To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 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 1st, 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.

SuperH RISC engine C/C++ Compiler Package

APPLICATION NOTE: [Compiler use guide] Object usage guide

This document describes how to use objects.

Table of contents

1. Changing the Section of a Specific Symbol in a Library

1.1 Overview

This chapter describes how to change the section of a specific symbol in a library. In the example used in this chapter, the section for the **_INITSCT()** function in the standard library will be changed.

When a program is to be copied from ROM to RAM for execution, the **_INITSCT()** function is used. Note, however, that, like user functions, standard library functions are placed in the **P** section. Accordingly, an attempt to copy the **P** section to RAM will fail because the **_INITSCT()** function itself is located in the **P** section that is being copied. To perform the copy successfully, the section for the **_INITSCT()** function must be changed to another section.

The section of a specific symbol can be changed by using either the librarian interface or the optimizing linkage editor.

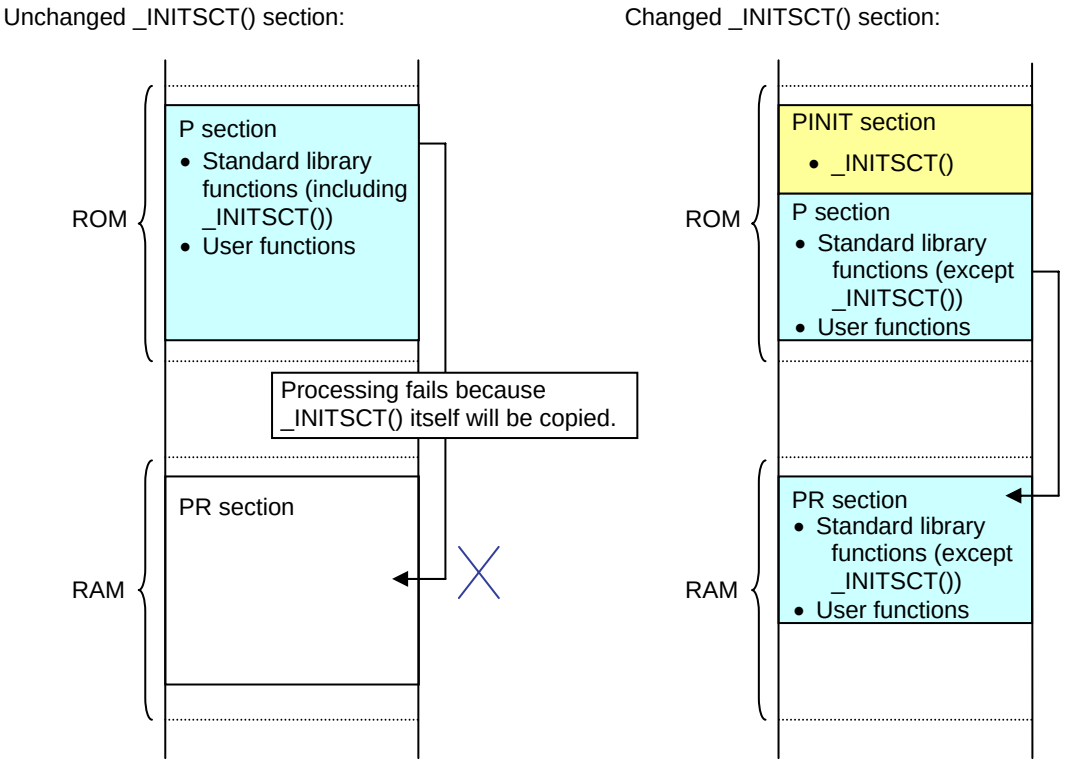


Figure 1-1

1.2 Librarian Interface

The librarian interface can be used to change the section of a specific module in a library. This section (*1.2*) describes how to use the librarian interface to change the section of a specific symbol in a library.

To change the section, start the librarian interface by choosing **Renesas H Series Librarian Interface** from the **tools** menu of High-performance Embedded Workshop (*Renesas IDE* hereafter). Then select the target library by choosing **Open** from the **File** menu of the librarian interface. Selecting the target library displays a list of the object modules in the library. From the list, select the section containing the symbol whose section is to be changed, and then choose **Rename Section** from the **Action** menu.

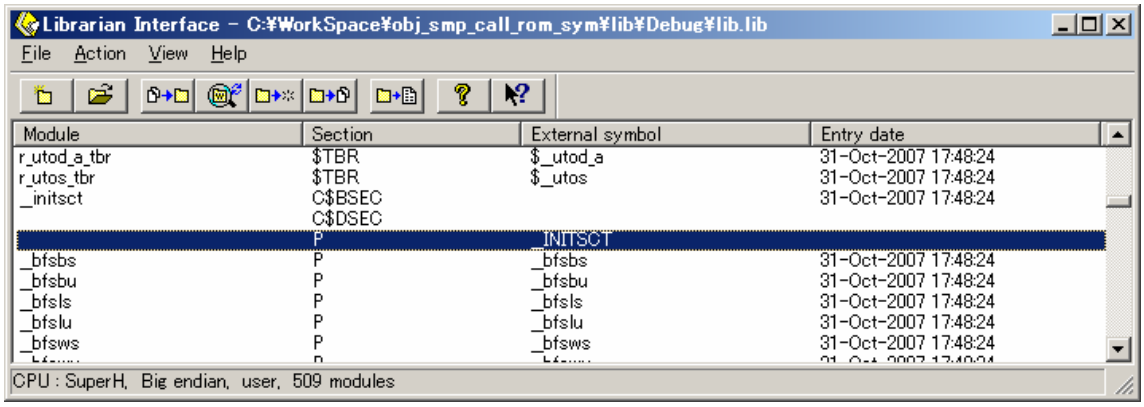


Figure 1-2

In the displayed **Rename Section** dialog box, click the **After** button to display the **After** dialog box. In the **Section Name After Renaming** text box, enter the name of the new section, and click the **OK** button.

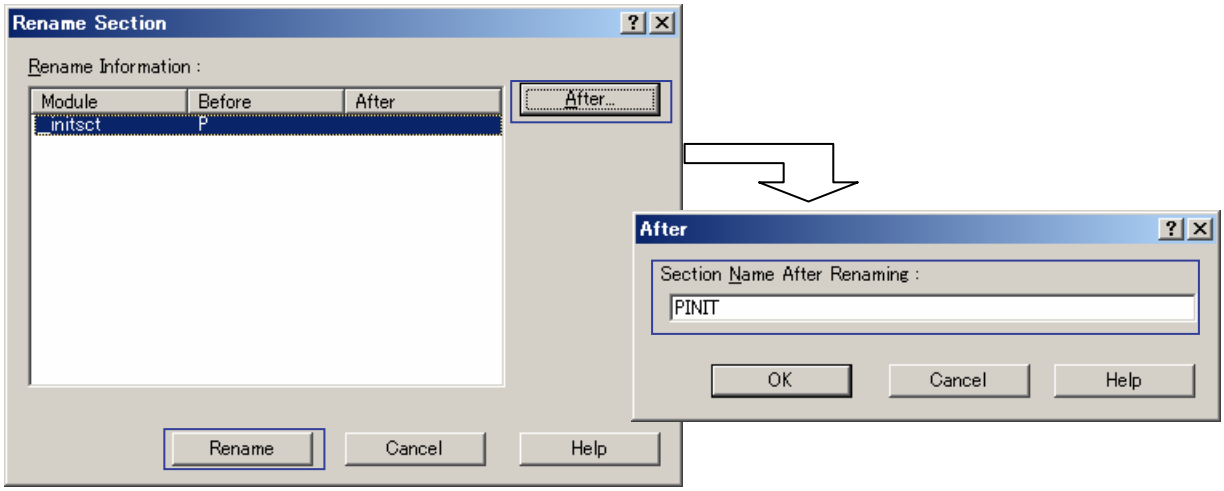


Figure 1-3

In the **Rename Section** dialog box, click the **Rename** button. The section is changed.

