RENESAS Tool News

RENESAS TOOL NEWS on April 16, 2015: 150416/tn2

Note on Using the CS+ Integrated Development Environment

When using the CS+ IDE, take note of the problem described in this note regarding the following point.

- Statements in source code which form a deeply-nested block

---

1. Products Concerned
   Products from the following list for which the version number of the CS+ common program is 3.00.00 to 3.01.00.
   - RX Family C/C++ Compiler Package (with IDE)
   - RL78 Family C Compiler Package (with IDE)
   - RH850 Family C Compiler Package (with IDE)
   - CS+ evaluation edition

   To check the version number of the product you have, refer to the following URL.
   https://www.renesas.com/cubesuite+_ver

2. Description
   CS+ might be terminated forcibly when a program is downloaded to a debugging tool or when an editor panel is opened after downloading a program.

3. Conditions
   Forced termination may occur when the source code for a project includes code that meets any of the following conditions.
   (a) { } blocks nested to a depth of 128 or more within a function.
   (b) 64 or more consecutive "else if" conditions are in sequence.
   (c) The total of double the number of consecutive "else if" conditions and the depth of the nesting of {} blocks at some point in the sequence of consecutive "else if" conditions is 128 or more.

With conditions (b) and (c) above, the problem only arises when the C99 option is designated and the product is the RX family C/C++ compiler package (with IDE).

4. Workaround

To avoid this problem, do any of the following.

- Workaround by suppressing the display of source code

  Change the debugging information option for the source file which satisfies a condition for the problem occurring to "No output".
  Note that source-level debugging will not proceed for the corresponded source file in this case.
  Refer to the description in the "help" system of CS+ regarding how to change the option.
  [Build] -> [Functions]
  -> [Set Build Options Separately]
  -> [Set build options at the file level]

- Workaround by making nesting in the code more shallow

  To avoid the condition for the problem occurring, change the source code for the function.
  The following is an example of the modification of code which satisfies condition (b) in 3.

  Before modification

```
----------------------------
   void func()
   {
    if (i == 0) {;}
       else if (i == 1) {;}
       else if (i == 2) {;}
       else if (i == 3) {;}
            :
   }
----------------------------
```

  After modification

```
----------------------------
   void func()
   {
    if (i == 0) {;}
    else if (i == 1) {;}
    else sub(i);
   }
```

```
void sub(int i)
{
  if (i == 2) {;}
  else if (i == 3) {;}
       :
}
----------------------------
```

- In the case of conditions (b) and (c), you can avoid the problem by not designating the C99 option with the RX family C/C++ compiler package (with IDE).

5. Schedule for Fixing the Problem
   This problem will be fixed in a later version of the product.

[Disclaimer]
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.

© 2010-2016 Renesas Electronics Corporation. All rights reserved.