

CubeSuite+ V2.02.00

Integrated Development Environment

User's Manual: RX Build

Target Device

RX Family

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

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.

How to Use This Manual

This manual describes the role of the CubeSuite+ integrated development environment for developing applications and systems for RX families, and provides an outline of its features.

CubeSuite+ is an integrated development environment (IDE) for RX families, integrating the necessary tools for the development phase of software (e.g. design, implementation, and debugging) into a single platform.

By providing an integrated environment, it is possible to perform all development using just this product, without the need to use many different tools separately.

Readers This manual is intended for users who wish to understand the functions of the CubeSuite+ and design software and hardware application systems.

Purpose This manual is intended to give users an understanding of the functions of the CubeSuite+ to use for reference in developing the hardware or software of systems using these devices.

Organization This manual can be broadly divided into the following units.

[1.GENERAL](#)
[2.FUNCTIONS](#)
[3.BUILD OUTPUT LISTS](#)
[A.WINDOW REFERENCE](#)
[B.COMMAND REFERENCE](#)

How to Read This Manual It is assumed that the readers of this manual have general knowledge of electricity, logic circuits, and microcontrollers.

Conventions	Data significance:	Higher digits on the left and lower digits on the right
	Active low representation:	$\overline{\text{XXX}}$ (overscore over pin or signal name)
	Note:	Footnote for item marked with Note in the text
	Caution:	Information requiring particular attention
	Remark:	Supplementary information
	Numeric representation:	Decimal ... XXXX Hexadecimal ... 0xXXXX

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Document Name		Document No.
CubeSuite+ Integrated Development Environment User's Manual	Start	R20UT2865E
	V850 Design	R20UT2134E
	RL78 Design	R20UT2684E
	78K0R Design	R20UT2137E
	78K0 Design	R20UT2138E
	RX Coding	R20UT2999E
	V850 Coding	R20UT0553E
	Coding for CX Compiler	R20UT2659E
	RL78, 78K0R Coding	R20UT2774E
	78K0 Coding	R20UT2141E
	RX Build	This manual
	V850 Build	R20UT0557E
	Build for CX Compiler	R20UT2142E
	RL78, 78K0R Build	R20UT2623E
	78K0 Build	R20UT0783E
	RX Debug	R20UT2875E
	V850 Debug	R20UT2446E
	RL78 Debug	R20UT2867E
	78K0R Debug	R20UT0732E
	78K0 Debug	R20UT0731E
	Analysis	R20UT2868E
	Message	R20UT2871E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest edition of each document when designing.

All trademarks or registered trademarks in this document are the property of their respective owners.

TABLE OF CONTENTS

1.	GENERAL	8
1.1	Overview	8
1.2	Copyrights	10
2.	FUNCTIONS.	11
2.1	Overview	11
2.1.1	Create a load module	11
2.1.2	Create a user library	11
2.2	Change the Build Tool Version.	12
2.3	Set Build Target Files.	13
2.3.1	Add a file to a project	13
2.3.2	Remove a file from a project.	17
2.3.3	Remove a file from the build target.	17
2.3.4	Classify a file into a category	17
2.3.5	Change the file display order	18
2.3.6	Update file dependencies	19
2.4	Set the Type of the Output File	21
2.4.1	Change the output file name.	22
2.4.2	Output an assemble list	24
2.4.3	Output map information	25
2.4.4	Output library information	26
2.5	Set Compile Options	27
2.5.1	Perform optimization with the code size precedence	27
2.5.2	Perform optimization with the execution speed precedence	28
2.5.3	Add an include path	28
2.5.4	Set a macro definition.	29
2.6	Set Assemble Options	31
2.6.1	Add an include path	31
2.6.2	Set a macro definition.	33
2.7	Set Link Options.	33
2.7.1	Add a user library	34
2.7.2	Preparation for the use of the overlay-section selection facility	36
2.8	Set Librarian Options.	39
2.8.1	Set the output of a library file	40
2.9	Set Library Generate Options	40
2.9.1	Set the output of a standard library file.	41
2.10	Preparation before Using the PIC/PID Function	42
2.11	Set Build Options Separately.	42

2.11.1	Set build options at the project level	42
2.11.2	Set build options at the file level	42
2.12	Make Settings for Build Operations	47
2.12.1	Import the build options of other project	47
2.12.2	Set the link order of files	48
2.12.3	Change the file build order of subprojects	51
2.12.4	Display a list of build options	51
2.12.5	Change the file build target project	51
2.12.6	Add a build mode	52
2.12.7	Change the build mode.	54
2.12.8	Delete a build mode	54
2.12.9	Set the current build options as the standard for the project	55
2.13	Run a Build	56
2.13.1	Run a build of updated files	58
2.13.2	Run a build of all files	59
2.13.3	Run a build in parallel with other operations.	59
2.13.4	Run builds in batch with build modes	60
2.13.5	Compile/assemble individual files.	61
2.13.6	Stop running a build	62
2.13.7	Save the build results to a file.	62
2.13.8	Delete intermediate files and generated files	63
2.14	Estimate the Stack Capacity	65
3.	BUILD OUTPUT LISTS	66
3.1	Assemble List File	66
3.1.1	Source Information	66
3.1.2	Object Information	66
3.1.3	Statistics Information.	68
3.1.4	Compiler Command Specification Information	69
3.1.5	Assembler Command Specification Information.	69
3.2	Link Map File	69
3.2.1	Structure of Linkage List.	69
3.2.2	Option Information	70
3.2.3	Error Information.	70
3.2.4	Linkage Map Information	71
3.2.5	Symbol Information.	71
3.2.6	Symbol Deletion Optimization Information	72
3.2.7	Cross-Reference Information	73
3.2.8	Total Section Size	74
3.2.9	Vector Information.	74
3.2.10	CRC Information.	74
3.3	Library List	75

3.3.1	Structure of Library List.	75
3.3.2	Option Information	75
3.3.3	Error Information.	76
3.3.4	Library Information	76
3.3.5	Module, Section, and Symbol Information within Library	77
3.4	S-Type and HEX File Formats	78
3.4.1	S-Type File Format	78
3.4.2	HEX File Format	79
A.	WINDOW REFERENCE.	82
A.1	Description.	82
B.	COMMAND REFERENCE	337
B.1	RX Family C/C++ Compiler	337
B.1.1	Input/Output Files	337
B.1.2	Operating Instructions.	339
B.1.3	Options	343
B.1.3.1	Compile Options	343
B.1.3.2	Assembler Command Options.	470
B.1.3.3	Optimizing Linkage Editor (rlink) Options	508
B.1.3.4	Library Generator Options.	582
	Revision Record	593

1. GENERAL

This chapter introduces the processing of compiling performed by CC-RX, and provides an example of program development using CC-RX.

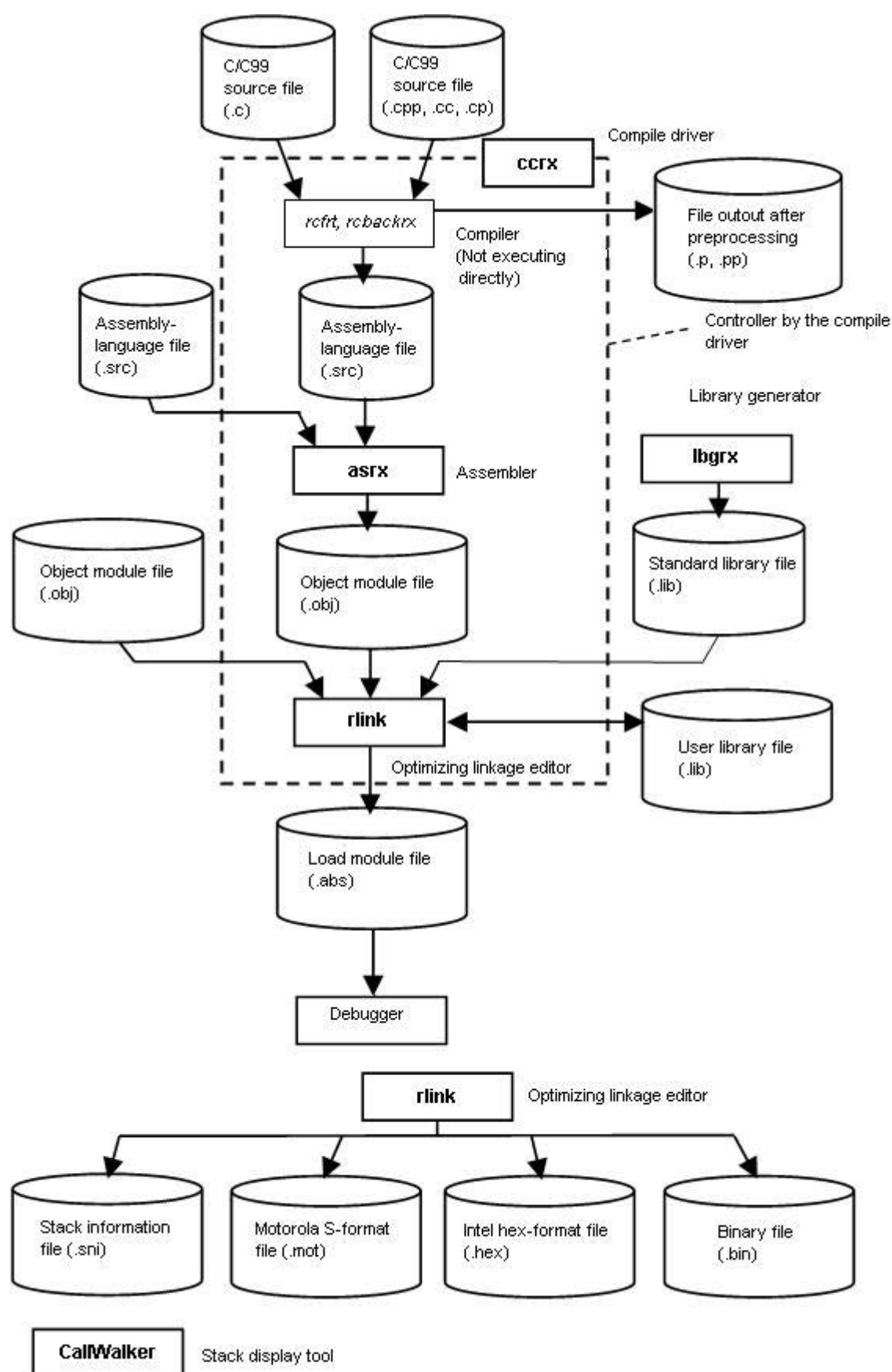
1.1 Overview

This section describes the components and processing flow of CC-RX. CC-RX is comprised of the 4 executable files listed below.