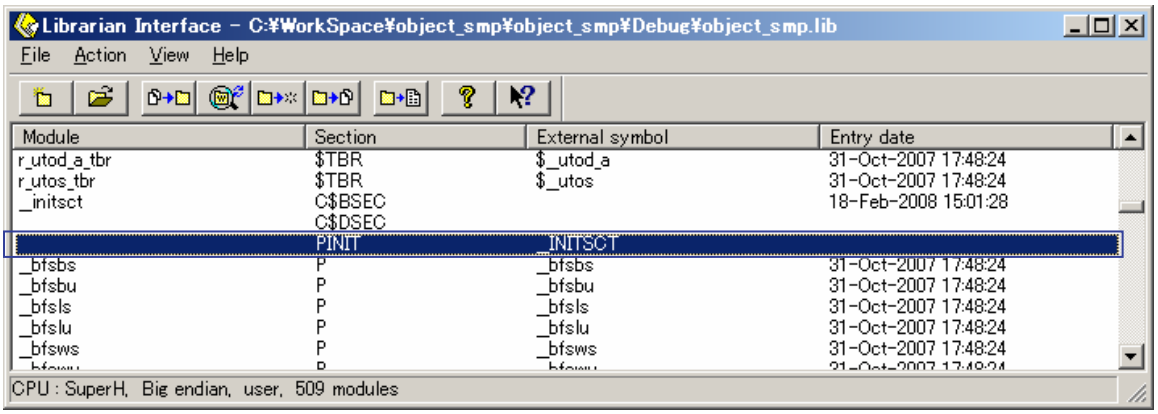


Figure 1-4

1.3 Optimizing Linkage Editor

This section describes how to use the optimizing linkage editor to change the section of a specific symbol in a library.

First, output the information about the object modules in the target library (library listing), including the information about the sections and symbols. The library listing including the information about the sections and symbols can be output by specifying both the **list** and **show =symbol,section** options. For example, to output the library listing for the **stdlib.lib** standard library, execute the optimizing linkage editor as follows:

```
optlnk -list -show=symbol,section -library=stdlib.lib -form=library
-output=tmp_stdlib.lib
```

The library listing uses the following headings:

```
          ...
*** Library List ***

MODULE    LAST UPDATE
  SECTION
    SYMBOL
```

MODULE:	Name of an object module
LAST UPDATE:	The date that the object module was registered. If a registered object module has been updated, the date of the last update is output.
SECTION	Name of a section in the object module
SYMBOL	Names of symbols in the section

The following is an example of the **tmp_stdlib.lib** library listing output by the optimizing linkage editor. This library listing shows that the **_INITSCT()** function is located in the **P** section of the **__initsct** object module.

```
                    ...
*** Library List ***

MODULE    LAST UPDATE
  SECTION
    SYMBOL

div
          30-Oct-2006 16:20:00
  P
    _div
    _ldiv
                    ...
__initsct
          30-Oct-2006 16:20:00
  C$BSEC
  C$DSEC
  P
    __INITSCT
                    ...
```

Next, specify the **rename** option to change the section of a specific module in the library. The format of the **rename** option is as follows:

rename=<suboption>[,…]
<suboption> : [<object module>](<old section>=<new section>)

To change the section of the **__initsct** object module containing the **_INITSCT()** function from **P** to **PINIT**, execute the optimizing linkage editor as follows. The **new_stdlib.lib** library is generated and contains the change.

```
optlnk -rename=__initsct(P=PINIT) -library=stdlib.lib -form=library
-output=new_stdlib.lib
```

The following shows the library listing for the generated **new_stdlib.lib** library.

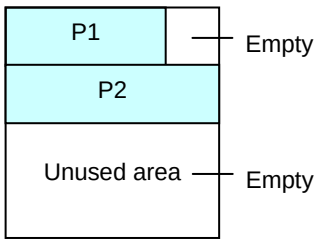
```
                        ...
*** Library List ***

MODULE    LAST UPDATE
  SECTION
    SYMBOL

div
          30-Oct-2006 16:20:00
  P
    _div
    _ldiv
                        ...
__initsct
          20-Feb-2008 15:35:14
  C$BSEC
  C$DSEC
  PINIT
    __INITSCT
                        ...
```

2. Filling Unused Areas with Dummy Data

2.1 Overview

A ROM image file (a S-Type file, Intel HEX file, or binary file) generated with the default settings contains only programs and ROM data. Therefore, if the ROM image file is written to memory, unused areas might exist. To generate a ROM image file without any unused areas, execute the optimizing linkage editor with the **space** option specified.

ROM image file with unused areas:



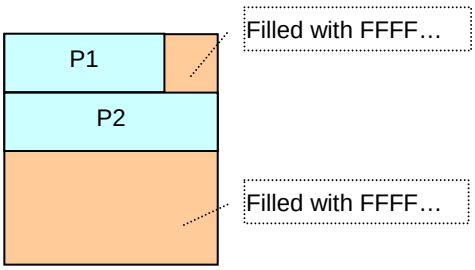
S-Type format file:

```
S00E000073706163653120206D6F7424
S1092000000900090009BB
S10B2010000900090009000 9A0
S9030800F4
```

Underlining indicates the data.

Unused areas are filled with FF

ROM image file with unused areas filled with dummy data:



S-Type format file:

```
S00E000073706163653120206D6F7424
S1092000000900090009BB
S1052006FFFFD6
S10B2008000900090009000 9A8
S1132010FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCC
S9030800F4
```

Underlining indicates the data.

Figure 2-1

2.2 Procedure

This section describes how to execute the optimizing linkage editor with the **space** option specified to generate a ROM image file that does not have any unused areas. First, the output range for the ROM image must be specified to generate the ROM image file. To specify the output range, specify the following settings on the **Link/Library** page in the **SuperH RISC engine Standard Toolchain** dialog box of Renesas IDE:

Category: Select **Output**.
Type of output file: Select **Hex via absolute**, **Stype via absolute**, or **binary via absolute**.
Show entries for: Select **Divided output files**.
Divide output files: Select this check box.
Add: Click this button to display the **Add output file** dialog box, and specify the output range and output file name.

