

[Notes]

R20TS0698EJ0100

Rev.1.00

C/C++ Compiler Package for RX Family (No.59)

Jun. 01, 2021

Overview

When using the C/C++ Compiler Package for RX Family, note the following point.

1. Initializing a union in a C++ source code (No.59)

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

1. Initializing a union in a C++ source code (No.59)

1.1 Applicable Products

CC-RX V2.00.00 to V3.03.00

1.2 Description

The memory area assigned to a union-type variable, union member of a structure-type variable, or element of a union-type array becomes smaller than the size that is supposed to be allocated. As a result, a range outside the area assigned to these variables might be read or written.

1.3 Conditions

This problem may arise if all of the conditions from (1) to (4) are met.

- (1) The compilation option or source file satisfies either of the following conditions:
 - (1-1) Either -lang=cpp or -lang=ecpp is specified as a compilation option.
 - (1-2) A source file with cpp, cc, or cp extension is compiled without using a -lang compilation option.
- (2) There is a global or static variable of one of the following types:
 - (2-1) Union type
 - (2-2) Structure type with a union member
 - (2-3) Array type with a union element
- (3) The size of the first member of a union type in condition (2) is greater than 1 byte.
- (4) A union-type variable, union member, or element covered by condition (2) is initialized by an empty initializer list ("{}").

1.4 Example

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

```
ccrx -isa=rxv2 -lang=cpp tp.c (1-1)
```

```
/* tp.c */
typedef union {
    unsigned long a[2];    // (3)
    unsigned char b[8];
} U;
U x = {};                // (2) (4)
```

When this example is compiled, the following assembly source is generated.

```
.glob _x
.section D,ROMDATA,ALIGN=4
_x:
    .byte 00H,00H,00H,00H
.END
```

While an 8-byte area must be allocated, an area is allocated only for 4 bytes.

1.5 Workarounds

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

- (1) Avoid initializing variables covered by condition (2).
- (2) Change the first member of the union type in condition (2) to a one-byte member or a bit field of 1 to 8 bits.
- (3) Avoid initializing a union-type variable, union member, or element covered by condition (2) by an empty initializer list.
(For example, by initializing it by {0}.)

1.6 Permanent Measure

The problem will be fixed in CC-RX V3.04.00. The release date has not yet been decided.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.01.21	-	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 URLs 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

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/.