

---

[Notes]  
C Compiler Package for RL78 Family  
(CCRL#027)

---

R20TS0671EJ0100  
Rev.1.00  
Mar. 16, 2021

## Overview

When using the C compiler package for RL78 family CC-RL, note the following point.

1. Using the pack function (CCRL#027)

Note: The number following the note is an identification number for the note.

## 1. Using the pack function (CCRL#027)

### 1.1 Applicable Products

CC-RL V1.02.00 to V1.10.00

### 1.2 Details

When the pack function is used, data may be written to a wrong area of a structure-type or union-type variable that has a double type or long long type member.

### 1.3 Conditions

Data may be written to a wrong area if all the conditions from (1) to (5) are met:

- (1) Either `-pack` or `#pragma pack(*)` is used.
- (2) The structure-type or union-type variable has either a double type<sup>(\*)</sup> or long long type member.
- (3) The member in (2) is not volatile-qualified.
- (4) A return value of a function that has the same type as (2) is directly assigned to the member in (2).
- (5) The start address of the member in (2) is an odd number.

\*1: `#pragma pack` function is available in CC-RL V1.05 or later.

\*2: This applies when `-dbl_size=8` is used.

## 1.4 Example

Below is an example of the error. The parts corresponding to the error conditions are shown in red.

[Example]

ccrl -cpu=S3 -Onothing -pack tp.c (1)

```
/* tp.c */
struct T1 {
    signed char m1;
    long long m2;    // (2) (3)
};
struct T1 gx1;
long long f1(void) {
    return 1;
}
void main(void) {
    gx1.m2 = f1();  // (4)
}
```

In this example, data is written to a wrong area because the return value of the function f1() (long long type) is directly assigned to the member m2 of the structure-type variable gx1 (an odd address of long long type).

## 1.5 Workaround

You can avoid this problem by one of the following methods:

- (a) Avoid using -pack or #pragma pack.
- (b) Qualify the member with volatile.
- (c) Assign the return value of the function to an automatic variable, and then assign the automatic variable to the member.
- (d) Allocate the address of the variable so that the start address of the member is an even number.

## 1.6 Permanent Measure

The problem will be fixed in CC-RL V1.11.00. The release date has not yet been decided.

**Revision History**

Rev.	Date	Description	
		Page	Summary
1.00	Mar.16.21	-	First edition issued

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.

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included.

The URL in the Tool News also may be subject to change or become invalid without prior notice.

**Corporate Headquarters**

TOYOSU FORESIA, 3- 2- 24 Toyosu,  
Koto-ku, Tokyo 135- 0061, Japan

[www.renesas.com](http://www.renesas.com)

**Trademarks**

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

**Contact information**

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

[www.renesas.com/contact/](http://www.renesas.com/contact/)