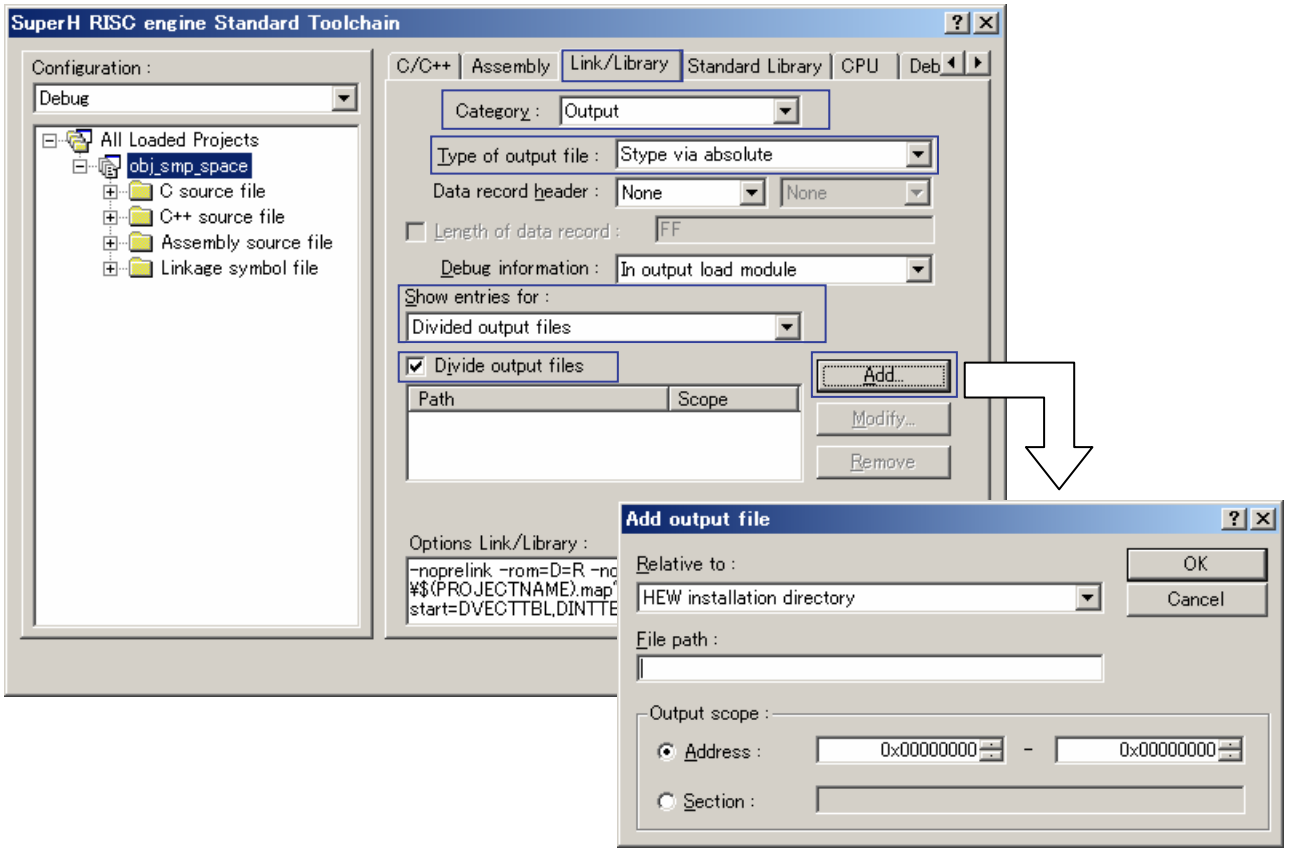


Figure 2-2

Next, to use the **space** option to fill unused areas with dummy data, specify the following settings:

Show entries for: Select **Specify value filled in unused area**.
Output padding data: Select **Custom** or **Random**.
If **Custom** is selected, specify a value.

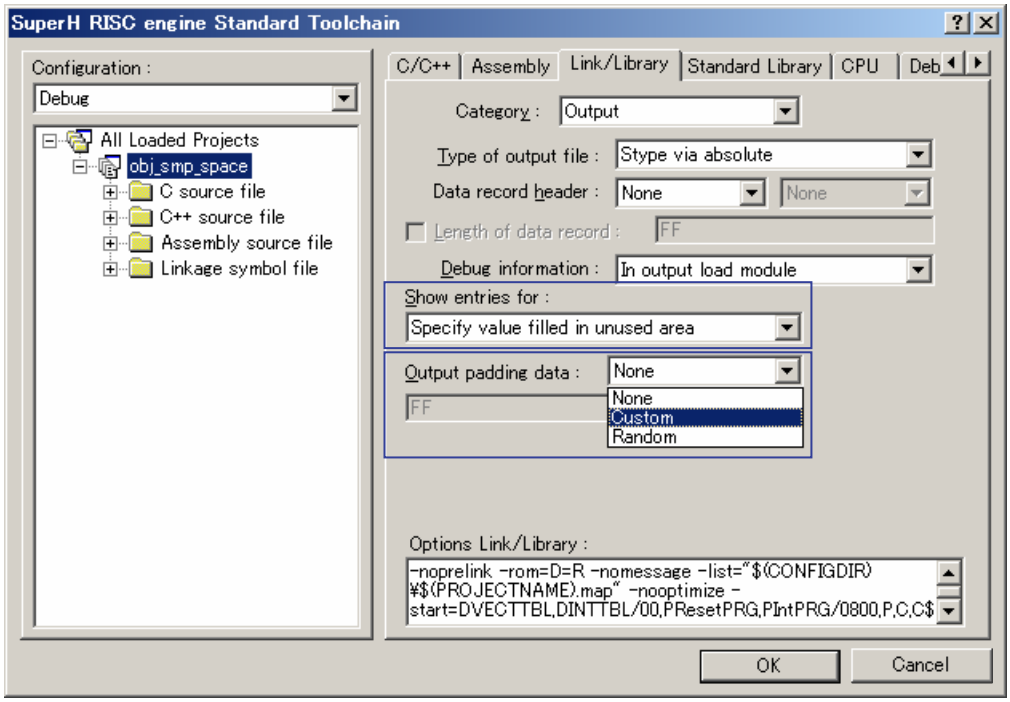


Figure 2-3

3. Calling a Symbol Fixed on ROM

3.1 Overview

This chapter describes how to call a symbol that has already been fixed in ROM from a new load module. In the example used in this chapter, a newly created load module is loaded into RAM, and the load module calls a load module function that has been fixed in ROM.

Before a function fixed in ROM can be called, the function address of the ROM program must be known. When the ROM program load module is generated, a symbol address file that contains symbol address information must be output. Since this file is written with assembler instructions, it can be used as an assembly source file. Calling a function for the ROM program from the RAM program is made possible by assembling and linking the symbol address file when the RAM program load module is generated.

