RL78 Family

EEPROM Emulation Library Pack02 Package Ver.2.00 Release Note

R20UT2645EJ0102
Rev.1.02
Dec 16, 2016

Thank you for using the RL78 Family EEPROM Emulation Library Pack02 Package Ver.2.00.

This document contains precautionary and other notes regarding use of the EEPROM Emulation Library Pack02 Package Ver.2.00. Please read this document before using the library.

Contents

Chapter 1 Target Product

The following shows the target product for this release note.

Product Name	Ver.	Installer Name	Ver.
RL78 Family EEPROM Emulation Library Pack02 for the CA78K0R Compiler	V1.01	RENESAS_RL78_EEL-FDL_T02_PACK02_2V00.exe	V2.00
RL78 Family EEPROM Emulation Library Pack02 for the CC-RL Compiler	V1.01		

Chapter 2 User’s Manual

The following user’s manual covers this version of the library:

Compiler Used	Title of User’s Manual	Document Number
CA78K0R Compiler for RL78 Family	RL78 Family EEPROM Emulation Library Pack02 User's Manual Japanese Release	R01US0068EJ0101
CC-RL Compiler for RL78 Family	RL78 Family EEPROM Emulation Library Pack02 User's Manual Japanese Release	R01US0068EJ0101
	RL78 Family EEPROM Emulation Library Pack02 Differences between the CA78K0R compiler version and the CC-RL compiler version	R20UT3486EJ0100

Chapter 3 Revisions

The following shows the items upgraded in the new version.

No.	Package Ver.	Target Library	Contents
1	V2.00	RL78 Family EEPROM Emulation Library Pack02 for CA78K0R	There are no changes in the library from the previous Zip version (JP_R_EEL_RL78_P02_V1.01_B_E).
		RL78 Family EEPROM Emulation Library Pack02 for CC-RL	There are no changes in the library from the previous Zip version (JP_R_EEL_RL78_P02_V1.01_CCRL_B_E).
		RL78 Family EEPROM Emulation Library Pack02 for CA78K0R User's Manual	There are no changes in the descriptions of the User's Manual (R01US0068EJ0101).
		RL78 Family EEPROM Emulation Library Pack02 for CC-RL User's Manual	There are no changes in the descriptions of the User's Manual (R01US0068EJ0101). There are no changes in the descriptions of differences between the CA78K0R compiler version and the CC-RL compiler version (R20UT3486EJ0100).

Chapter 4 Supported Tools

Use the following tool version when using tools in combination with this library:

Target Library	Tool Used	Version
EEPROM Emulation Library Pack02 for CA78K0R	Integrated development environment CubeSuite+	V1.00.00 or later
	Integrated development environment CS+	V3.00.00 or later
EEPROM Emulation Library Pack02 for CC-RL	Integrated development environment CS+	V3.01.00 or later

Chapter 5 Installation

This chapter describes how to install and uninstall the EEPROM Emulation Library Pack02 package V2.00.

5.1 Installation

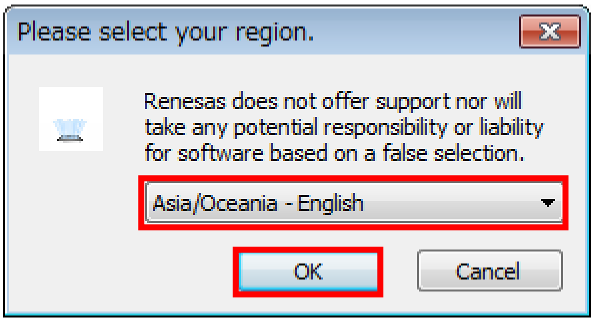
Install the EEPROM Emulation Library Pack02 by using the following procedure:

(1) Start Windows.

(2) Decompress the file that contains the EEPROM Emulation Library Pack02 package and run the installer.

(3) Select “Asia/Oceania – English” from the drop-down list.

(4) Click on the “OK” button to proceed installation according to the instructions of the installer.



Please select your region.

Renesas does not offer support nor will take any potential responsibility or liability for software based on a false selection.

Asia/Oceania - English

5.2 Uninstallation

Uninstall the EEPROM Emulation Library Pack02 by using the following procedure:

(1) Start Windows.

(2) Delete the folder that contains the EEPROM Emulation Library Pack02 files.

5.3 File Organization

The file organization after this library is installed is shown below.

```
Installation folder
├─ r20ut2465ejxxxx_rl78.pdf          : Release Note (this document)
├─ support.txt                       : Support information file for EEL
└─ CA78K0R_110 or CCRL_100
      ├─ doc
      │   ├─ r01us0068ejxxxx_rl78.pdf     : User’s Manual
      │   └─ r20ut3486ejxxxx_rl78.pdf     : Difference document (CC-RL version only)
      ├─ lib
      │   ├─ eel.lib                      :EEPROM emulation library (EEL)
      │   ├─ fdl.lib                      :Data flash library (FDL)
      │   ├─ eel.h                        :EEL header file for C program
      │   ├─ eel.inc                      :EEL header file for assembler
      │   ├─ eel_types.h                  :EEL header file that specifies definitions for C program
      │   ├─ eel_types.inc                :EEL header file that specifies definitions for assembler
      │   ├─ fdl.h                        :FDL header file for C program
      │   ├─ fdl.inc                      :FDL header file for assembler
      │   └─ fdl_types.h                  :FDL header file that specifies definitions for C program
      │
      └─ Sample
          ├─ asm
          │   ├─ eel_descriptor.inc       :EEL descriptor header file
          │   ├─ eel_descriptor.asm       :EEL descriptor source file
          │   ├─ eel_sample_linker_file.dr :EEL descriptor source file  (CA78K0R version only)
          │   ├─ fdl_descriptor.inc       :FDL descriptor header file
          │   └─ fdl_descriptor.asm       :FDL descriptor source file
          └─ C
              ├─ eel_descriptor.h         :EEL descriptor header file
              ├─ eel_descriptor.c         :EEL descriptor source file
              ├─ eel_user_types.h         :EEL user-defined header file
              ├─ eel_sample_linker_file.dr :EEL sample directive file  (CA78K0R version only)
              ├─ fdl_descriptor.h         :FDL descriptor header file
              ├─ fdl_descriptor.c         :FDL descriptor source file
              ├─ r_eel_sample_c.c         :EEL sample program file (CA78K0R version only)
              └─ r_eel_sample_c.dr        :EEL sample Link directive file (CA78K0R version only)
```

Notes: 1. x indicates the omitted numerals in version or revision numbers.

2. If you wish to use the sample program for CA78K0R, include both the program file (*.c) and the link directive file (*.dr).

For the installer for CC-RL, the link directive file (*.dr) [setting file for link information] is not included.

Set link information while the CC-RL compiler is in use on the Link Options tabbed page in the CS+ window.

Chapter 6 How To Build a Program

This chapter describes how to build a program using the EEPROM Emulation Library Pack02.

6.1 Software to be Used

Below are the system requirements for building programs using the EEPROM Emulation Library Pack02.