- (1) ccrx: Compile driver
- (2) asrx: Assembler
- (3) rlink: Optimizing linkage editor
- (4) lbgrx: Library generator

CC-RX Flow illustrates the CC-RX processing flow.

Figure 1.1 CC-RX Processing Flow



1.2 Copyrights

This LLVM-based software was developed in compliance with the LLVM Release License. Copyrights of other software components are owned by Renesas Electronics Corporation.

=====

LLVM Release License

=====

University of Illinois/NCSA
Open Source License

Copyright (c) 2003-2012 University of Illinois at Urbana-Champaign.
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

2. FUNCTIONS

This chapter describes the build procedure using CubeSuite+ and about the main build functions.

2.1 Overview

This section describes how to create a load module and user library.

2.1.1 Create a load module

The procedure for creating a load module is shown below.

- (1) Create or load a project
Create a new project, or load an existing one.

Remark See "CubeSuite+ Start" for details about creating a new project or loading an existing one.
- (2) Set a build target project
Set a build target project (see "[2.12 Make Settings for Build Operations](#)").
If there is no subproject, the project is always active.

Remarks 1. If there is no subproject in the project, the project is always active.
Remarks 2. When setting a build mode, add the build mode (see "[2.12.6 Add a build mode](#)").
- (3) Set build target files
Add or remove build target files and update the dependencies (see "[2.3 Set Build Target Files](#)").

Remarks 1. See "[2.7.1 Add a user library](#)" for the method of adding a user library to the project.
Remarks 2. Also, you can set the link order of object module files (see "[2.12.2 Set the link order of files](#)").
- (4) Specify the output of a load module
Select the type of the load module to be generated (see "[2.4 Set the Type of the Output File](#)").
- (5) Set build options
Set the options for the compiler, assembler, linker, library generator, and the like (see "[2.5 Set Compile Options](#)", "[2.6 Set Assemble Options](#)", "[2.7 Set Link Options](#)", "[2.9 Set Library Generate Options](#)").
- (6) Run a build
Run a build (see "[2.13 Run a Build](#)").
The following types of builds are available.
 - Build (see "[2.13.1 Run a build of updated files](#)")
 - Rebuild (see "[2.13.2 Run a build of all files](#)")
 - Rapid build (see "[2.13.3 Run a build in parallel with other operations](#)")
 - Batch build (see "[2.13.4 Run builds in batch with build modes](#)")
Remark If there are any commands you wish to run before or after the build process, on the [Property panel](#), from the [\[Common Options\]](#) tab, in the [Others] category, set the [Commands executed before build processing] and [Commands executed after build processing] properties.
If there are any commands you wish to run before or after the build process at the file level, you can set them from the [\[Individual Compile Options\(C\)\]](#) tab (for a C source file), [\[Individual Compile Options\(C++\)\]](#) tab (for a C++ source file), and [\[Individual Assemble Options\]](#) tab (for an assembler source file).
- (7) Save the project
Save the setting contents of the project to the project file.

Remark See "CubeSuite+ Start" for details about saving the project.

2.1.2 Create a user library

The procedure for creating a user library is shown below.

- (1) Create or load a project
Create a new project, or load an existing one.

When you create a new project, set a library project.

Remark See "CubeSuite+ Start" for details about creating a new project or loading an existing one.

(2) Set a build target project

Set a build target project (see "[2.12 Make Settings for Build Operations](#)").

If there is no subproject, the project is always active.

Remarks 1. If there is no subproject in the project, the project is always active.

Remarks 2. When setting a build mode, add the build mode (see "[2.12.6 Add a build mode](#)").

(3) Set build target files

Add or remove build target files and update the dependencies (see "[2.3 Set Build Target Files](#)").

(4) Set build options

Set the options for the compiler, assembler, library generator, and the like (see "[2.5 Set Compile Options](#)", "[2.6 Set Assemble Options](#)", "[2.8 Set Librarian Options](#)").

(5) Run a build

Run a build (see "[2.13 Run a Build](#)").

The following types of builds are available.

- Build (see "[2.13.1 Run a build of updated files](#)")
- Rebuild (see "[2.13.2 Run a build of all files](#)")
- Rapid build (see "[2.13.3 Run a build in parallel with other operations](#)")
- Batch build (see "[2.13.4 Run builds in batch with build modes](#)")

Remark If there are any commands you wish to run before or after the build process, on the [Property panel](#), from the [\[Common Options\] tab](#), in the [Others] category, set the [Commands executed before build processing] and [Commands executed after build processing] properties.

If there are any commands you wish to run before or after the build process at the file level, you can set them from the [\[Individual Compile Options\(C\)\] tab](#) (for a C source file), [\[Individual Compile Options\(C++\)\] tab](#) (for a C++ source file), and [\[Individual Assemble Options\] tab](#) (for an assembler source file).

(6) Save the project

Save the setting contents of the project to the project file.

Remark See "CubeSuite+ Start" for details about saving the project.

2.2 Change the Build Tool Version

You can change the version of the build tool (compiler package) used in the project (main project or subproject).

Select the build tool node on the project tree and select the [\[Common Options\] tab](#) on the [Property panel](#). Select [Always latest version which was installed] or the version in the [Using compiler package version] property in the [Version Select] category.

Figure 2.1 [Using compiler package version] Property



Remarks 1. When the build tool used in the main project and subprojects is the same, you can collectively change the build tool version by selecting all of the Build tool nodes and setting the property.

Remarks 2. If you have selected a compiler package that has not been installed (e.g. if you open a project created in another execution environment), then that version is also displayed.

Remarks 3. If the options change depending on the compiler package, then the display of the build tool's properties will change according to the selected version.

Properties that are hidden when the version is changed are saved in the project file's settings, and the values will be reproduced when the properties are displayed again.

Options are changed in accordance with the following rules. Information about changes is displayed in the [Output panel](#).

- If you change from an older version to a newer version, the option settings will be inherited and converted (only if necessary).
- If you change from a newer version to an older version, only identical option settings will be inherited. Options that only exist in the older version will be set to the default values.

2.3 Set Build Target Files

Before running a build, you must add the build target files (such as C source file, assembler source file) to the project. This section explains operations on setting files in the project.

2.3.1 Add a file to a project

Files can be added to a project by the following methods.

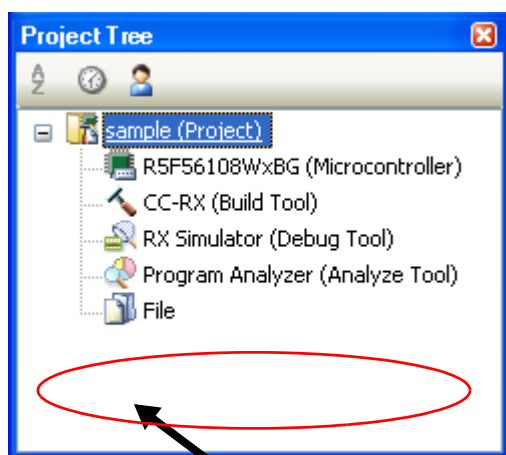
- [Adding an existing file](#)
- [Creating and adding an empty file](#)

(1) Adding an existing file

(a) Add individual files

Drag a folder from Explorer or the like, and drop it onto the empty space below the project tree. The file is added below the File node.

Figure 2.2 Project Tree Panel (File Drop Location)



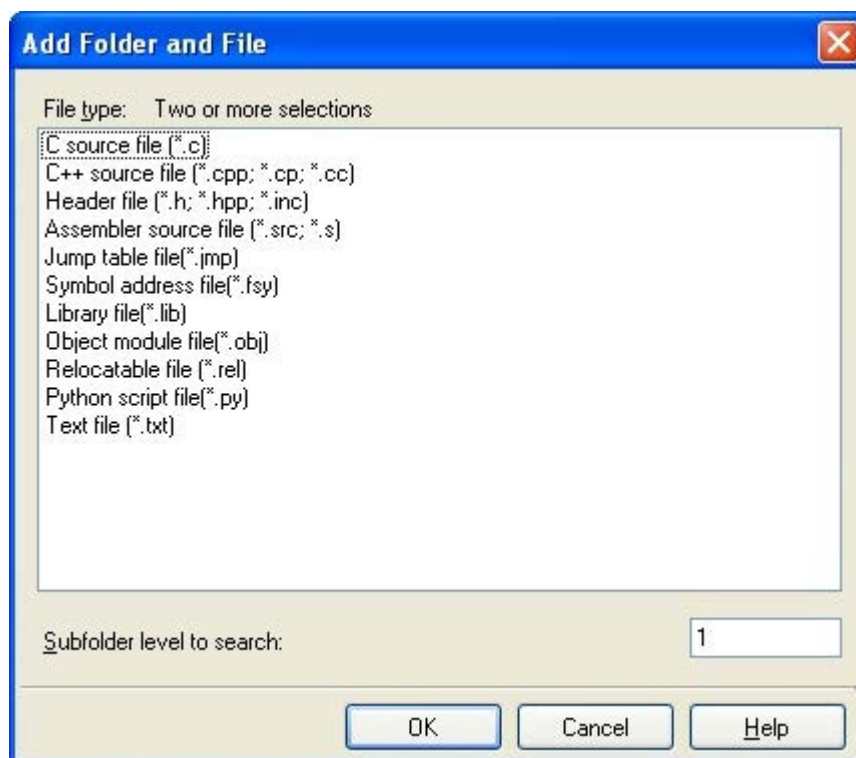
(b) Add a folder

Drag a folder from Explorer or the like, and drop it onto the empty space below the project tree. The [Add Folder and File dialog box](#) opens.

Remark You can also add multiple folders to the project at the same time by dragging multiple folders at same time and dropping them onto the project tree.

Caution When a folder with the name that is more than 200 characters is dropped, the folder is added to the project tree as a category with the name that 201st character and after are deleted.

Figure 2.3 Add Folder and File Dialog Box



In the dialog, select the file types to add to the project, specify the number of subfolder levels to add, and then click the [OK] button.

Remark You can select multiple file types by left clicking while holding down the [Ctrl] or [Shift] key. If nothing is selected, it is assumed that all types are selected.

The folder is added below the File node.