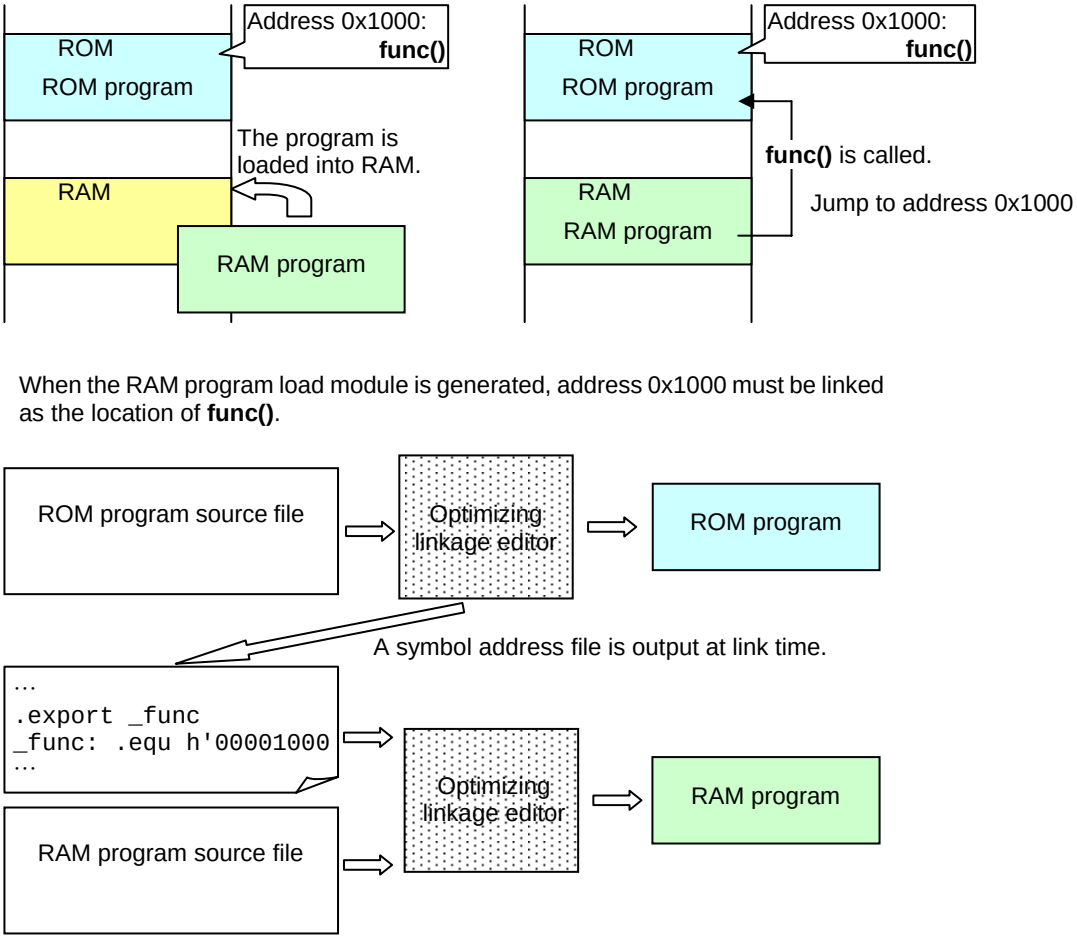


Figure 3-1

3.2 Procedure

This section describes how to generate the load module of the RAM program described in the previous section.

First, generate the symbol address file while generating the ROM program load module. To generate the symbol address file, specify the following settings on the **Link/Library** page in the **SuperH RISC engine Standard Toolchain** dialog box of Renesas IDE. When the load module is generated, the symbol address file for the specified section is output with the extension **fsy**.

Category: Select **Section**.
Show entries for: Select **Symbol file**.
Add: Click this button, and specify the section for which the symbol address file is to be generated.

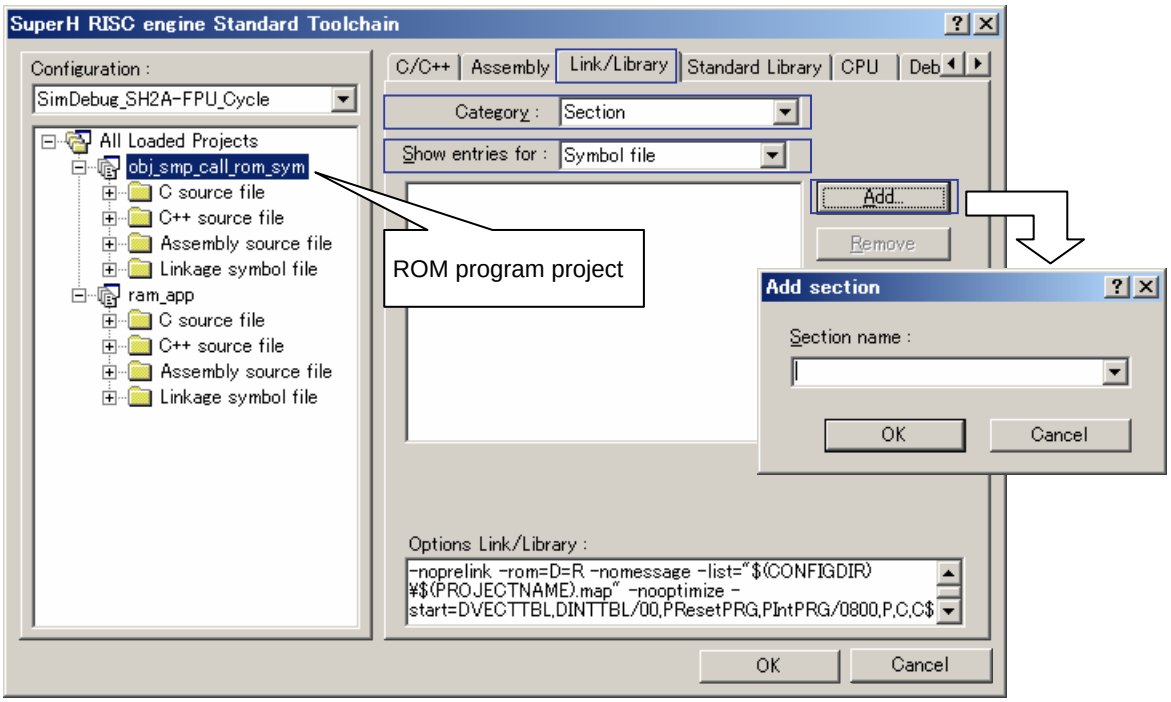


Figure 3-2

Next, bring the generated symbol address file into the project when creating the RAM program that generates the load module.