- For CA78K0R compiler : Integrated development environment CubeSuite+ V1.00.00 or later or integrated development environment CS+ V3.00.00 or later
- For CC-RL compiler : integrated development environment CS+ V3.01.00 or later

6.2 Building Using CS+ (Formerly CubeSuite+)

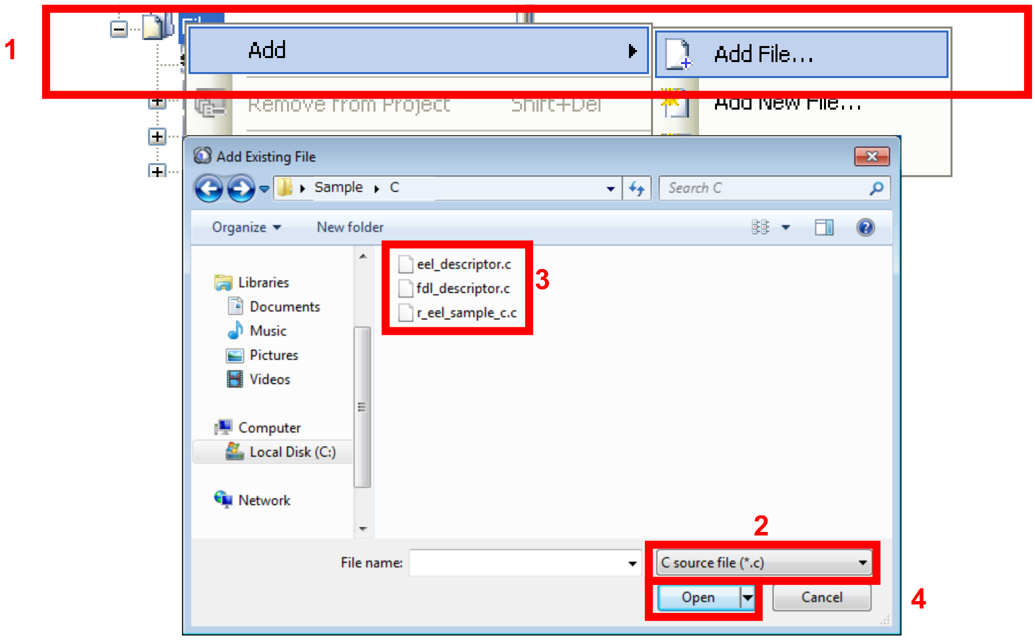
This section describes how to include the EEPROM Emulation Library Pack02 in a user-created program and build the user program by using CS+.

6.2.1 Building a C Program

(1) Creating a project and specifying the source files

Create a project by using CS+. In the Project Tree window displayed on the left, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-1).
Click the Files of type drop-down list, select C source file (*.c), and then register the user-created program file (r_eel_sample_c.c for the sample file of source code) and the descriptor files for the EEPROM emulation library and data flash library (eel_descriptor.c and fdl_descriptor.c) as the source files.

Figure 6-1. Specifying the Source Files



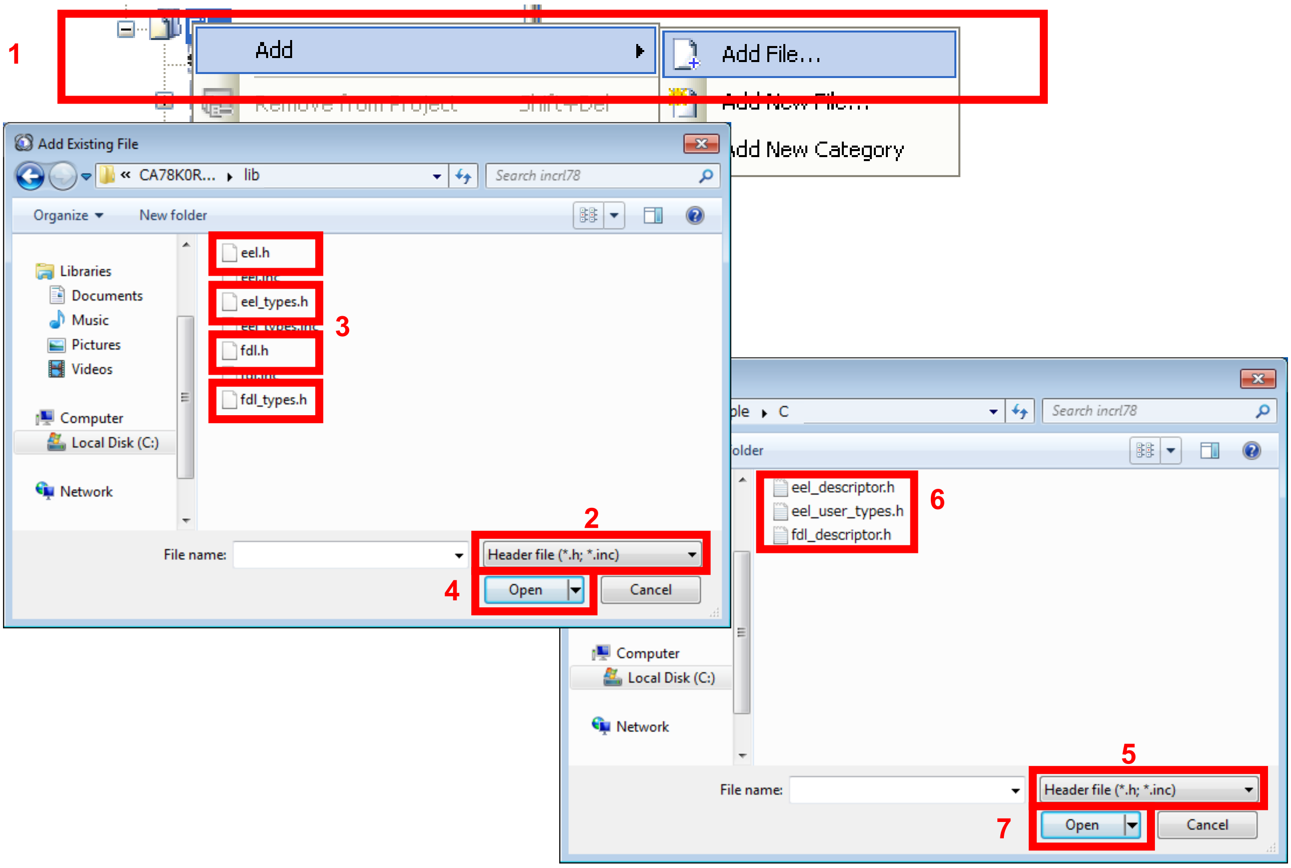
(2) Specifying the include file

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File.

The Add Existing File dialog box is displayed (as shown in Figure 6-2).

Click the Files of type drop-down list, select Header file (*.h;*.inc), and then register the header files and descriptor header files for the EEPROM emulation library and data flash library (eel.h, eel_types.h, fdl.h, fdl_types.h, eel_descriptor.h, fdl_descriptor.h, and eel_user_types.h).

Figure 6-2. Specifying the Include Files

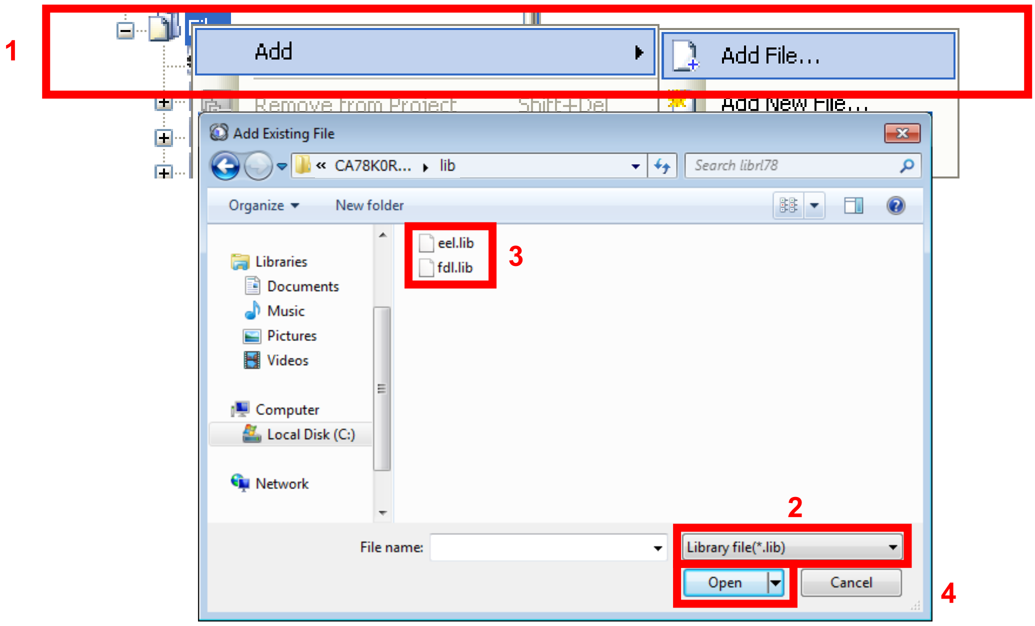


(3) Specifying the library file

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-3).