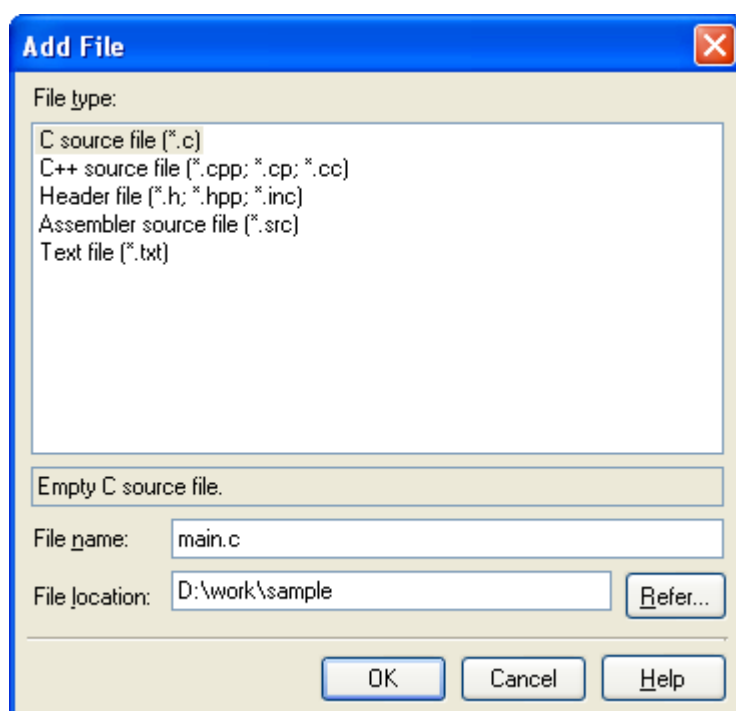
Note that on the project tree, the folder is the category.

Remark When the category node created by the user exists, you can add a file below the node by dropping the file onto the node (see "2.3.4 Classify a file into a category" for a category node).

(2) Creating and adding an empty file

On the project tree, select either one of the Project node, Subproject node, or File node, and then select [Add] >> [Add New File...] from the context menu. The [Add File dialog box](#) opens.

Figure 2.4 Add File Dialog Box



In the dialog box, specify the file to be created and then click the [OK] button.
The file is added below the File node.

The project tree after adding the file will look like the one below.

Figure 2.5 Project Tree Panel (After Adding File "main.c")

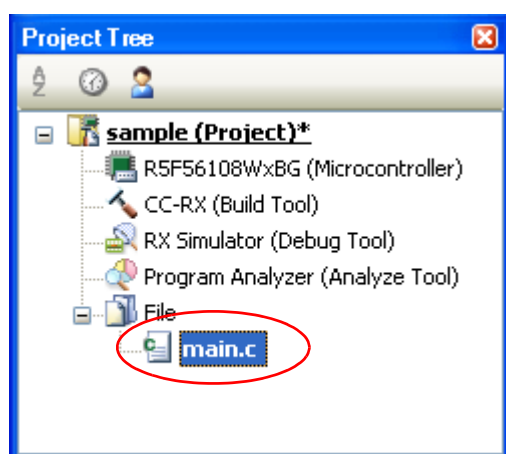
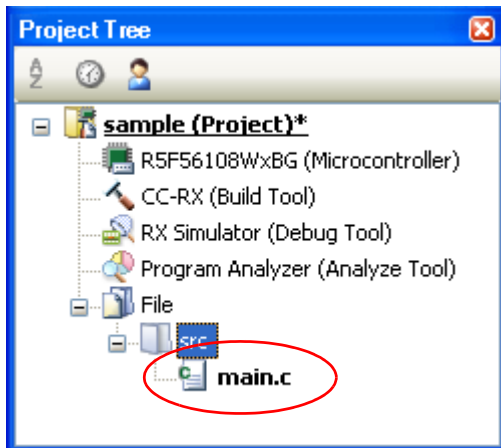


Figure 2.6 Project Tree Panel (After Adding Folder "src")



Remark The location of the file added below the File node depends on the current file display order setting. See ["2.3.5 Change the file display order"](#) for the method of changing the file display order.

Cautions 1. If the paths differ, you can add source files with the same name. Note, however, that if the setting of the output file name is left as the default, the output files will have the same name, which will prevent the build from running correctly (for example, when adding D:\sample1\func.c and D:\sample2\func.c, the default output file name for these files is both func.obj). To avoid this problems, set the output folder for each of those files to a different folder with the individual build options for the source files. Changing the output folder of the C source file is made with the [Path of the output folder] property in the [Object] category from the [\[Individual Compile Options\(C\)\]](#) tab. Changing the output folder of the C++ source file is made with the [Path of the output folder] property in the [Object] category from the [\[Individual Compile Options\(C++\)\]](#) tab. Changing the output folder of the assembler source file is made with the [Path of the output folder] property in the [Object] category from the [\[Individual Assemble Options\]](#) tab. See ["Set build options at the file level"](#) for how to set the individual build options.

Cautions 2. If source files with the same name are added, the target file cannot opened during debugging.

Cautions 3. Up to 5000 files can be added to the main project or subproject.

When a new file is added, an empty file is created in the location specified in the [Add File dialog box](#).

By double clicking the file name on the project tree, you can open the [Editor panel](#) and edit the file.

The files that can be opened with the [Editor panel](#) are shown below.

- C source file (*.c)
- C++ source file (*.cpp, *.cp, *.cc)
- Assembler source file (*.src)
- Header file (*.h, *.hpp, *.inc)
- Map file (*.map, *.lbp)
- Hex file (*.hex)
- S record file (*.mot)
- Text file (.txt)

Remarks 1. You can use one of the methods below to open files other than those listed above in the [Editor panel](#).

- Drag a file and drop it onto the [Editor panel](#).
- Select a file and then select [Open with Internal Editor...] from the context menu.

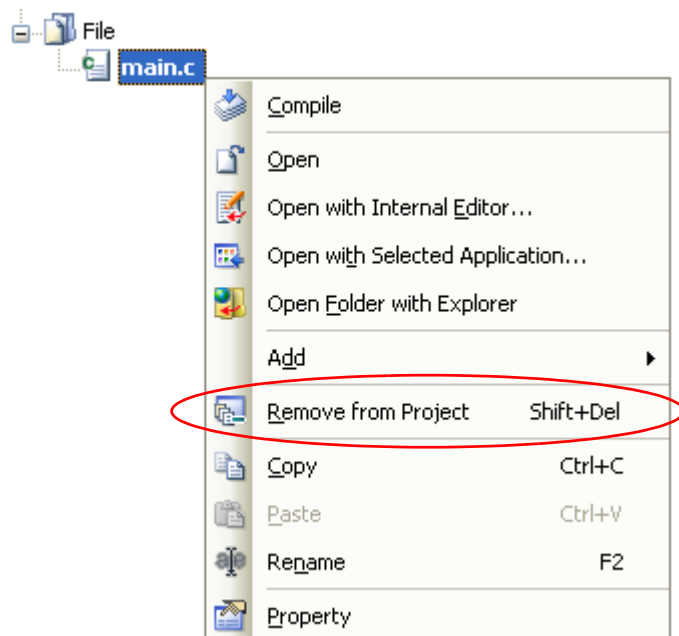
Remarks 2. When the environment is set to use an external editor on the [Option dialog box](#), the file is opened with the external editor that has been set. Other files are opened with the applications associated by the host OS.

2.3.2 Remove a file from a project

To remove a file added to a project, select the file to be removed from the project on the project tree and then select [Remove from Project] from the context menu.

In addition, the file itself is not deleted from the file system.

Figure 2.7 [Remove from Project] Item



2.3.3 Remove a file from the build target

You can remove a specific file from the build target out of all the files added to the project.

Select the file to be removed from the build target on the project tree and select the [Build Settings] tab on the Property panel. Select [No] on the [Set as build-target] property in the [Build] category.

Figure 2.8 [Set as build-target] Property



Remark The files that can be applied this function are C source files, C++ source files, assembler source files, object module files, and library files.

2.3.4 Classify a file into a category

You can create a category under the File node and classify files by the category. This makes it easier to view files added to the project on the project tree, and makes it easier to manage files according to function.

To create a category node, select either one of the Project node, Subproject node, or File node on the project tree, and then select [Add] >> [Add New Category] from the context menu.

Figure 2.9 [Add New Category] Item (For File Node)

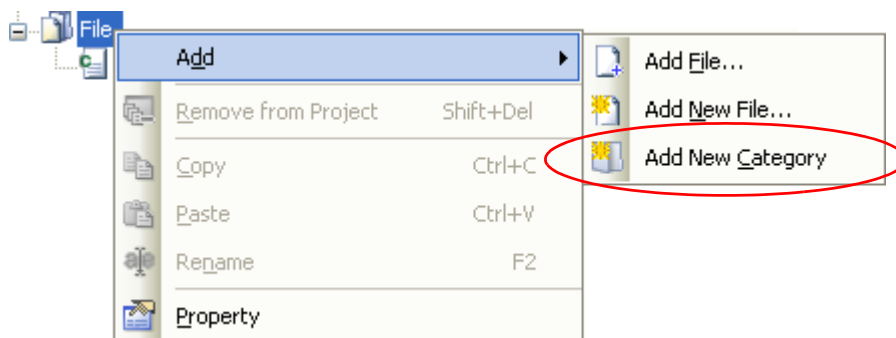
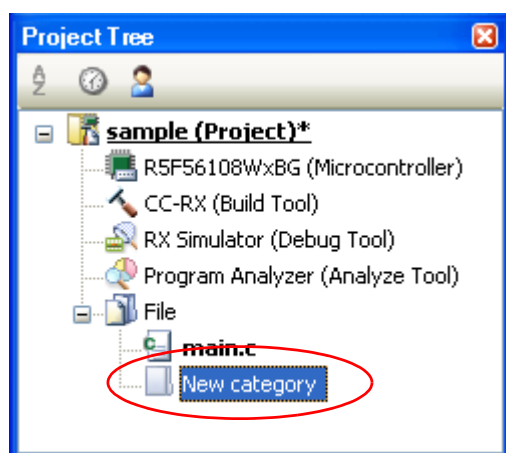


Figure 2.10 Project Tree Panel (After Adding Category Node)



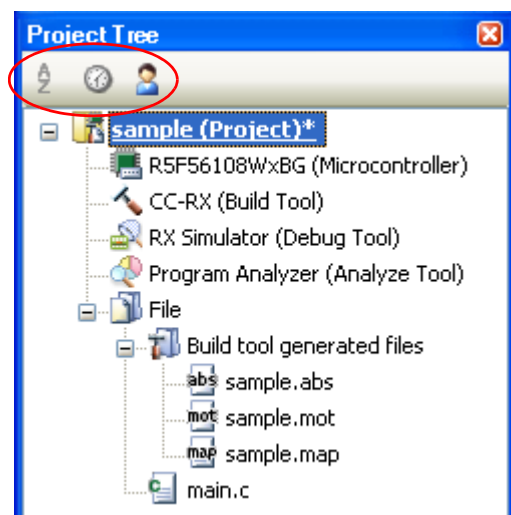
- Remarks 1. The default category name is "New category".
To change the category name, you can use [Rename] from the context menu of the category node.
- Remarks 2. You can also add a category node with the same name as an existing category node.
- Remarks 3. Categories can be nested up to 20 levels.