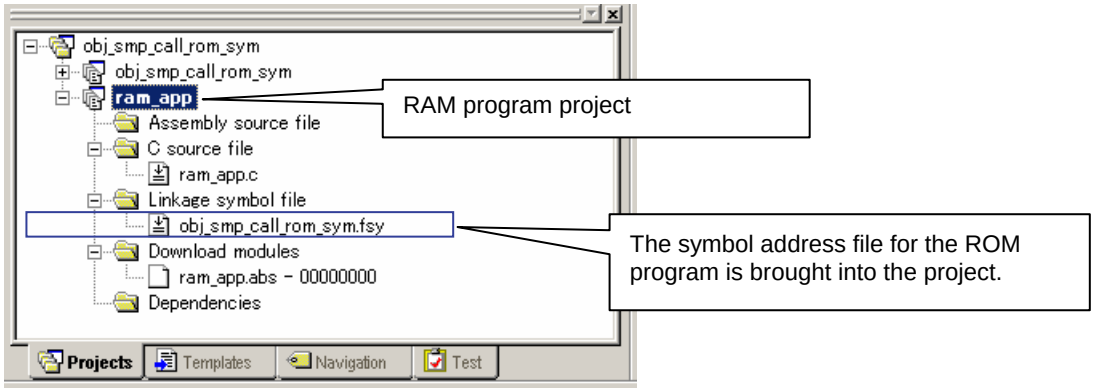


Figure 3-3

4. Library Files and Relocatable Files

4.1 Differences Between a Library File and a Relocatable File

This section describes the differences between a library file and a relocatable file. A library file can be linked only to object modules that include referenced symbols. By contrast, a relocatable file can be linked to all object modules.

(1) Library file

A library file can be linked only to an object module that includes a symbol that will be referenced either directly or indirectly. In the example shown in Figure 4-1, since **func1_A()** of library object module **module1.obj** is referenced by Object 1, **module1.obj** is linked. In addition, since **func3_A()** of library object module **module3.obj** is referenced by Object 3, **module3.obj** is also linked.

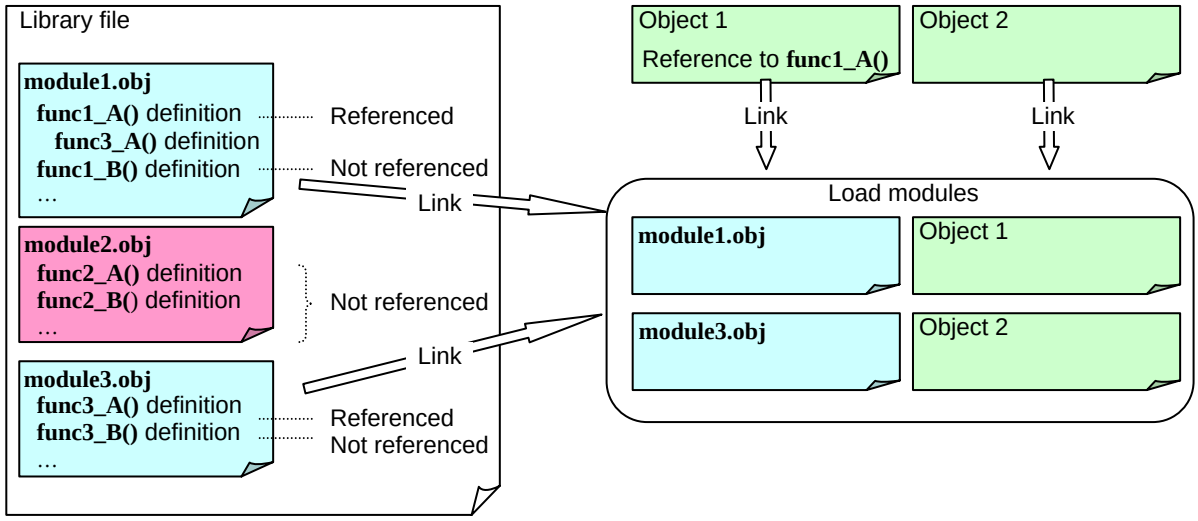


Figure 4-1

(2) Relocatable file

A relocatable file is linked to all object modules whether or not the modules include referenced symbols. In the example shown in Figure 4-2, the relocatable file is linked to **module2.obj**, which is an object module for the relocatable file, even though the **module2.obj** symbol is not referenced from anywhere.

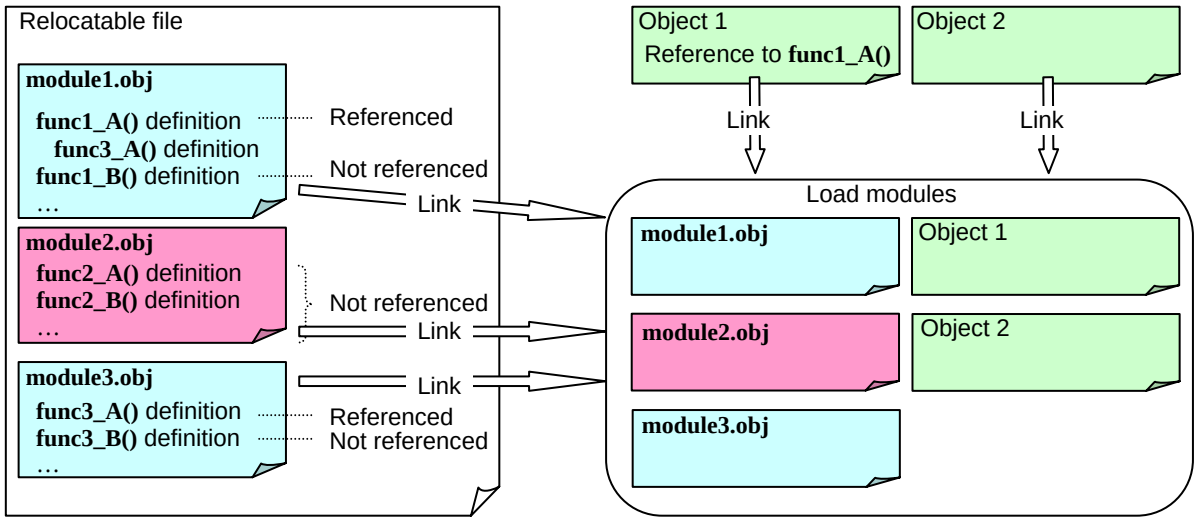


Figure 4-2

4.2 Linking Only Functions That Will Be Used

To link only functions that will be used, create a library file with source files, each of which defines only one function. By using a library file, only the functions that will be used are linked. In the example shown in the following figure, the library file is linked to only the **module1A.obj** module, which includes **func1_A()** directly referenced by Object 1, and to the **module3A.obj** module, which includes the indirectly referenced **func3_A()**.