Click the Files of type drop-down list, select Library file (*.lib), and then register the EEPROM emulation library and data flash library files (eel.lib and fdl.lib).

Figure 6-3. Specifying the Library Files

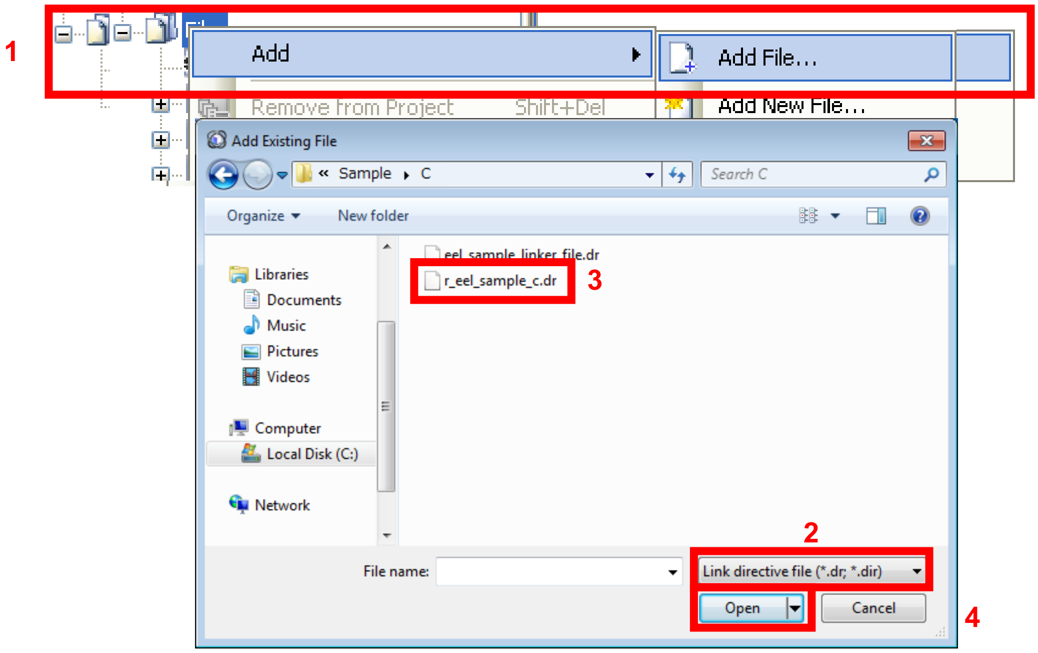


(4) Specifying the link directive file

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-4).

Click the Files of type drop-down list, select Link Directive File (*.dr;*.dir), and then register the link directive file that has the same name as the user-created program (r_eel_sample_c.dr for the sample file of source code [Note]).

Figure 6-4. Specifying the Link Directive File



Note: The sample directive file that comes with the library may require editing or modification before use.

(5) Building

On the CS+ Build menu, click Build Project to build the project.

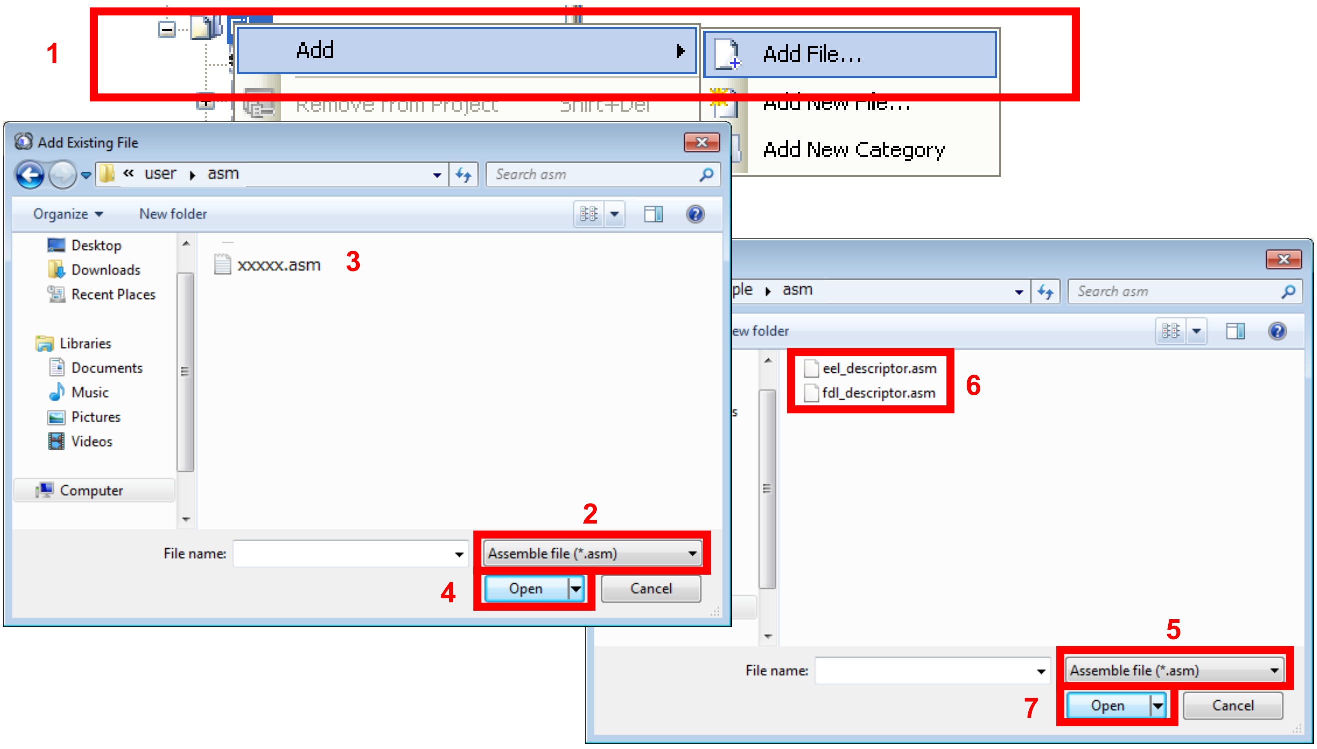
6.2.2 Building an Assembly Language Program

(1) Creating a project and specifying the source files

Create a project by using CS+. In the Project Tree window displayed on the left, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-5).

Click the Files of type drop-down list, select Assemble file (*.asm), and then register the user-created program file and the descriptor files for the EEPROM emulation library and data flash library (eel_descriptor.asm and fdl_descriptor.asm) as the source files.