You can classify files into the created category node by dragging and dropping the file.










2.3.5 Change the file display order

You can change the display order of the files and category nodes using the buttons on the project tree.

Figure 2.11 Toolbar (Project Tree Panel)



Select any of the buttons below on the toolbar of the [Project Tree panel](#).

Button	Description
	Sorts category nodes and files by name.  : Ascending order  : Descending order  : Ascending order
	Sorts category nodes and files by timestamp.  : Descending order  : Ascending order  : Descending order
	Displays category nodes and files in the specified order by the user (default). You can change the display order of the category nodes and files arbitrarily by dragging and dropping them.

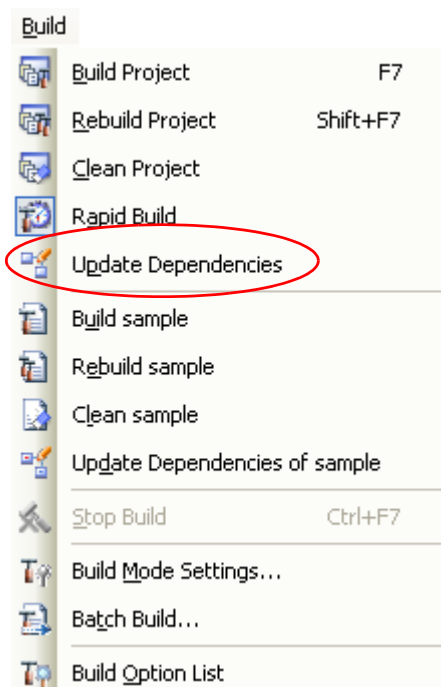
2.3.6 Update file dependencies

When you perform a change (changing include file paths, adding an include statement of the header file to the C source file, C++ source file, and assembler source file, etc.) that effects the file dependencies in the compile option settings or assemble option settings, you must update the dependencies of the relevant files.

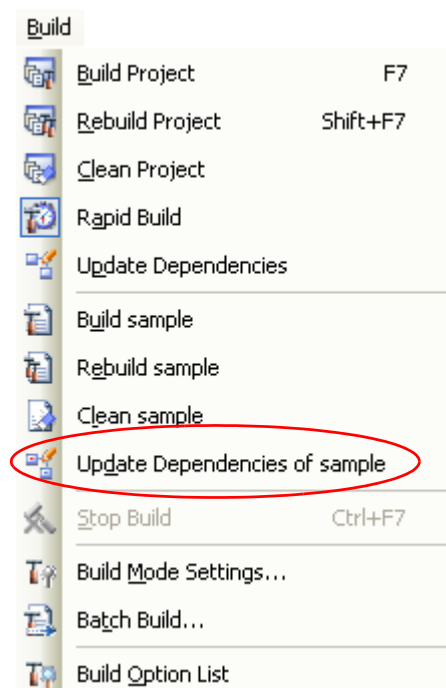
Updating file dependencies is performed for the entire project (main project and subprojects) or active project.

- (1) For the entire project
From the [Build] menu, select [Update Dependencies].

Figure 2.12 [Update Dependencies] Item



- (2) For the active project
From the [Build] menu, select [Update Dependencies of *active project*].

Figure 2.13 [Update Dependencies of *active project*] Item

Remark If there are files being edited with the [Editor panel](#) when updating file dependencies, then all these files are saved.

Cautions 1. During checking of dependence relationships of include files with CubeSuite+, condition statements such as `#if` and comments are ignored. Therefore, include files not required for build are mistaken as required files (In the example below, `header1.h` and `header5.h` are judged as required for build).

```
#if      0
#include  "header1.h"      /* Dependence relationship judged to exist */
#else
#include  "header2.h"      /* Dependence relationship to exist */
#endif

#define   AAA
#ifdef    AAA
#include  "header3.h"      /* Dependence relationship to exist */
#else
#include  "header4.h"      /* Dependence relationship to exist */
#endif

/*
#include  "header5.h"      /* Dependence relationship judged to exist */
*/
```

Cautions 2. During checking of dependence relationships of include files with CubeSuite+, include statements described after comments are ignored. Therefore, include files required for build are mistaken as no-required files (In the example below, `header6.h` and `header7.h` are judged as no-required for build).

```
/* Dependence relationship judged not to exist */
/* comment */  #include    "header6.h"

/* Dependence relationship judged not to exist */
/*
comment
*/  #include    "header7.h"
```

2.4 Set the Type of the Output File

Set the type of the file to be output as the product of the build.

- (1) For the application project
 - A load module file is generated.
 - The load module file will be the debug target.
 - Select the type of the convert file to be output as the product of the build other than the load module file.

Select the build tool node on the project tree and select the [\[Link Options\]](#) tab on the [Property panel](#). Select the file type in the [Type of output file] property in the [Output] category. Select the file type on the [Load module file convert format] property in the [Convert Load Module File] category.

Figure 2.14 [Convert Load Module File] Property

Convert Load Module File	
Load module file convert format	S record file (-FOrM=Stype)
Unifies the record size	No
Divides the conversion file	No
Path of the conversion file output folder	%BuildModeName%
Conversion file name	%ProjectName%.mot
Outputs the calculation result of CRC	No
Outputs the S9 record at the end	No

- (a) When [Hex file (-FOrM=Hexadecimal)] is selected
 - A hex file is outputted from the generated load module file.
 - (b) When [S record file (-FOrM=Stype)] is selected (default)
 - A Motorola S type file is outputted from the generated load module file.
 - (c) When [Binary file (-FOrM=Binary)] is selected
 - A binary file is outputted from the generated load module file.
- (2) For the library project
 - Select the build tool node on the project tree and select the [\[Librarian Options\]](#) tab on the [Property panel](#). Select the file type in the [Output file type] property in the [Output] category.

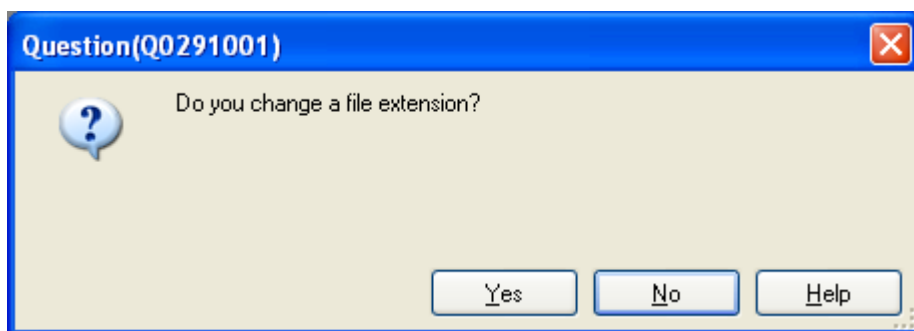
Figure 2.15 [Output file type] Property

Output	
Output file type	User library file (-FOrM=Library=U)
Path of the output folder	%BuildModeName%
Output file name	%ProjectName%.lib
Enables information-level message output	No (-NOMessage)
Suppresses the number of information-level messages	

- (a) When [User library file (-FOrM=Library=U)] is selected (default)
 - A user library file is generated.
 - (b) When [System library file (-FOrM=Library=S)] is selected
 - A system library file is generated.
 - (c) When [Relocatable module file (-FOrM=Relocate)] is selected
 - A relocatable module file is generated.

If the extension of output files is changed, the following message dialog box will open.

Figure 2.16 Message Dialog Box



Clicking [Yes] in the dialog box replaces the current file extension with the one for the output file type. Clicking [No], on the other hand, does not replace the current file extension.

2.4.1 Change the output file name

The names of the load module file, hex file, S record file, binary data file, relocatable module file, and library file output by the build tool are set to the following names by default.

Load module file name: %ProjectName%.abs
 Hex file name: %ProjectName%.hex
 S record file name: %ProjectName%.mot
 Binary data file name: %ProjectName%.bin
 Relocatable module file name: %ProjectName%.rel
 Library file name: %ProjectName%.lib

Remark "%ProjectName%" is a placeholder. It is replaced with the project name.

The method to change these file names is shown below.

- (1) When changing the load module file name
 Select the build tool node on the project tree and select the [\[Link Options\]](#) tab on the [Property panel](#). Enter the file name to be changed to on the [Output file name] property in the [Output] category.

Figure 2.17 [Output file name] Property (For Load Module File)

Output	
Output file type	Load module file (-FOrn=Absolute)
Outputs debugging information	Yes (Outputs to the output file) (-DEBog)
ROM to RAM mapped section	ROM to RAM mapped section[3]
Divides load module file	No
Path of the output folder	%BuildModeName%
Output file name	test.abs
Outputs the external symbol-allocation information file	No
Enables information-level message output	No (-NOMessage)
Suppresses the number of information-level messages	
Fills in padding data at the end of a section	No
Address setting for specified vector number	Address setting for specified vector number[0]
Address setting for unused vector area	
Outputs the jump table	No

This property supports the following placeholders.

%ActiveProjectName%: Replaces with the active project name.
 %MainProjectName%: Replaces with the main project name.
 %ProjectName%: Replaces with the project name.

Remark When [Yes] on the [Divides load module file] property is selected, the load module file is split into multiple files.

- (2) When changing the hex file name
 Select the build tool node on the project tree and select the [\[Link Options\]](#) tab on the [Property panel](#). Enter the file name to be changed to on the [Conversion file name] property in the [Convert Load Module File] category.

Figure 2.18 [Conversion file name] Property (For Hex File)

Convert Load Module File	
Load module file convert format	Hex file (-FOrM=Hexadecimal)
Unifies the record size	No
Divides the conversion file	No
Path of the conversion file output folder	%BuildModeName%
Conversion file name	test.hex
Specifies byte count for data record	No
Outputs the calculation result of CRC	No

This property supports the following placeholders.

%ActiveProjectName%: Replaces with the active project name.

%MainProjectName%: Replaces with the main project name.

%ProjectName%: Replaces with the project name.

Remark When [Yes] on the [Divides the conversion file] property is selected, the hex file is split into multiple files.

- (3) When changing the S record file name

Select the build tool node on the project tree and select the [\[Link Options\]](#) tab on the [Property panel](#). Enter the file name to be changed to on the [Conversion file name] property in the [Convert Load Module File] category.

Figure 2.19 [Conversion file name] Property (For S Record File)

Convert Load Module File	
Load module file convert format	S record file (-FOrM=Stype)
Unifies the record size	No
Divides the conversion file	No
Path of the conversion file output folder	%BuildModeName%
Conversion file name	test.mot
Outputs the calculation result of CRC	No
Outputs the S9 record at the end	No

This property supports the following placeholders.