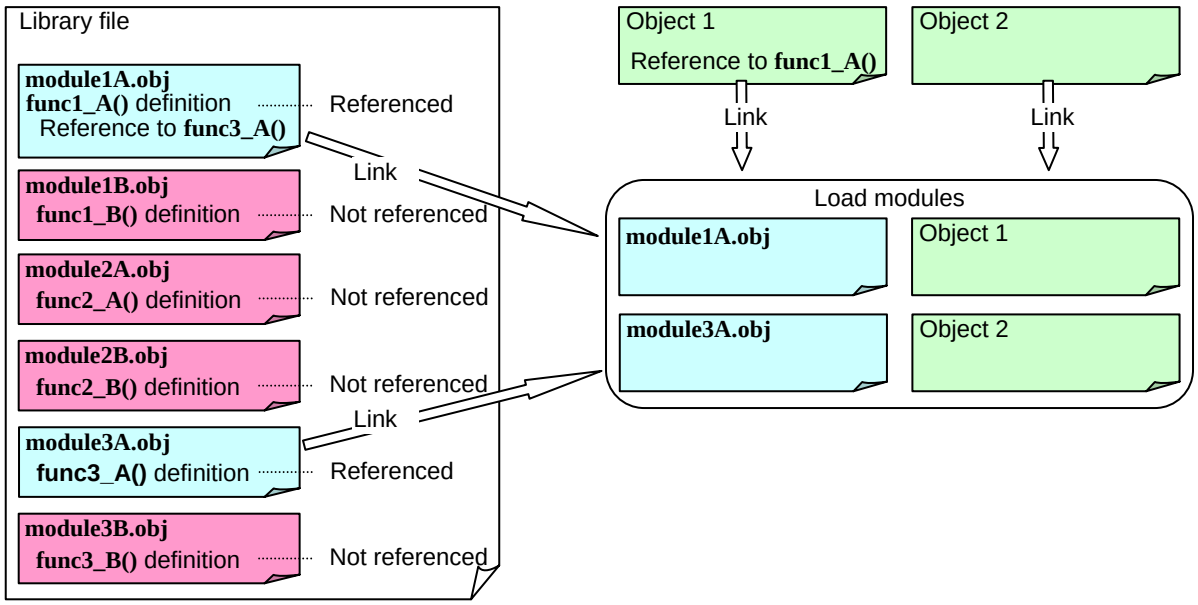


Figure 4-3

4.3 Enabling All Object Modules in an Existing Library File to Be Linked

When, as described earlier, an application is provided as a library file, only the functions actually called are linked. However, all functions in the library file, including those that are not currently called, might need to be linked for future expansion. In cases such as this, convert the library file into a relocatable file before linkage.

There are several ways that a library file can be converted into a relocatable file. This section shows how to create a relocatable file with one or more object files that have been extracted from a library file.

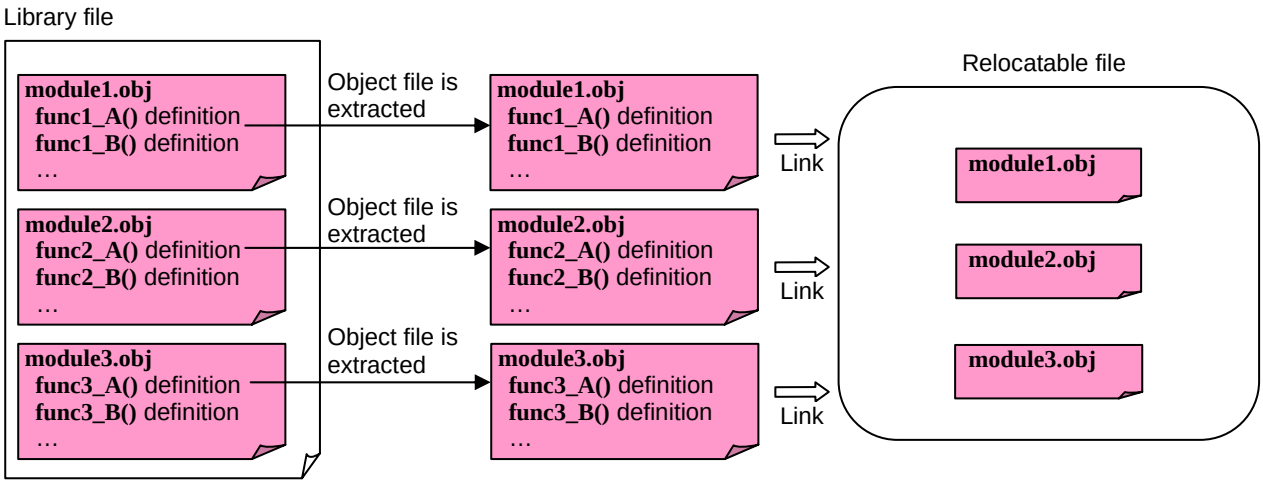


Figure 4-4

Object files can be extracted from a library file by using either the librarian interface or the optimizing linkage editor.

(1) Using the librarian interface to extract object files

The following describes how to use the librarian interface to extract object files.

From the module list of the librarian interface (Figure 1-2), select the object modules to be extracted. Then, from the **Action** menu, choose **Extract** to display the **Extract** dialog box. In the dialog box, specify the following settings and click the **OK** button.

The selected object modules will be extracted as object files.

Output file type: Select **Object file**.
Output folder: Specify the folder to which the extracted object files are to be output.

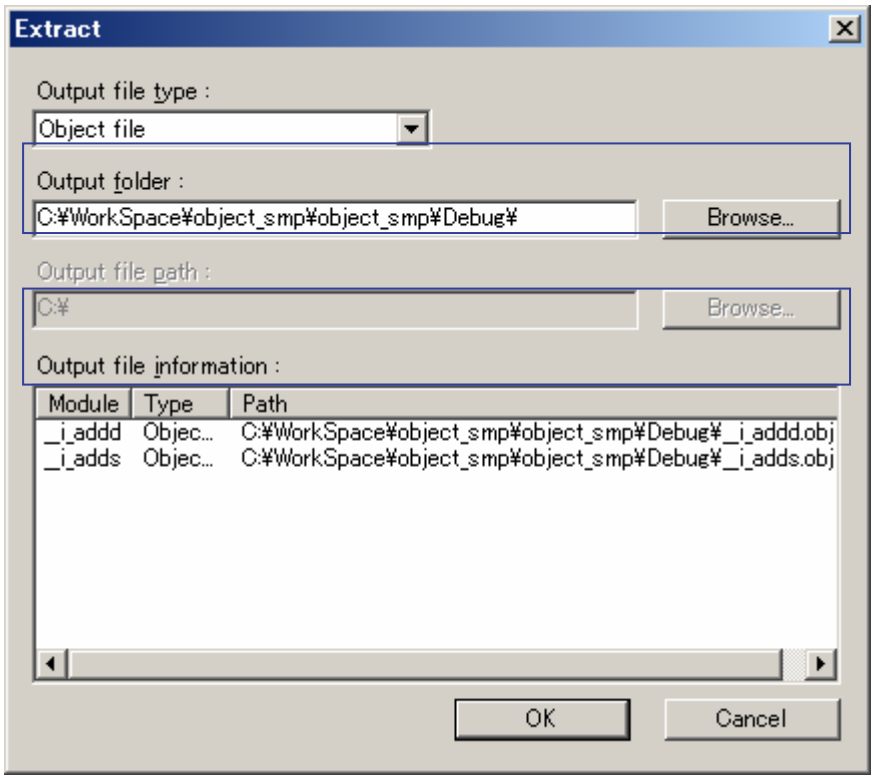


Figure 4-5

In Renesas IDE, bring the extracted object files into the project.