Figure 6-5. Specifying the Assemble Files



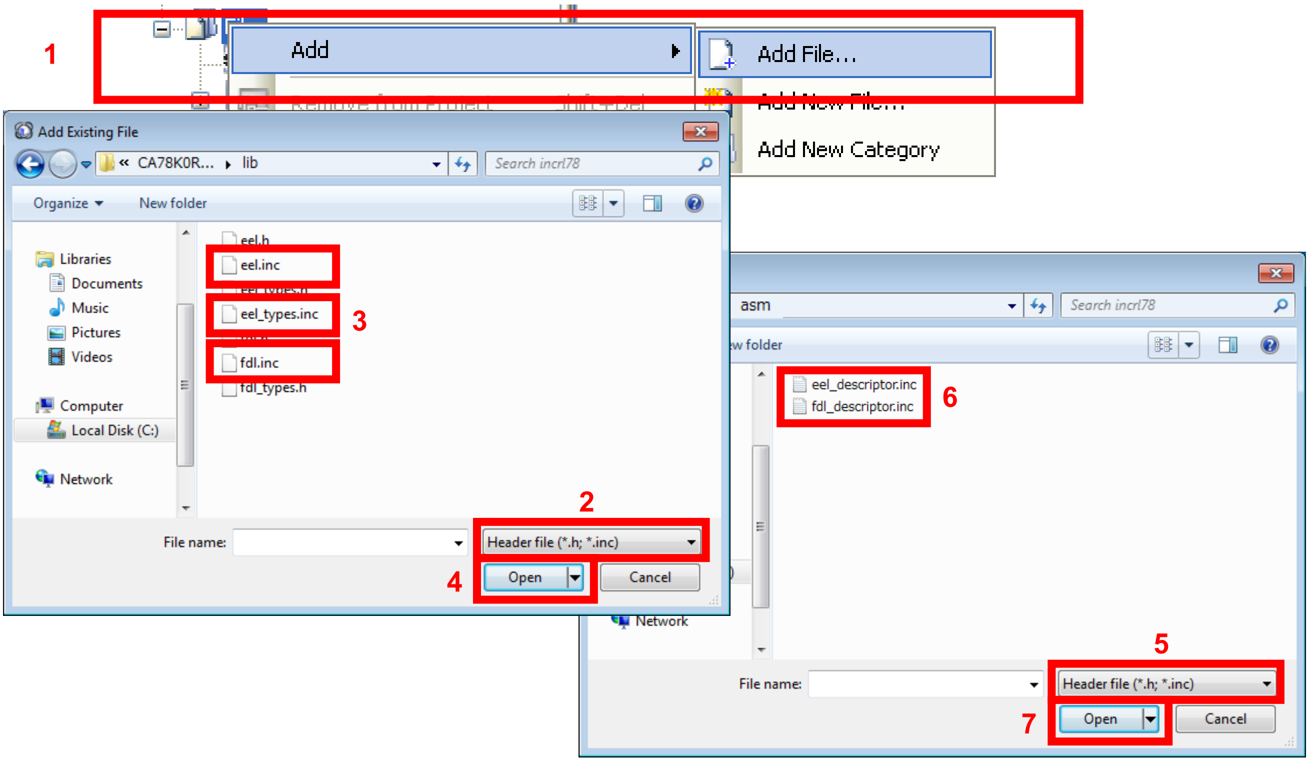
(2) Specifying the include file

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File.

The Add Existing File dialog box is displayed (as shown in Figure 6-6).

Click the Files of type drop-down list, select Header file (*.h;*.inc), and then register include files and the descriptor include files for the EEPROM emulation library and data flash library (eel.inc, eel_types.inc, fdl.inc, eel_descriptor.inc, fdl_descriptor.inc).

Figure 6-6. Specifying the Include Files

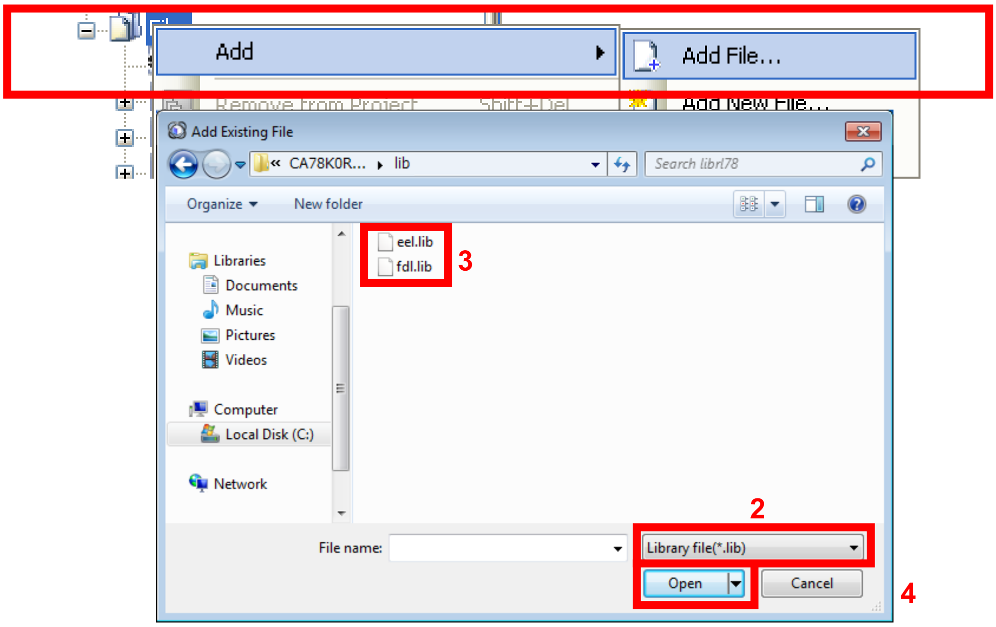


(3) Specifying the library file

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-7).

Click the Files of type drop-down list, select Library file (*.lib), and then register the EEPROM emulation library and data flash library files (eel.lib and fdl.lib).

Figure 6-7. Specifying the Library Files

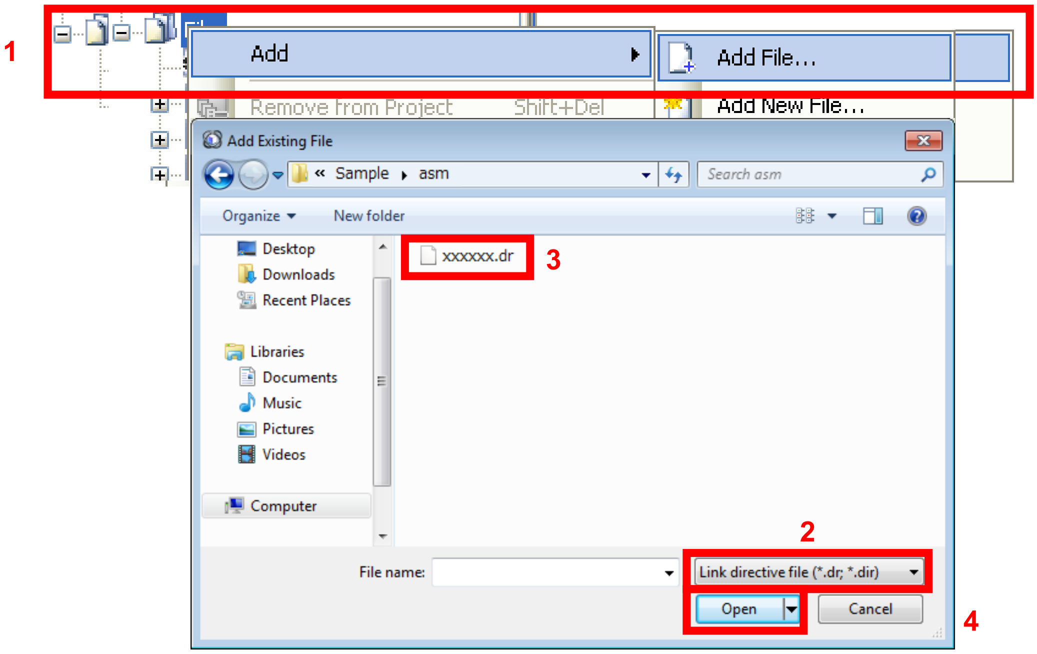


(4) Specifying the link directive file (only when the CA78K0R compiler is used)

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-8).

Click the Files of type drop-down list, select Link Directive File (*.dr;*.dir), and then register the link directive file that has the same name as the user-created program.