%ActiveProjectName%: Replaces with the active project name.

%MainProjectName%: Replaces with the main project name.

%ProjectName%: Replaces with the project name.

Remark When [Yes] on the [Divides the conversion file] property is selected, the S record file is split into multiple files.

- (4) When changing the binary data file name

Select the build tool node on the project tree and select the [\[Link Options\]](#) tab on the [Property panel](#). Enter the file name to be changed to on the [Conversion file name] property in the [Convert Load Module File] category.

Figure 2.20 [Conversion file name] Property (For Binary Data File)

Convert Load Module File	
Load module file convert format	Binary data file (-FOrM=Binary)
Divides the conversion file	No
Path of the conversion file output folder	%BuildModeName%
Conversion file name	test.bin

This property supports the following placeholders.

%ActiveProjectName%: Replaces with the active project name.

%MainProjectName%: Replaces with the main project name.

%ProjectName%: Replaces with the project name.

Remark When [Yes] on the [Divides the conversion file] property is selected, the binary data file is split into multiple files.

- (5) When changing the user library file name

Select the build tool node on the project tree and select the [\[Librarian Options\]](#) tab on the [Property panel](#). Enter the file name to be changed to on the [Output file name] property in the [Output] category.

Figure 2.21 [Output file name] Property (For User Library File)

Output	
Output file type	User library file (-FOrM=Library=U)
Path of the output folder	%BuildModeName%
Output file name	test.lib
Enables information-level message output	No [-NUMessage]
Suppresses the number of information-level messages	

This property supports the following placeholders.

%ActiveProjectName%: Replaces with the active project name.

%MainProjectName%: Replaces with the main project name.

%ProjectName%: Replaces with the project name.

- (6) When changing the system library file name

Select the build tool node on the project tree and select the [\[Librarian Options\]](#) tab on the [Property panel](#). Enter the file name to be changed to on the [Output file name] property in the [Output] category.

Figure 2.22 [Output file name] Property (For System Library File)

Output	
Output file type	System library file (-FOrM=Library=S)
Path of the output folder	%BuildModeName%
Output file name	test.lib
Enables information-level message output	No [-NUMessage]
Suppresses the number of information-level messages	

This property supports the following placeholders.

%ActiveProjectName%: Replaces with the active project name.

%MainProjectName%: Replaces with the main project name.

%ProjectName%: Replaces with the project name.

- (7) When changing the relocatable module file name

Select the build tool node on the project tree and select the [\[Librarian Options\]](#) tab on the [Property panel](#). Enter the file name to be changed to on the [Output file name] property in the [Output] category.

Figure 2.23 [Output file name] Property (For Relocatable Module File)

Output	
Output file type	Relocatable module file (-FOrM=Relocate)
Outputs debugging information	Yes (Outputs to the output file) (-DEBug)
Path of the output folder	%BuildModeName%
Output file name	test.rel
Enables information-level message output	No [-NUMessage]
Suppresses the number of information-level messages	

This property supports the following placeholders.

%ActiveProjectName%: Replaces with the active project name.

%MainProjectName%: Replaces with the main project name.

%ProjectName%: Replaces with the project name.

2.4.2 Output an assemble list

The results of the assembly are output to the assemble list file.

- (1) For a C source file and C++ source file

Select the build tool node on the project tree and select the [\[Compile Options\]](#) tab on the [Property panel](#).

To output the assemble list, select [Yes (-listfile)] on the [Outputs a source list file] property in the [List] category.

Figure 2.24 [Outputs a source list file] Property

List	
Outputs a source list file	Yes(-listfile)
Outputs the C/C++ source file	No
Outputs the statements unsatisfied in conditional assembly	No
Outputs the information before .DEFINE replacement	No
Outputs the assembler macro expansion statements	No

- (2) For an assembler source file
 Select the build tool node on the project tree and select the [\[Assemble Options\]](#) tab on the [Property panel](#).
 To output the assemble list, select [Yes (-listfile)] on the [Outputs a assemble list file] property in the [List] category.

Figure 2.25 [Outputs a assemble list file] Property

List	
Outputs a assemble list file	Yes(-listfile)
Outputs the statements unsatisfied in conditional assembly	No
Outputs the information before .DEFINE replacement	No
Outputs the assembler macro expansion statements	No

Remark See "3.1 Assemble List File" for the assemble list.

2.4.3 Output map information

The map information (i.e. information on the result of linkage) is output to the linkage list file.

- (1) For the load module file
 Select the build tool node on the project tree and select the [\[Link Options\]](#) tab on the [Property panel](#).
 The setting to output the linkage list file is made with the [List] category.

Figure 2.26 [Outputs the linkage list file] Property

List	
Outputs the linkage list file	Yes (List contents=specify) (-LIST)
Outputs a symbol name list in a module	No
Outputs the number of symbol references	No
Outputs the cross-reference information	No
Shows the total sizes of sections	No
Outputs vector information	No

To output the linkage list file, select [Yes (List contents=specify) (-LIST)] in the [Outputs the linkage list file] property.
 When outputting the linkage list file, you can select the contents of the linkage list output by the linker.

- (a) When outputting a symbol name list in a module
 Select [Yes (-SHow=SYmbol)] in the [Outputs a symbol name list in a module] property.
 - (b) When outputting the number of symbol references
 Select [Yes (-SHow=Reference)] in the [Outputs the number of symbol references] property.
 - (c) When outputting the cross-reference information
 Select [Yes (-SHow=Xreference)] in the [Outputs the cross-reference information] property.
 - (d) When outputting the total sizes of sections
 Select [Yes (-SHow=Total_size)] in the [Shows the total sizes of sections] property.
 - (e) When outputting vector information
 Select [Yes (-SHow=VECTOR)] in the [Outputs vector information] property.
- (2) For the relocatable module file
 Select the build tool node on the project tree and select the [\[Librarian Options\]](#) tab on the [Property panel](#).
 The setting to output the linkage list file is made with the [List] category.

Figure 2.27 [Outputs the linkage list file] Property

List	
Outputs the linkage list file	Yes (List contents=specify) (-LIST)
Outputs a symbol name list in a module	No
Outputs the cross-reference information	No
Shows the total sizes of sections	No
Outputs vector information	No

To output the linkage list file, select [Yes (List contents=specify) (-LIST)] in the [Outputs the linkage list file] property. When outputting the linkage list file, you can select the contents of the linkage list output by the linker.

- (a) When outputting a symbol name list in a module
Select [Yes (-SHow=SYmbol)] in the [Outputs a symbol name list in a module] property.
- (b) When outputting the cross-reference information
Select [Yes (-SHow=Xreference)] in the [Outputs the cross-reference information] property.
- (c) When outputting the total sizes of sections
Select [Yes (-SHow=Total_size)] in the [Shows the total sizes of sections] property.

Remark See "3.2 Link Map File" for the linkage list file.

2.4.4 Output library information

The library information (i.e. information on the result of linkage) is output to the library list file. Select the build tool node on the project tree and select the [Librarian Options] tab on the Property panel. The setting to output a library list file is made with the [List] category.

Figure 2.28 [Outputs the linkage list file] Property

List	
Outputs the linkage list file	Yes (List contents=specify) (-LIST)
Outputs a symbol name list in a module	No
Outputs a section list in a module	No

To output the library list file, select [Yes (List contents=specify) (-LIST)] in the [Outputs the linkage list file] property. When outputting the library list file, you can select the contents of the library list output by the linker.

- (1) When outputting a symbol name list in a module
Select [Yes (-SHow=SYmbol)] in the [Outputs a symbol name list in a module] property.
- (2) When outputting a section list in a module
Select [Yes (-SHow=SEction)] in the [Outputs a section list in a module] property.

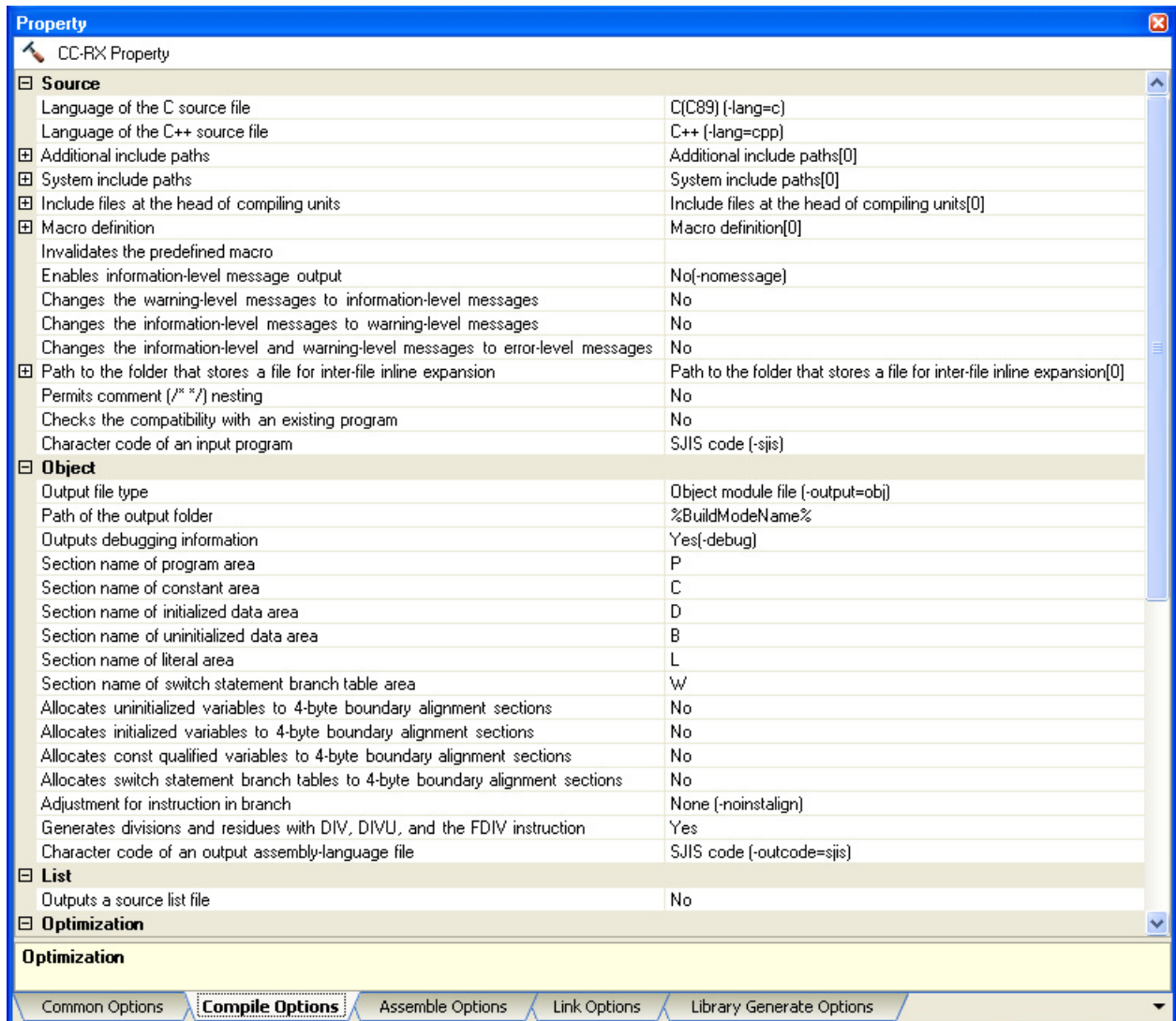
Remark See "3.3 Library List" for the library list file.