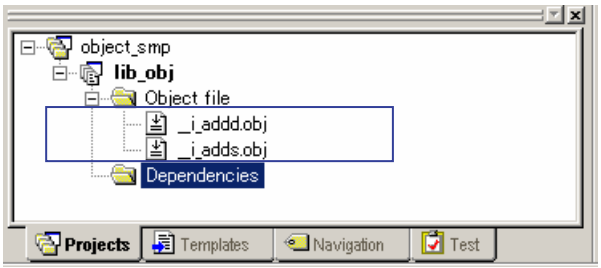


Figure 4-6

Specify the following settings on the **Link/Library** page in the **SuperH RISC engine Standard Toolchain** dialog box to generate a relocatable file from the object files brought into the project:

Category: Select **Output**.
Type of output file: Select **Relocatable**.

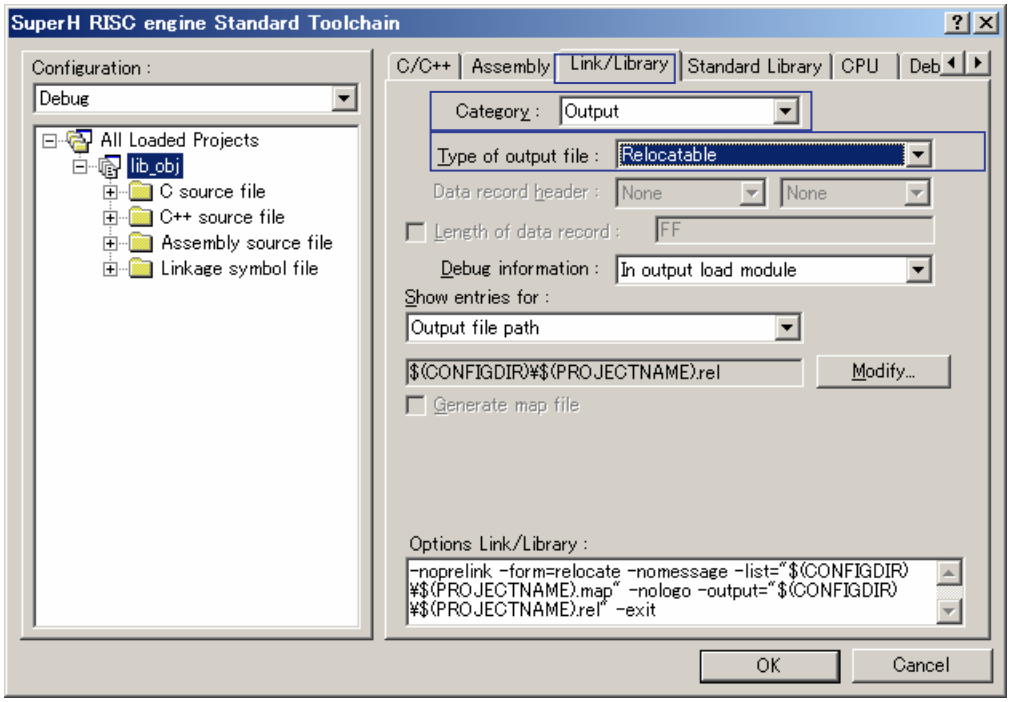


Figure 4-7

(2) Using the optimizing linkage editor to extract object files

The following describes how to use the optimizing linkage editor to extract object files.

First, output a library listing to acquire information about the object modules in the library. For details about how to output a library listing, see *1.3 Optimizing Linkage Editor*. Next, use the **extra** option to specify object modules as follows. The specified object modules are extracted as object files.

optlnk -extra=<object modules> -form=object -library=<library file>

Finally, use the **form=relocate** option to generate a relocatable file. For example, to generate a relocatable file from the object files **__i_addd.obj** and **__i_adds.obj**, specify the following:

optlnk __i_addd.obj __i_adds.obj -form=relocate

5. Creating a Load Module with a Physical Address

5.1 Overview

When an object is to be generated for a CPU that has a logical address space, address resolution is based on the addresses in the logical address space. For example, if a project for an SH7750 is generated in Renesas IDE, sections are allocated as shown in Figure 5-1. For example, RSTHandler is allocated at 0xA0000000, which represents an address in the physical address space. If this load module is downloaded to the logical address space in Renesas IDE, the first three bits are automatically ignored. That is, RSTHandler is downloaded at address 0x0000000, not 0xA0000000. However, some third-party flush writing tools might not have a function that ignores the first three bits. If such a tool is used to download a load module generated in a logical address space, the load module is downloaded as is to the logical address. This means that RSTHandler in this example is downloaded at address 0xA0000000. This problem is avoided by creating a load module whose addresses are resolved by using logical addresses and whose data is allocated by using physical addresses.