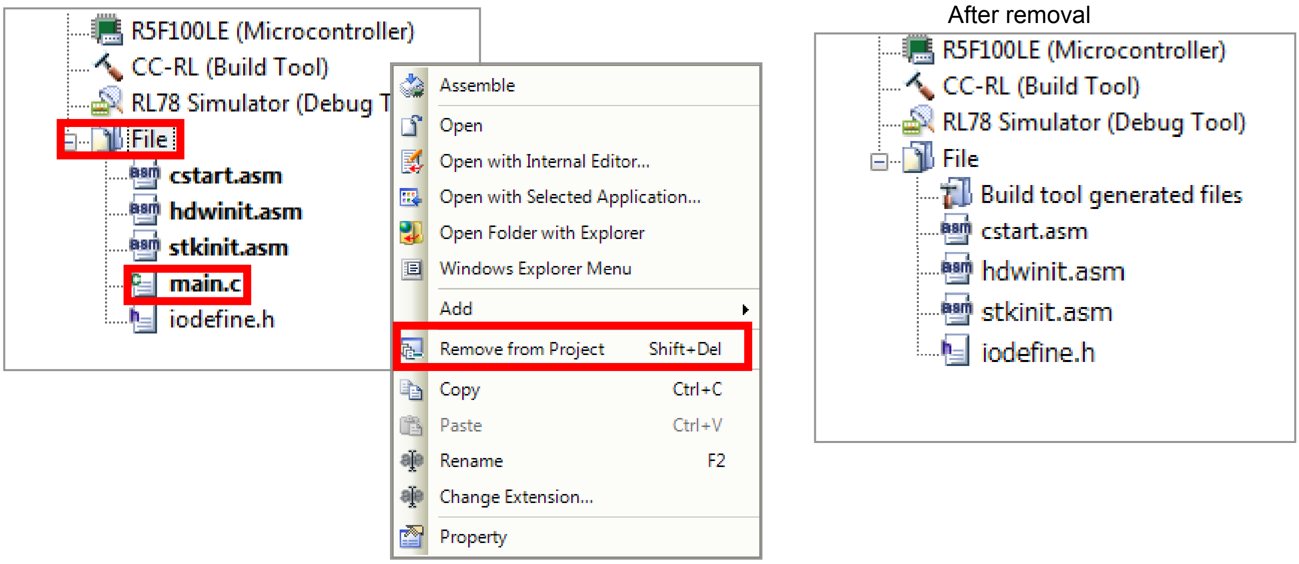
Figure 6-8. Specifying the Link Directive File



(5) Removing the automatically generated files (only when the CC-RL compiler is used)

CS+ for the CC-RL compiler automatically generates some files under the File node in the Project Tree window. Among these, the processing of the "main.c" file is included in the EEPROM emulation library. Therefore, remove this file from the target of the build process.

Figure 6-9. Removing the Automatically Generated Files



(6) Building

On the CS+ Build menu, click Build Project to build the project.

6.3 Note at Build

6.3.1 When the CA78K0R Compiler is Used

(1) When the on-chip debugging function is in use

After the on-chip debugging function is enabled in the CS+, building a program generates the following type of error.

```
RA78K0R error E3212: Default segment can't allocate to memory - ignored
                              Segment '??OCDROM' at xxxxxH-200H
```

This error occurs when the segment for the monitor area (OCDROM) used by the on-chip debugging function cannot be allocated. Therefore, to avoid this error, add the following code to the link directive file (*.dr) embedded in the project and prepare a separate area for allocating the segment.

```
MEMORY OCD_ROM : ( 0xxxxxH, 00200H )
```

Notes: 1. xxxxx: Start address of the location where the error occurred

2. The area name “OCD_ROM” is an example of the notation.

6.3.2 When the CC-RL Compiler is Used

(1) When the on-chip debugging function is in use

After the on-chip debugging function is enabled in the CS+, building a program may generate the following type of error.

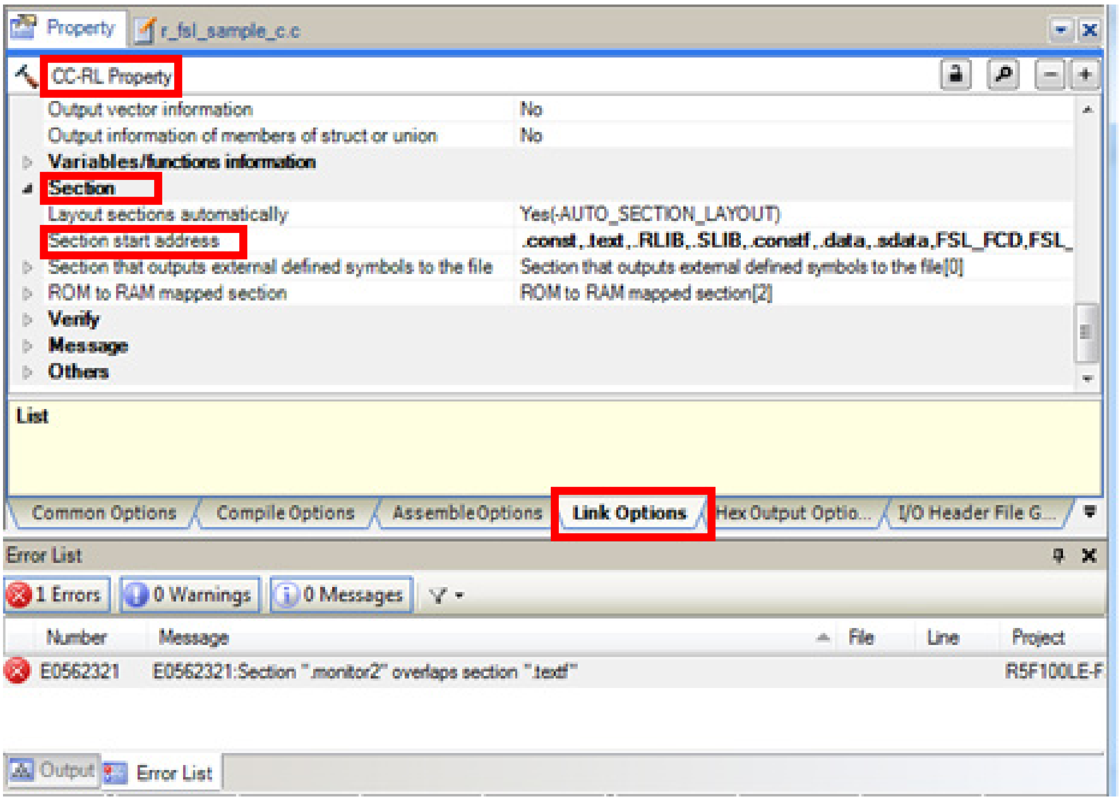
```
E0562321:Section ".monitor2" overlaps section "xxxxx
```

This error occurs when the section for the monitor area (OCDROM) used by the on-chip debugging function cannot be allocated. Therefore, to avoid this error, right click the CC-RL (Build Tool) node (1) in the CS+ Project Tree window, select Property to open the CC-RL Property panel (2), and select the Link Options tab (3). In the Section category (4), modify the setting for Section start address (5) so that no other areas overlap the area where the section for the on-chip debugger monitor is allocated (monitor2: the initial address range is 0xFE00 to 0xFFFF in R5F100LE). (See Figure 6-10.)

For details of the section settings, refer to the CC-RL Compiler User's Manual.

Remark 1. xxxxx: Indicates the section name.

Figure 6-10. Modifying the Section Allocation



Output vector information: No
Output information of members of struct or union: No
Variables/functions information
Section
Layout sections automatically: Yes(-AUTO_SECTION_LAYOUT)
Section start address: .const,.text,.RLIB,.SLIB,.constf,.data,.sdata,FSL_FCD,FSL_
Section that outputs external defined symbols to the file: Section that outputs external defined symbols to the file[0]
ROM to RAM mapped section: ROM to RAM mapped section[2]
Verify
Message
Others

Number	Message	File	Line	Project
E0562321	E0562321:Section ".monitor2" overlaps section ".textf"			R5F100LE-F

Chapter 7 How To Debug a Program

For details about how to perform debugging by using IECUBE or the on-chip debug emulator E1 or E20, see the following document.