2.5 Set Compile Options

To set options for the compile phase, select the Build tool node on the project tree and select the [\[Compile Options\] tab](#) on the [Property panel](#).

You can set the various compile options by setting the necessary properties in this tab.

Figure 2.29 Property Panel: [Compile Options] Tab



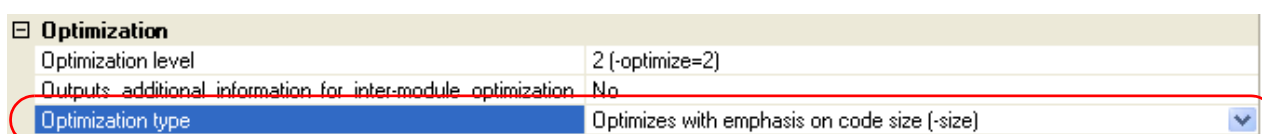
Remark Often used options have been gathered under the [Frequently Used Options(for Compile)] category on the [\[Common Options\] tab](#).

2.5.1 Perform optimization with the code size precedence

Select the build tool node on the project tree and select the [\[Compile Options\] tab](#) on the [Property panel](#).

To perform optimization with the code size precedence, select [Optimizes with emphasis on code size (-size)] on the [Optimization type] property in the [Optimization] category.

Figure 2.30 [Optimization type] Property (Code Size Precedence)



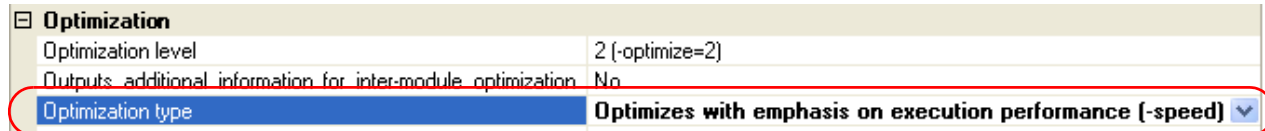
Remark You can also set the option in the same way with the [Optimization type] property in the [Frequently Used Options(for Compile)] category on the [\[Common Options\] tab](#).

2.5.2 Perform optimization with the execution speed precedence

Select the build tool node on the project tree and select the [\[Compile Options\]](#) tab on the [Property panel](#).

To perform optimization with the execution speed precedence, select [Optimizes with emphasis on execution performance (-speed)] on the [Optimization type] property in the [Optimization] category.

Figure 2.31 [Optimization type] Property (Execution Speed Precedence)



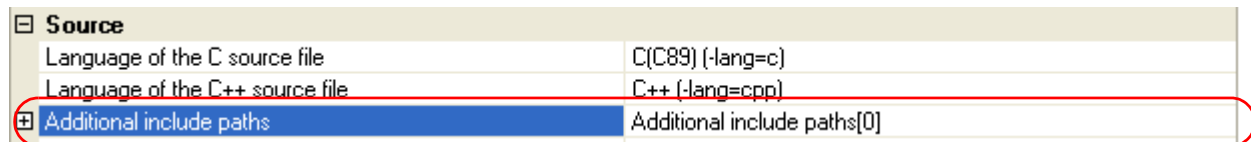
Remark You can also set the option in the same way with the [Optimization type] property in the [Frequently Used Options(for Compile)] category on the [\[Common Options\]](#) tab.

2.5.3 Add an include path

Select the build tool node on the project tree and select the [\[Compile Options\]](#) tab on the [Property panel](#).

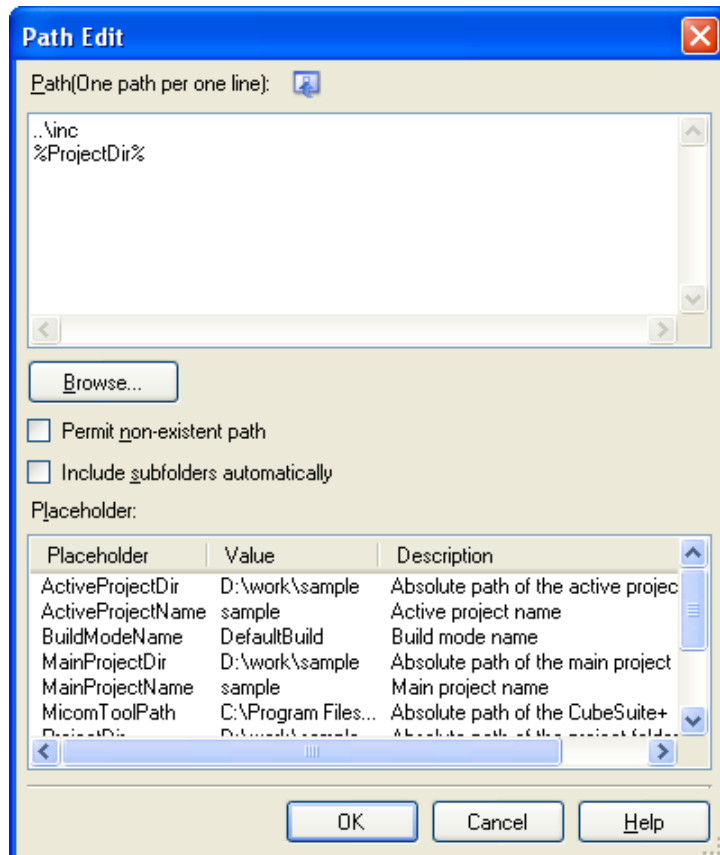
The include path setting is made with the [Additional include paths] property in the [Source] category.

Figure 2.32 [Additional include paths] Property



If you click the [...] button, the [Path Edit dialog box](#) will open.

Figure 2.33 Path Edit Dialog Box



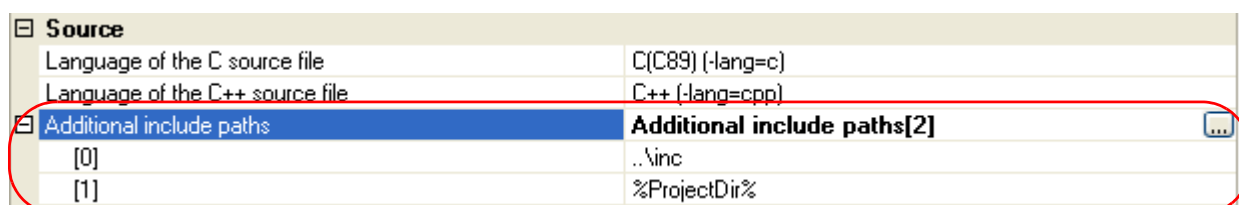
Enter an include path per line in [Path(One path per one line)].

You can specify up to 247 characters per line.

- Remarks 1. This property supports placeholders.
If a line is double clicked in [Placeholder], the placeholder will be reflected in [Path(One path per one line)].
- Remarks 2. You can also specify the include path by one of the following procedures.
- Drag and drop the folder using such as Explorer.
 - Click the [Browse...] button, and then select the folder in the [Browse For Folder dialog box](#).
 - Double click a row in [Placeholder].
- Remarks 3. Select the [Subfolders are automatically included] check box before clicking the [Browse...] button to add all paths under the specified one (down to 5 levels) to [Path(One path per one line)].

If you click the [OK] button, the entered include paths are displayed as subproperties.

Figure 2.34 [Additional include paths] Property (After Adding Include Paths)



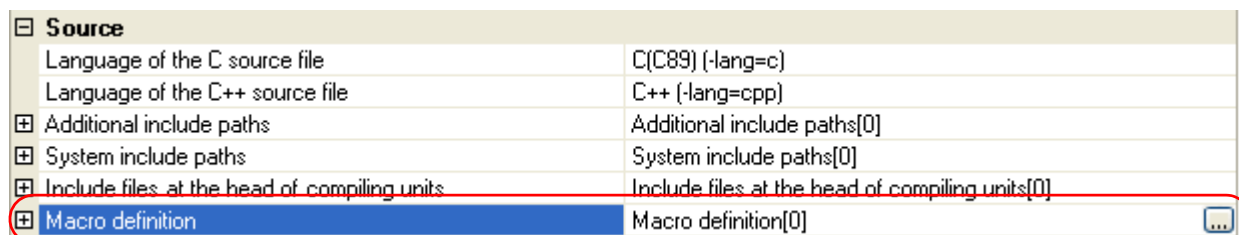
To change the include paths, you can use the [...] button or enter the path directly in the text box of the subproperty. When the include path is added to the project tree, the path is added to the top of the subproperties automatically.

Remark You can also set the option in the same way with the [Additional include paths] property in the [Frequently Used Options(for Compile)] category on the [\[Common Options\] tab](#).

2.5.4 Set a macro definition

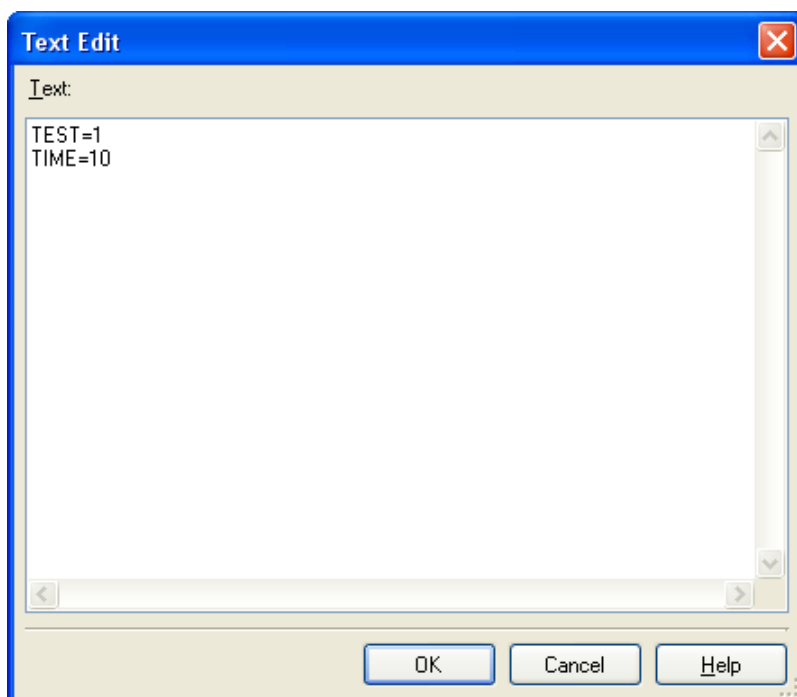
Select the build tool node on the project tree and select the [\[Compile Options\]](#) tab on the [Property panel](#). The macro definition setting is made with the [Macro definition] property in the [Source] category.