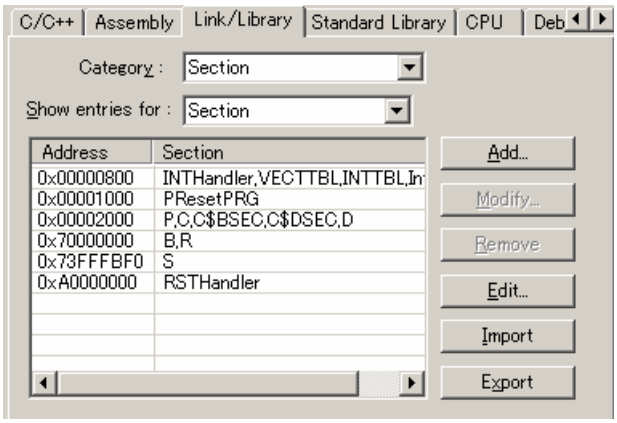


Figure 5-1

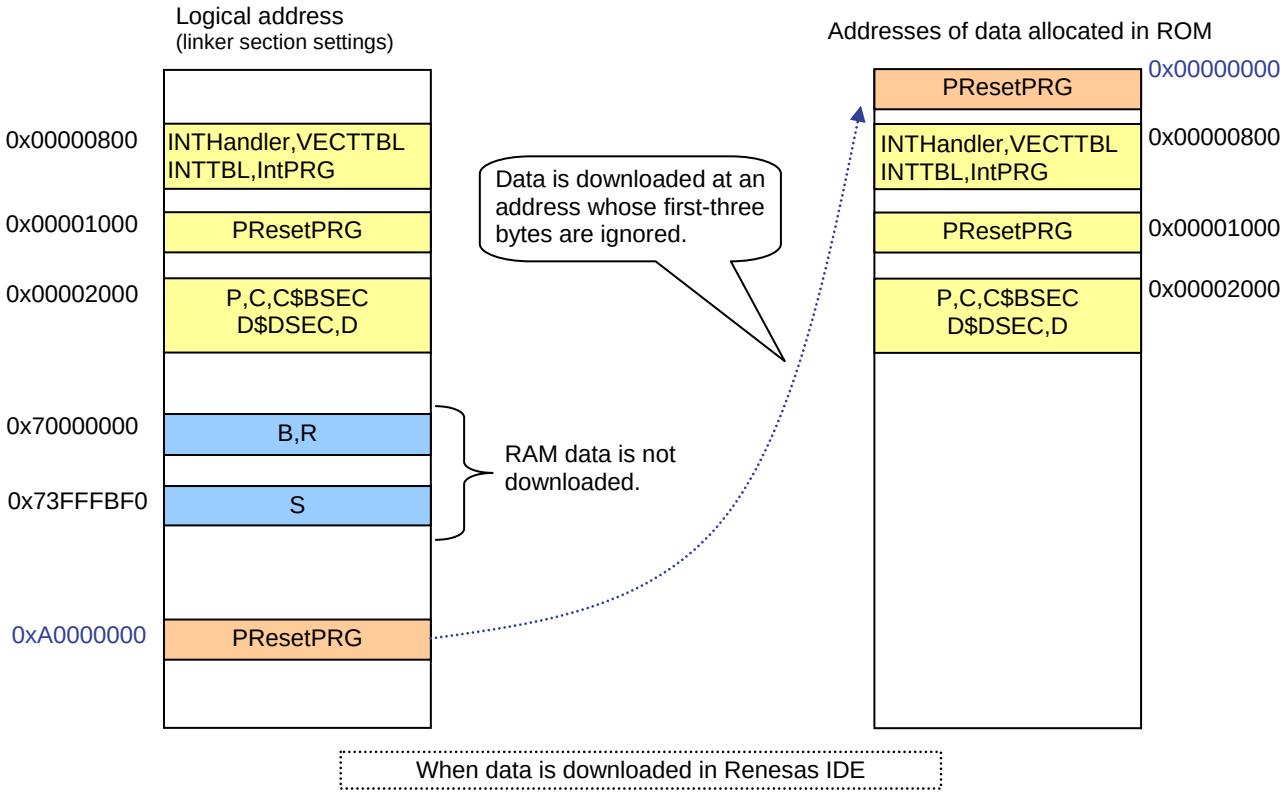


Figure 5-2

5.2 Procedure

Use the ROM support function of the optimizing linkage editor when, for example, variables having initial values are transferred from ROM to RAM. Employing this function allows a RAM address to be used to resolve the address of data allocated at a ROM address.

To generate a load module allocated in a physical address space, use the ROM support function and assume that the physical address space is ROM data and the logical address space is RAM. The following shows an example of specifying settings when the CPU is an SH7750.

(1) Section settings

Before the ROM support function settings can be specified, section settings must be specified.

First, change the address of RSThandler from an address in the logical address space to an address in the physical address space by replacing the first three bytes of the address with 0s. Next, secure a new section for address resolution at the logical address where RSThandler was allocated. In the following example, **V_RSTHandler** is secured as the new section for address resolution at address 0xA0000000.

Specify the following settings on the **Link/Library** page in the **SuperH RISC engine Standard Toolchain** dialog box:

Category: Select **Section**.
Show entries for: Select **Section**.
Modify: Click this button, and change the address of RSTHandler from 0xA0000000 to 0x00000000.
Add: Click this button, and secure **V_RSTHandler** at address 0xA0000000.

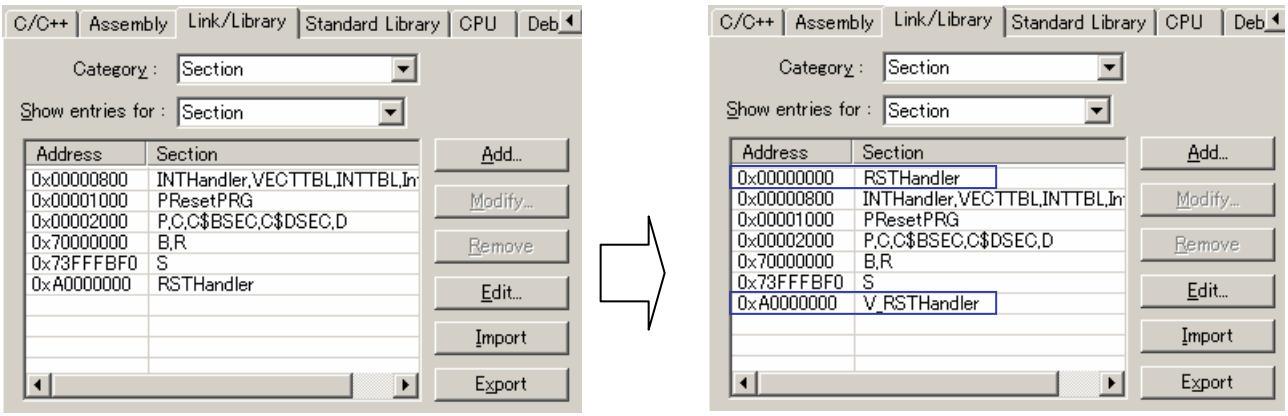


Figure 5-3

(2) ROM support function settings

Specify the following settings on the **Link/Library** page in the **SuperH RISC engine Standard Toolchain** dialog box:

Category: Select **Output**.
Show entries for: Select **ROM to RAM mapped sections**.
Add: Click this button to display the **Modify Rom to Ram** dialog box, and specify the following settings:
ROM section: Select **RSTHandler**.
RAM section: Select **V_RSTHandler**.

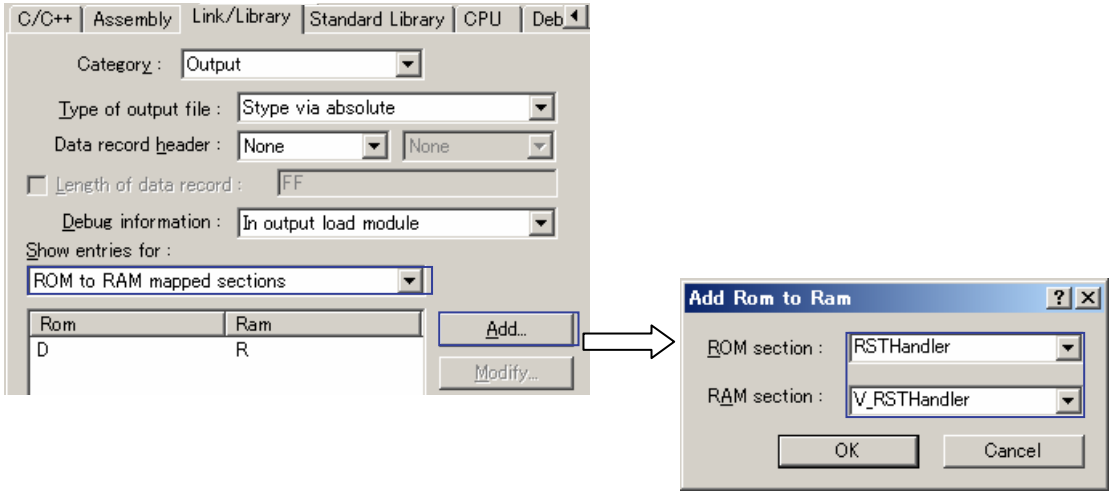


Figure 5-4

Website and Support <website and support,ws>

Renesas Technology Website

http://japan.renesas.com/

Inquiries

http://japan.renesas.com/inquiry

csc@renesas.com

Revision Record <revision history,rh>

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	April.01.08	--	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.