Title
CubeSuite+ Integrated Development Environment User's Manual: RL78 Debug[CS+ for CA,CX]Note
CS+ Integrated Development Environment User's Manual: RL78 Debug Tool[CS+ for CC] Note

Note: You can download this document from the "CS+ Integrated Development Environment" page of the Renesas Electronics website.

7.1 Notes at Debug

The following describes notes apply when using the EEPROM Emulation Library Pack02 with the E1 or E20 on-chip debugging emulator.

(1) When a command of the EEPROM Emulation Library Pack02 is executed in a version older than CubeSuite+ Ver. 1.01 and the E1 or E20 on-chip debugging emulator is in use, do not execute a break until you have confirmed completion of the command by the sequencer. The sequencer will malfunction if a break occurs before the sequencer has completed the command.

(2) The simulator cannot be used to debug the flash library.

Chapter 8 Sample Program

The attached sample program (r_eel_sample_c.c) is provided to enable the usage method of the EEPROM Emulation Library Pack02 to be easily confirmed on the QB-R5F100LE-TB boards with R5F100LEA (RL78/G13) as the target microcontrollers. The sample program is just a reference example and the user program does not have to be created to match the sample program. The sample program should be used as a simple program to confirm operation.

The link directive file (r_eel_sample_c.dr) for the sample program has a purpose to specify that a stack or data buffer used by the sample program is not allocated to an area where allocation is prohibited [Note1]. When using the sample program, this file should also be embedded with the sample program.[Note2]

The sample program for the CC-RL compiler does not need the link directive file (r_eel_sample_c.dr), but sections should be allocated appropriately in the Section category on the Link Options tabbed page in the CS+ window so that a stack or data buffer used by the sample program is not allocated to an area where allocation is prohibited [Note1].

Notes: 1. For details, refer to chapter 6.2 “Software Resource” in the user’s manual.

2. The data in usage may be placed at an unintended area depending on how the environment in use or the program is changed. After an execution module is generated, the map file and allocation state of programs or data must be confirmed. For the definition method and allocation conditions of each code or data, refer to the user’s manual of the CS+.

8.1 Initial Settings of the Sample Program

The sample program operates with the following initial settings. When these settings need to be changed, modify the sample program.

- CPU operating frequency: High-speed on-chip oscillator 32 MHz
- Voltage mode: Full-speed mode

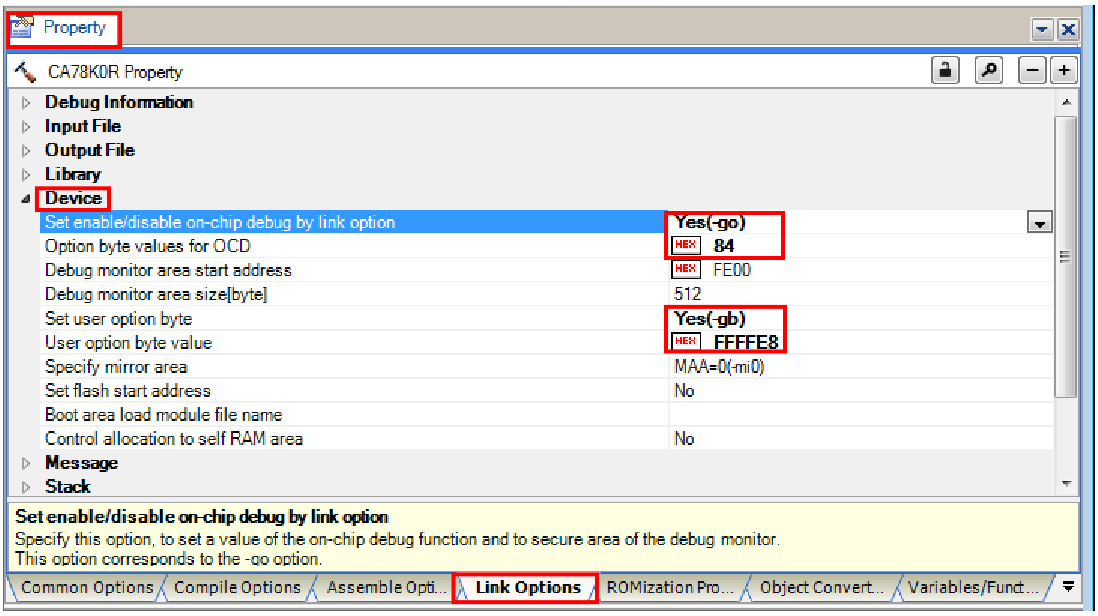
8.2 Settings of Option Byte and On-Chip Debugging

When performing on-chip debugging, set ”Set enable/disable on-chip debug by link option” to ”Yes” and specify “84” for “Option byte values for OCD”.

For the property of the CC-RL compiler, set ”Set debug monitor area” to ”Yes”.

The sample program normally operates by setting the high-speed on-chip oscillator at 32 MHz. After setting ”Set user option byte” to ”Yes” on Link Options, specify “xxxxE8” for ”User option byte value” and set the high-speed on-chip oscillator at 32 MHz.

Figure 8-1 (a) Setting of Option Byte (CA78K0R Compiler)



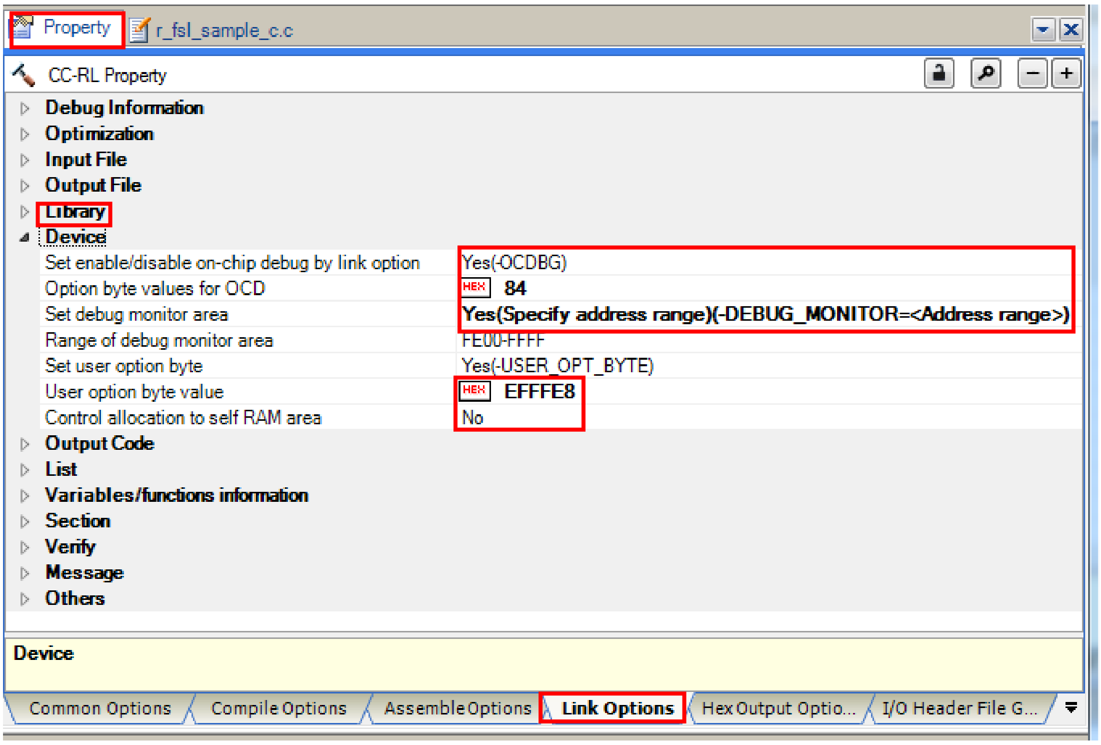
- Debug Information
- Input File
- Output File
- Library
- Device

Set enable/disable on-chip debug by link option	Yes(-go)
Option byte values for OCD	HEX 84
Debug monitor area start address	HEX FE00
Debug monitor area size[byte]	512
Set user option byte	Yes(-gb)
User option byte value	HEX FFFFE8
Specify mirror area	MAA=0(-mi0)
Set flash start address	No
Boot area load module file name	
Control allocation to self RAM area	No

- Message
- Stack

Set enable/disable on-chip debug by link option
Specify this option, to set a value of the on-chip debug function and to secure area of the debug monitor.
This option corresponds to the -go option.