Figure 2.35 [Macro definition] Property



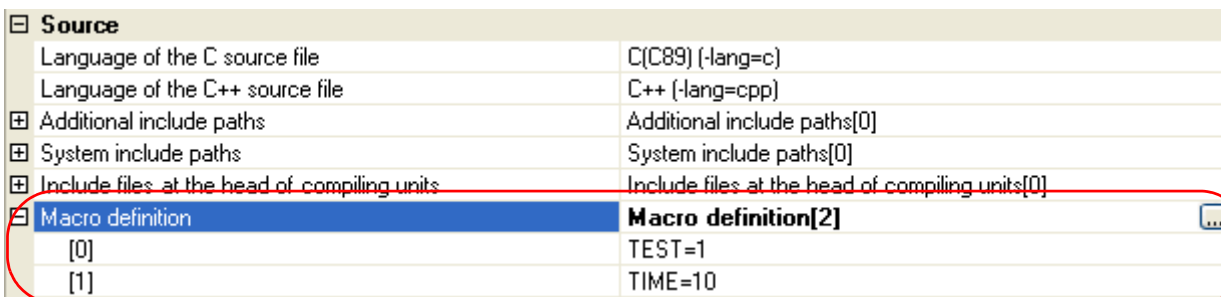
If you click the [...] button, the [Text Edit dialog box](#) will open.

Figure 2.36 Text Edit Dialog Box



Enter the macro definition in the format of "*macro name=string*", with one macro name per line. You can specify up to 256 characters per line, up to 65535 line. The "*=string*" part can be omitted, and in this case, the macro name is assumed to be defined. If you click the [OK] button, the entered macro definitions are displayed as subproperties.

Figure 2.37 [Macro definition] Property (After Setting Macros)



To change the macro definitions, you can use the [...] button or enter the path directly in the text box of the subproperty.

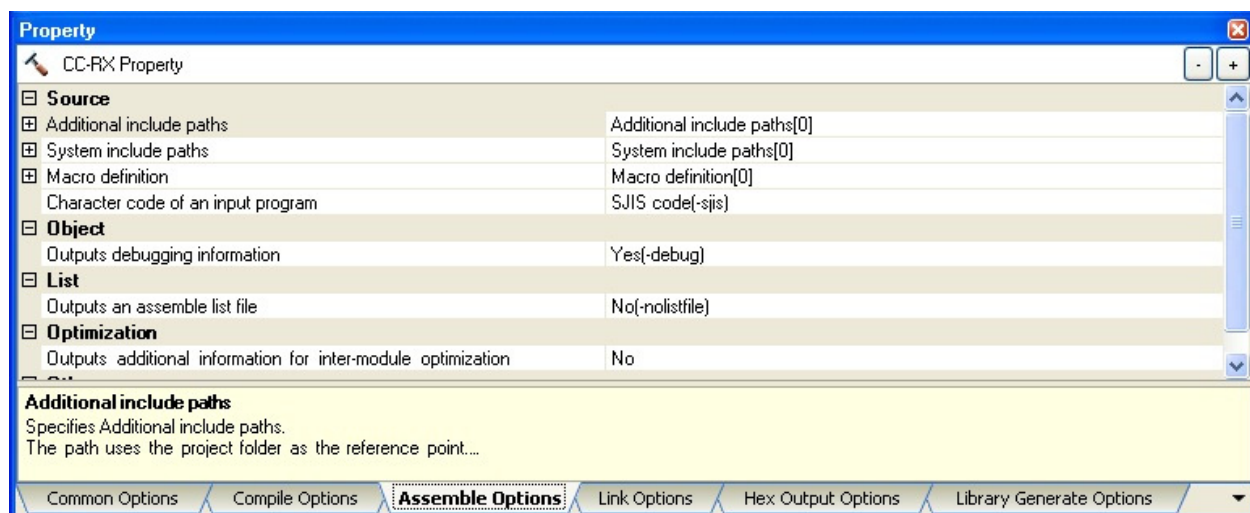
Remark You can also set the option in the same way with the [Macro definition] property in the [Frequently Used Options(for Compile)] category on the [\[Common Options\] tab](#).

2.6 Set Assemble Options

To set options for the assemble phase, select the Build tool node on the project tree and select the [\[Assemble Options\]](#) tab on the [Property panel](#).

You can set the various assemble options by setting the necessary properties in this tab.

Figure 2.38 Property Panel: [Assemble Options] Tab



Remark Often used options have been gathered under the [Frequently Used Options(for Assemble)] category on the [\[Common Options\]](#) tab.

2.6.1 Add an include path

Select the build tool node on the project tree and select the [\[Assemble Options\]](#) tab on the [Property panel](#).

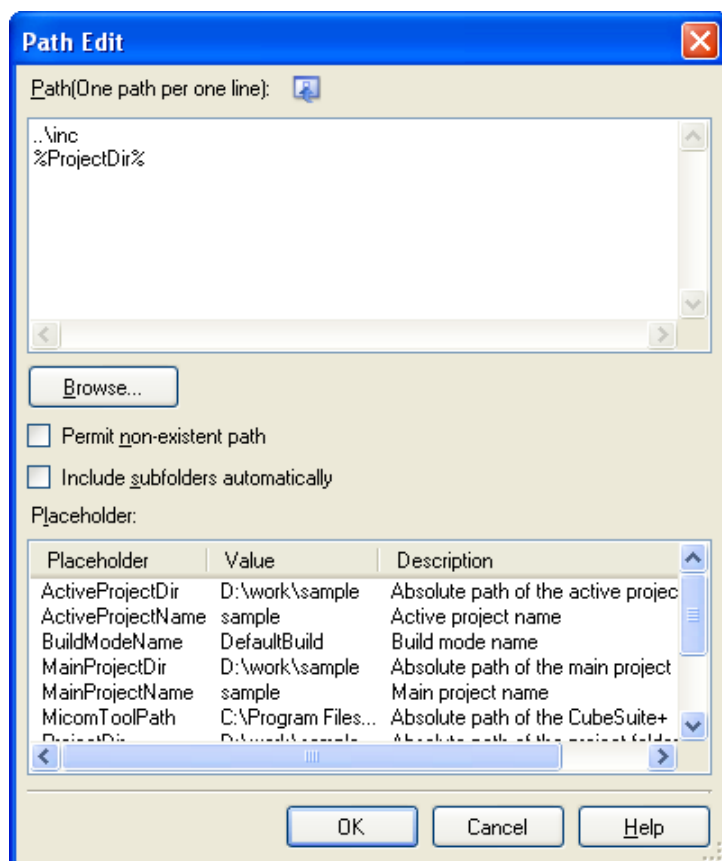
The include path setting is made with the [Additional include paths] property in the [Source] category.

Figure 2.39 [Additional include paths] Property



If you click the [...] button, the [Path Edit dialog box](#) will open.

Figure 2.40 Path Edit Dialog Box



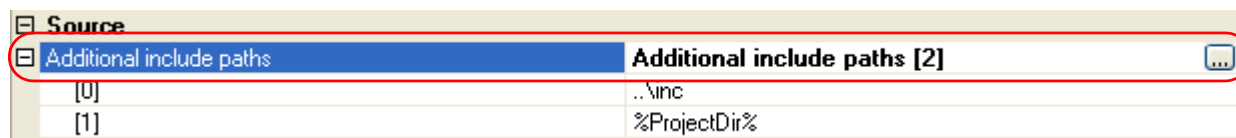
Enter the include path per line in [Path(One path per one line)].

You can specify up to 247 characters per line.

- Remarks 1. This property supports placeholders.
If a line is double clicked in [Placeholder], the placeholder will be reflected in [Path(One path per one line)].
- Remarks 2. You can also specify the include path by one of the following procedures.
- Drag and drop the folder using such as Explorer.
 - Click the [Browse...] button, and then select the folder in the [Browse For Folder dialog box](#).
 - Double click a row in [Placeholder].
- Remarks 3. Select the [Subfolders are automatically included] check box before clicking the [Browse...] button to add all paths under the specified one (down to 5 levels) to [Path(One path per one line)].

If you click the [OK] button, the entered include paths are displayed as subproperties.

Figure 2.41 [Additional include paths] Property (After Adding Include Paths)



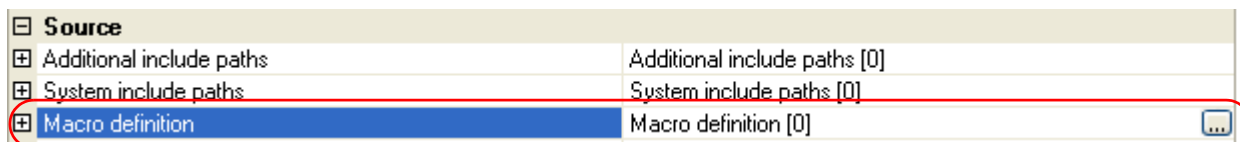
To change the include paths, you can use the [...] button or enter the path directly in the text box of the subproperty. When the include path is added to the project tree, the path is added to the top of the subproperties automatically.

Remark You can also set the option in the same way with the [Additional include paths] property in the [Frequently Used Options(for Assemble)] category on the [\[Common Options\] tab](#).

