Suppress optimization in the -O4 level.  
Optimization covering the -O4 option will be performed when optimizing option -O4, -O5, -O6, -O7, -Otime only or -Ospace only has been selected. If you want to use -Otime or -Ospace, select any of these options, -O0, -O1, -O2, and -O3, at the same time.
- (2) Change this expression as it becomes safe  
If all specified condition is satisfied, this expression result is always zero. Therefore, to avoid this problem, please replace this expression with 0.

- On masking the members in the same union with a constant of the range of 0x80000000-0xffffffff, internal error will occur. (C/C++ Compiler)

When the program of masking two or more different type members in the same union with the same constants of the range of 0x80000000-0xffffffff is compiled by CC32R with C++ mode, the following internal error will occur, and the compiling will abort.

```
internal error: illegal IL, Illegal type conversion.
```

**Conditions:**

This problem occurs if the following conditions are satisfied:

- (1) Any of the optimizing options -O7, -O6, -O5, and -O4, or -Ospace only or -Otime only is selected.
- (2) The program is compiled with C++ mode.
- (3) In the same union, there are same size two members of signed and unsigned integers.
- (4) Two expressions that are bitwise-and of each member values of (3) and same constant values are described.
- (5) The constant values of (4) are range of 0x80000000-0xffffffff.
- (6) Before the computation of (4), this union is used for the initialization of another same type union.

**Examples:**

[Source file]

[sample.c]

```
-----
union utag {
    signed int m1;          /* Condition (3) */
    unsigned int m2;       /* Condition (3) */
} xu;
int func(void)
{
    const union utag xu = u; /* Condition (6) */
                          /* Condition (4),(5) */
    if ((0xffffffff & u.m1) && (0xffffffff & u.m2)) {
        return 0;
    }
    return 1;
}
-----
```

**Solutions:**

This problem will be circumvented in any of the following ways:

- (1) Suppress optimization in the -O4 level.  
Optimization covering the -O4 option will be performed when optimizing option -O4, -O5, -O6, -O7, -Otime only or -Ospace only has been selected. If you want to use -Otime or -Ospace, select any of these options, -O0, -O1, -O2, and -O3, at the same time.
- (2) Delete the expression of initialization.  
On the example, if the expression of initialization of condition (6) is not necessary, please delete it.
- (3) Use the initialized union.  
Please change the expression that corresponds in the condition (4) to the initialized union of condition (6).

[Example:]

```
-----
if ((0xffffffff & u.m1) && (0xffffffff & u.m2)) {
    ==> if ((0xffffffff & xu.m1) && (0xffffffff & xu.m2)) {
-----
```

## 6. Software table

Table 7 shows the look of the directory and file that are made after installation.

Table 7 Table of the directory and file after installation

Directory names	File names	Notes
bin32R	cc32R.exe as32R.exe lnk32R.exe lib32R.exe lmc32R.exe map32R.exe strip32R.exe	Compile driver (V.3.00.02.013) Assemble driver (V.2.05.00.000) Linker (V.1.15.00.000) Librarian (V.1.06.00.000) Load module converter (V.1.14.00.000) Map generator (V.1.23.00.000) Debug information discarding utility (V.1.02.00.000)
lib32R	cpre.exe cpfirt32R.exe cprt.exe postpar.exe opt.exe cg32R.exe genstk.exe a032R.exe a132R.exe alis32R.exe parafilt.exe plink32R.exe conv32R.exe cmerge.exe libg32R.exe m32RcR.lib m32RcRM.lib m32RcRL.lib m32RcR.stk m32RcRM.stk m32RcRL.stk	Preprocessor (V.2.08.00.000) C++ Parser (V.1.02.02.000) Parser (V.2.28.00.000) Post parser (V.1.04.00.000) Optimizer (V.1.29.00.000) Code generator (V.4.05.00.000) Stack file generator (V.1.00.00.000) Macro processor (V.1.02.00.000) Assembler (V.4.05.01.000) List processor (V.1.02.00.000) Parallel processor (V.1.01.00.000) Pre-linker (V.1.00.00.000) SYSROF to ELF converter (V.1.02.01.001) C source merge processor (V.1.03.00.000) Library generator (V.1.00.00.000) C Library (Small model) C Library (Medium model) C Library (Large model) Stack utilize display file for C library (for m32RcR.lib) Stack utilize display file for C library (for m32RcRM.lib) Stack utilize display file for C library (for m32RcRL.lib)
inc32R	assert.h, ctype.h, errno.h, float.h, limits.h, locale.h, long64.h, math.h, mathf.h, setjmp.h, signal.h, stdarg.h, stddef.h, stdio.h, stdlib.h, string.h, time.h, cstddef, cstdio, cstdlib, exception, new, new_ecpp.h, new_std.h, stdexcept, typeinfo	C Library headers
inc32R\sys	assert.h, ctype.h, errno.h, float.h, limits.h, locale.h, math.h, mathf.h, setjmp.h, signal.h, stdarg.h, stddef.h, stdio.h, stdlib.h, string.h, time.h	System definition headers (*1)
inc32R\com	ANSI_erro.h, def.h, SBPP	
UnSpt32R (*2)	abslist.exe stk32R.exe license.txt license.sj abslist.txt stk.txt abslist.sj stk.sj	Absolute listing utility (V.1.02.00.000) Stack size calculation utility (V.1.02.01.000) Development support utility guide (English) Development support utility guide (Japanese) Utility manual of abslist (English) Utility manual of stk32R (English) Utility manual of abslist (Japanese) Utility manual of stk32R (Japanese)
lib32R	lbg_cc32r.dep	
lib32R\src	~.cpp ~.c ~.ms	Files for C library generating
lib32R\pack	~.pack	

support	mtoolspt.htm	Link-page for "Renesas Technology Software and Tools"
support\cc32r	support.txt regist.txt	Customer and Product information record files
smp32R	start.ms	Startup, Low-level functions example
bin	mapviewer.exe map_inspect.dll mapviewer.hlp mapviewer.cnt	Map Viewer (V.3.00.00) DLL file for Map Viewer Help file for Map Viewer Help setting file for Map Viewer
manual	CC32Rue.pdf AS32Rue.pdf CC32Ruj.pdf AS32Ruj.pdf mapuj.pdf	User's manual < C Compiler > [English] User's manual < C Compiler > [English] User's manual < C Compiler > [Japanese] User's manual < C Compiler > [Japanese] Map Viewer Manual [English or Japanese (will be selected by the installer)]
	~.msf	Manual attribute files
hew	~.det ~.dll ~.tbp	HEW setting files
lib32R	tmcpp.exe ~.opt cc32r.mkt	TM setting files (*3)
tmp	<< Empty >>	Temporary directory for CC32R

## NOTES

- (\*1) The System definition headers are unable to be deleted and modified, because CC32R refers to them in compiling.  
If these files are deleted or changed CC32R does not run normally.
- (\*2) The program in the UnSpt32R directory differ from the constitution things of other CC32R regarding the handling of the license and support. Please confirm the document file "license.txt" in this directory.
- (\*3) There are some files for TM in lib32R directory. They are attached for maintaining compatibility with the V.4.30 or older version compilers.  
Furthermore, in this version, the TM is not included in product.