Figure 8-1 (b) Setting of Option Byte (CC-RL Compiler)



- Debug Information
- Optimization
- Input File
- Output File
- Library
- Device

Set enable/disable on-chip debug by link option	Yes(-OCDBG)
Option byte values for OCD	HEX 84
Set debug monitor area	Yes(Specify address range)(-DEBUG_MONITOR=<Address range>)
Range of debug monitor area	FE00-FFFF
Set user option byte	Yes(-USER_OPT_BYTE)
User option byte value	HEX EFFFE8
Control allocation to self RAM area	No

- Output Code
- List
- Variables/functions information
- Section
- Verify
- Message
- Others

Device

8.3 Defining the On-Chip RAM Area

8.3.1 When the CA78K0R Compiler is Used

The following describes how to define the on-chip RAM area in the link directive file.

Normally, the entire on-chip RAM area is automatically defined as an area with the name "RAM" unless otherwise stated in the link directive file. The stack and data buffers are to be allocated to this area except when specifically stated otherwise[Note]. However, in this case, the stack and data buffers would be allocated by default to an area (FFE20H to FFEDFH in self-RAM) for which use by the EEPROM Emulation Library Pack02 is prohibited, so the program may not run correctly.

In the attached link directive file for the sample program, as a solution, re-define the area with the name "RAM" so that it does not include the above area, ensuring that stack and so on are not allocated to the area for which usage is prohibited.

```
MEMORY RAM        :(0FF080H, 000DA0H)
```

The above statement redefines the area with the name "RAM" to be the DA0H bytes area starting from the address FF080H (FF080H to FFE1FH)[Note]. This prevents attempted use of the area which the EEPROM Emulation Library Pack02 is prohibited to use by excluding the prohibited portion from the area with the name "RAM".

However, if this is the only change setting that is explicitly made, the area from FFE20H to FFEDFH is also unusable for any other purpose. Accordingly, separately add the following definition. No particular restrictions apply to the name of this area.

```
MEMORY SADDR_RAM:(0FFE20H, 0001E0H)
```

If there is a self-RAM area, automatic allocation of variables to this area can be restricted by defining its range as an area with the name "SELFRAM".

```
MEMORY SELFRAM   :(0FEF00H, 000180H)
```

An example of the settings for an RL78/G13 (the product with 4 KB of RAM and 64 KB of ROM) is given below.

(In this example, the general-purpose register and the SFR area are included in RAM_SADDR, however, they can be omitted.)

```
; ----------------------------------------------------------
; Define new memory entry for Self-RAM
; ----------------------------------------------------------
MEMORY SELFRAM     : ( 0FEF00H, 000180H ) ←———  Definition of the self-RAM area

; ----------------------------------------------------------
; Redefined default data segment RAM
; ----------------------------------------------------------
MEMORY RAM          : ( 0FF080H, 000DA0H ) ←———  Definition of the RAM area to be used normally

; ----------------------------------------------------------
; Define new memory entry for saddr area
; ----------------------------------------------------------
MEMORY RAM_SADDR : ( 0FFE20H, 0001E0H )←———  Definition of the area from FFE20H to FFFFFH
```

Note: The CA78K0R linker allocates data with a non-specified destination for allocation (segment types DSEG and BSEG) to the on-chip RAM area according to the re-allocation attribute of the data. Accordingly, specific data may not be allocated to the area with the name “RAM” in some situations.

For details on the methods of defining and allocating the individual categories of data, refer to the user’s manual for CS+.

Reference to the map file (*.map) generated at the time of building is required to confirm the state of allocation.

8.3.2 When the CC-RL Compiler is Used

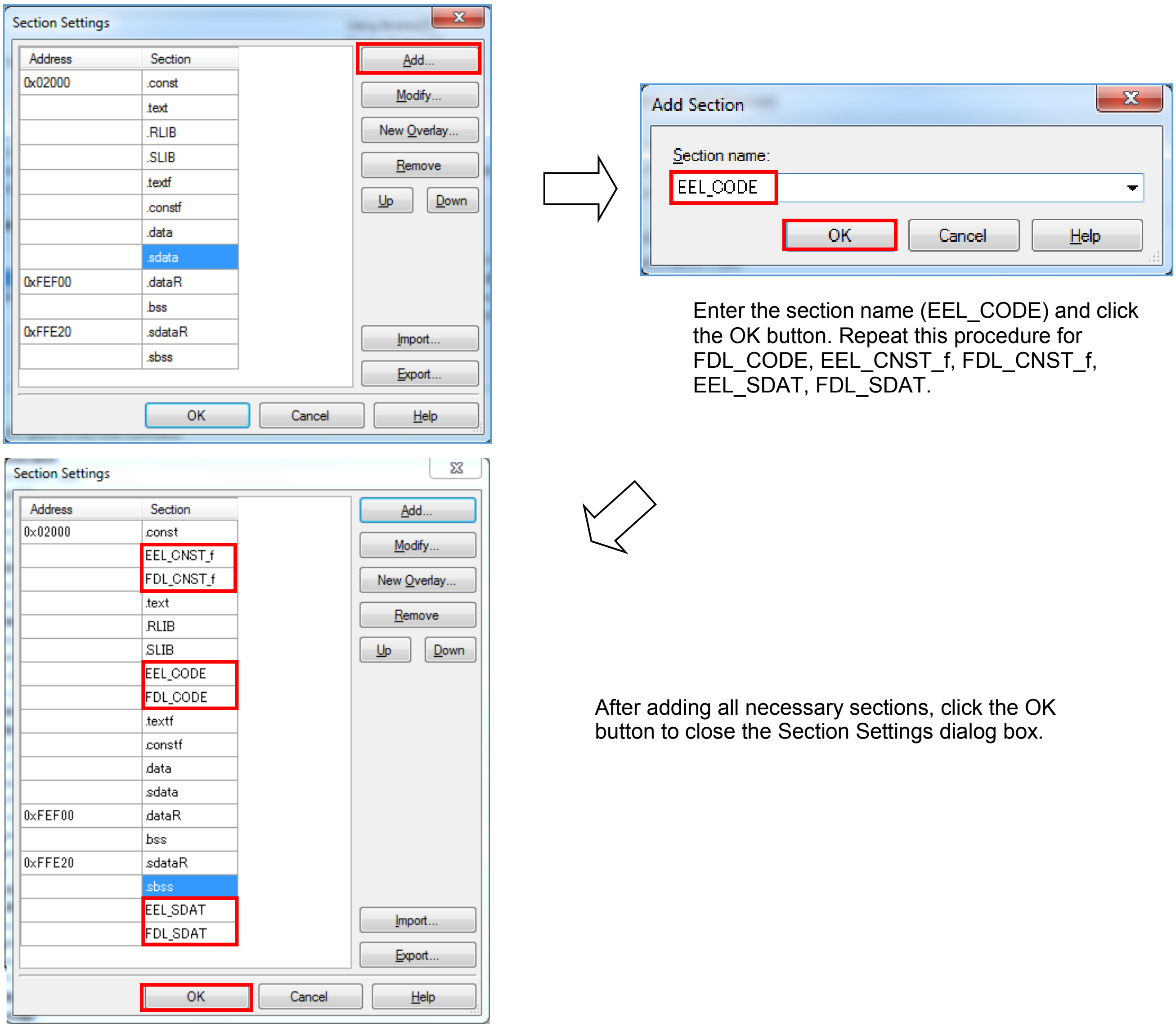
(1) Adding the include path

In CS+ for the CC-RL compiler, no include path is specified in the initial state; the include paths for the header files used by the EEPROM emulation library need to be added. The EEPROM emulation library uses header files ”eel_user_types.h”, ”eel.h”, ”fdl.h”, ”eel_types.h”, ”fdl_types.h”, ”eel_descriptor.h”, ”fdl_descriptor.h” and "iodefine.h" (this file is automatically generated by CS+). Add the include path for each header file in the Additional include paths in the Preprocess category on the Compile Options tabbed page.

(2) Defining sections

When CS+ for the CC-RL compiler is used, the sections used for the ROM and RAM areas need to be defined. Sections can be defined in the Section category on the Link Options tabbed page in the CS+ window. When the Layout sections automatically property is set to No, select the Section start address property to open the Section Settings dialog box and add the sections necessary for the EEPROM emulation library to the ROM area (Figure 8-2). (In this example, the EEL_CODE, FDL_CODE, EEL_CNST_f, FDL_CNST_f, EEL_SDAT, FDL_SDAT sections that are necessary for operation of the sample are added.)

Figure 8-2. Example of Section Settings for EEPROM emulation Library (ROM Area)



Section Settings

Address	Section
0x02000	.const
	.text
	.RLIB
	.SLIB
	.textf
	.constf
	.data
	.sdata
0xFEF00	.dataR
	.bss
0xFFE20	.sdataR
	.sbss

Add Section

Section name: EEL_CODE

Enter the section name (EEL_CODE) and click the OK button. Repeat this procedure for FDL_CODE, EEL_CNST_f, FDL_CNST_f, EEL_SDAT, FDL_SDAT.

Section Settings

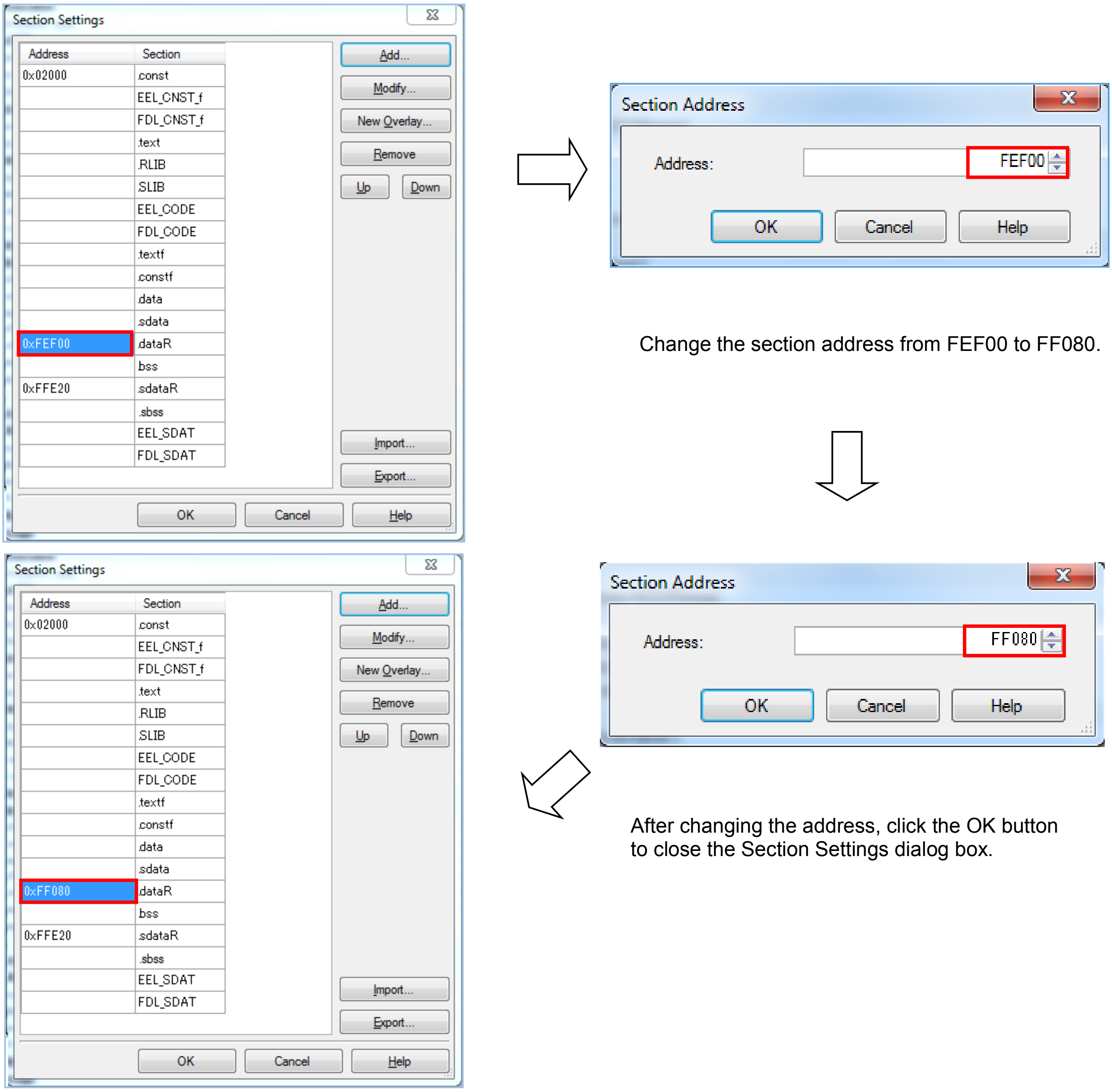
Address	Section
0x02000	.const
	EEL_CNST_f
	FDL_CNST_f
	.text
	.RLIB
	.SLIB
	EEL_CODE
	FDL_CODE
	.textf
	.constf
	.data
	.sdata
0xFEF00	.dataR
	.bss
0xFFE20	.sdataR
	.sbss
	EEL_SDAT
	FDL_SDAT

After adding all necessary sections, click the OK button to close the Section Settings dialog box.

(3) Allocating the Self-RAM Area

In the initial state of the section settings in CS+ for the CC-RL compiler, the user RAM area is allocated at the beginning of the internal RAM area (from address FEF00H for R5F100LEA, which is the target microcontroller of the sample program). However, in R5F100LEA, the EEPROM emulation library uses the address range from 0xFEF00 to 0xFF07F as the self-RAM area. Therefore, the user RAM area must be allocated outside this area. In this example, the user data start address 0xFEF00 is changed to 0xFF080.

Figure 8-3. Example of Changing the User RAM Area Allocation (RAM Area)



Address	Section
0x02000	.const
	EEL_CNST_f
	FDL_CNST_f
	.text
	.RLIB
	.SLIB
	EEL_CODE
	FDL_CODE
	.textf
	.constf
	.data
	.sdata
0xFEF00	.dataR
	.bss
0xFFE20	.sdataR
	.sbss
	EEL_SDAT
	FDL_SDAT

Address: FEF00

Change the section address from FEF00 to FF080.

Address: FF080

After changing the address, click the OK button to close the Section Settings dialog box.

Address	Section
0x02000	.const
	EEL_CNST_f
	FDL_CNST_f
	.text
	.RLIB
	.SLIB
	EEL_CODE
	FDL_CODE
	.textf
	.constf
	.data
	.sdata
0xFF080	.dataR
	.bss
0xFFE20	.sdataR
	.sbss
	EEL_SDAT
	FDL_SDAT

Note: The sections including the user-specified sections are automatically re-allocated when the Layout sections automatically property is temporarily set to No, the user RAM allocation is changed, and then the property is again set to Yes. In this case, sections may be allocated to areas that are not specified by the user; that is, data may be placed in unintended areas. Be sure to refer to the map file to check if the software resources (especially RAM data) used by the EEPROM emulation library are placed in relocatable areas.

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.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, 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 1207, 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 India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0