2.6.2 Set a macro definition

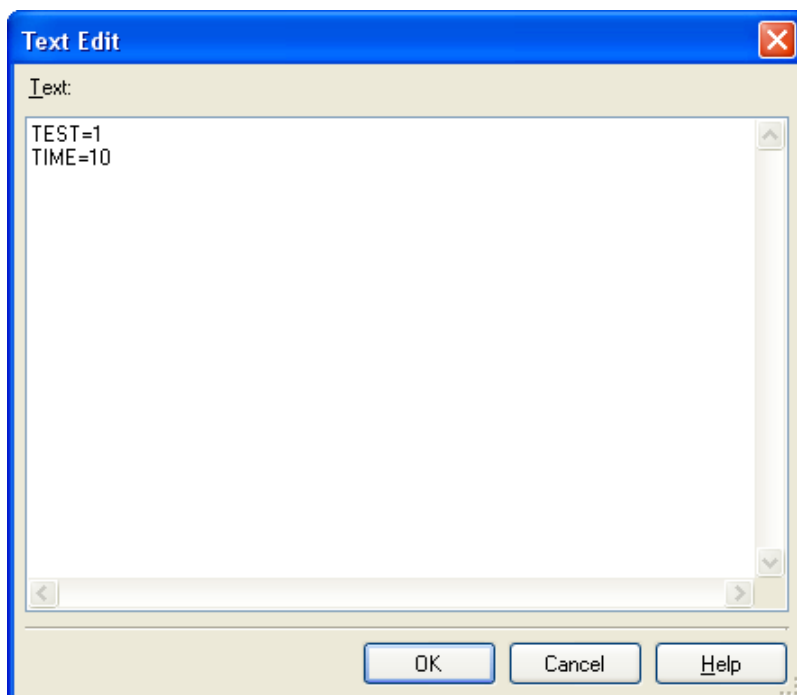
Select the build tool node on the project tree and select the [\[Assemble Options\]](#) tab on the [Property](#) panel. The macro definition setting is made with the [Macro definition] property in the [Source] category.

Figure 2.42 [Macro definition] Property



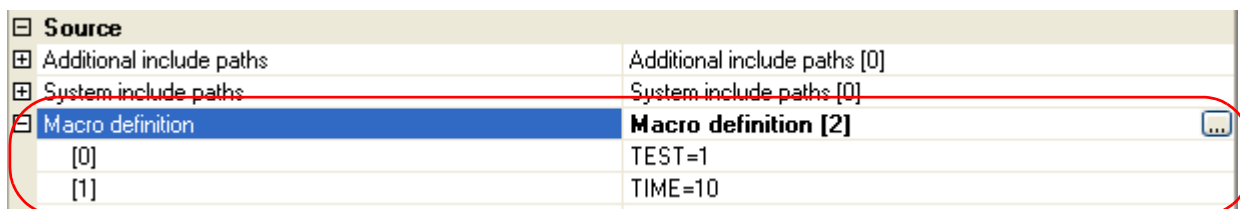
If you click the [...] button, the [Text Edit dialog box](#) will open.

Figure 2.43 Text Edit Dialog Box



Enter the macro definition in the format of "*macro name=string*", with one macro name per line. You can specify up to 256 characters per line, up to 65535 line. If you click the [OK] button, the entered macro definitions are displayed as subproperties.

Figure 2.44 [Macro definition] Property (After Setting Macros)



To change the macro definitions, you can use the [...] button or enter the path directly in the text box of the subproperty.

Remark You can also set the option in the same way with the [Macro definition] property in the [Frequently Used Options(for Assemble)] category on the [\[Common Options\]](#) tab.

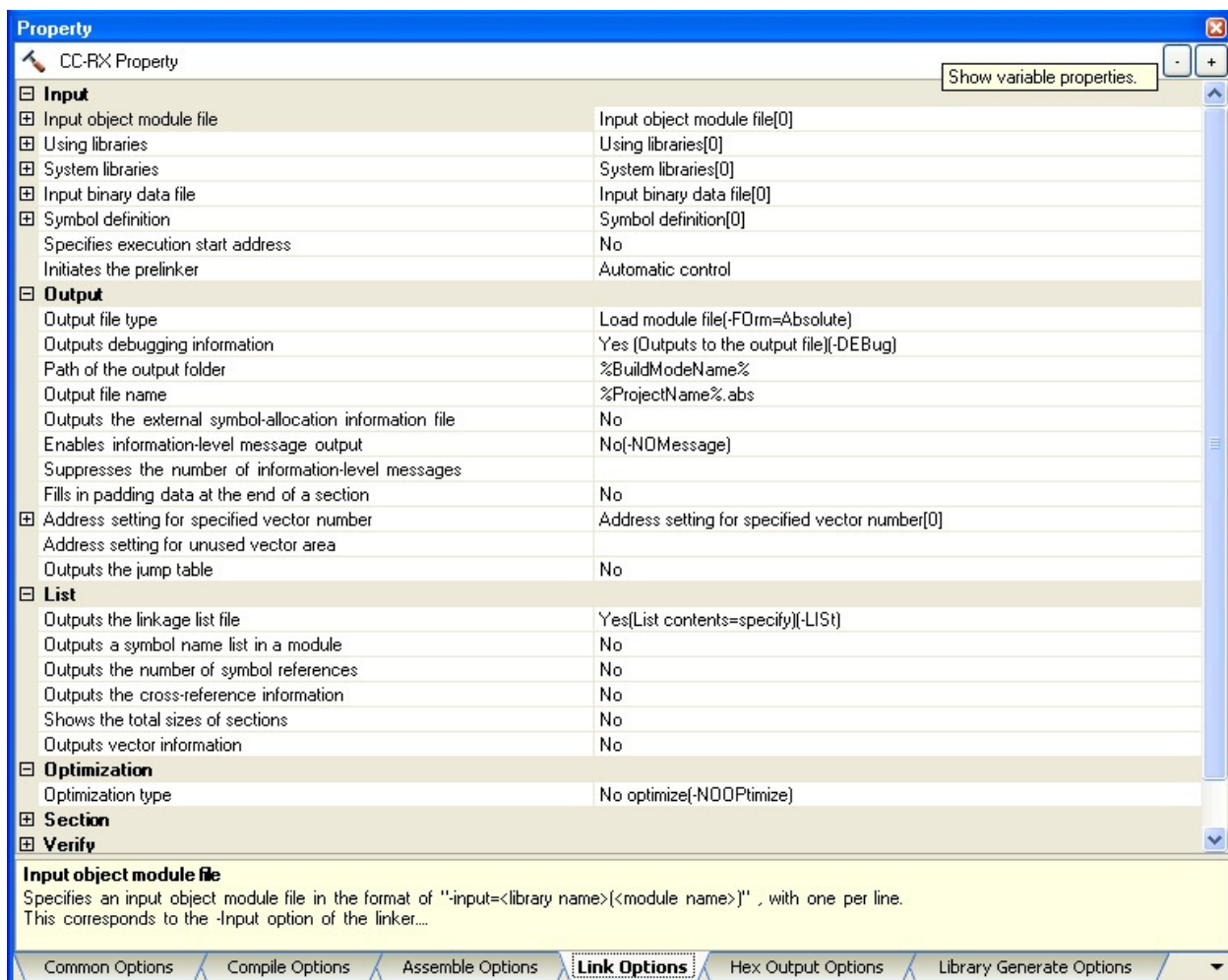
2.7 Set Link Options

To set options for the link phase, select the Build tool node on the project tree and select the [\[Link Options\]](#) tab on the [Property](#) panel.

You can set the various link options by setting the necessary properties in this tab.

Caution This tab is not displayed for the library project.

Figure 2.45 Property Panel: [Link Options] Tab

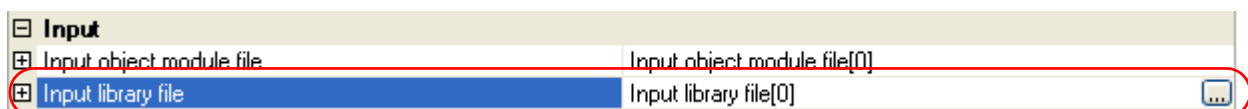


Remark Often used options have been gathered under the [Frequently Used Options(for Link)] category on the [Common Options] tab.

2.7.1 Add a user library

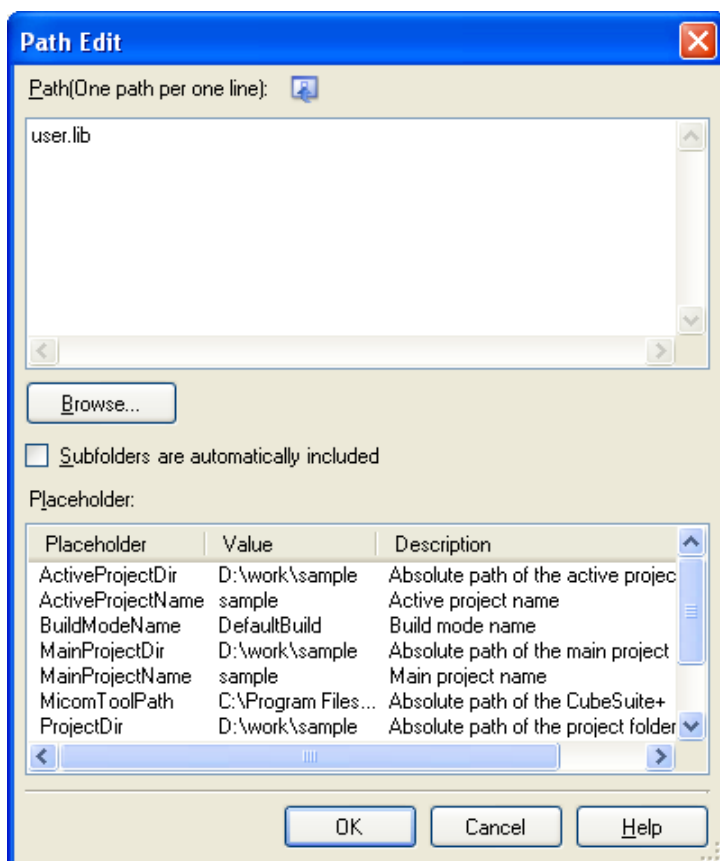
Select the build tool node on the project tree and select the [Link Options] tab on the Property panel. Adding a user library is made with the [Input library file] property in the [Input] category.

Figure 2.46 [Input library file] Property



If you click the [...] button, the Path Edit dialog box will open.

Figure 2.47 Path Edit Dialog Box



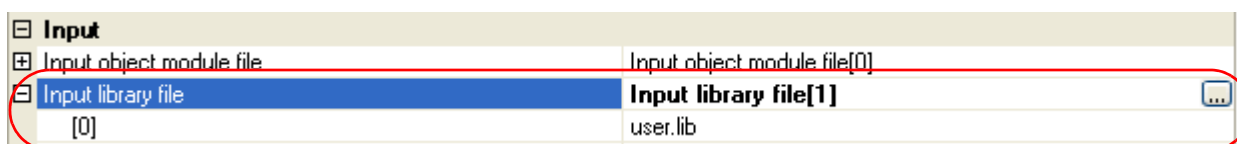
Enter an library file per line in [Path(One path per one line)].

You can specify up to 247 characters per line, up to 256 line.

- Remarks 1. This property supports placeholders.
If a line is double clicked in [Placeholder], the placeholder will be reflected in [Path(One path per one line)].
- Remarks 2. You can also specify the include path by one of the following procedures.
- Drag and drop the folder using such as Explorer.
 - Click the [Browse...] button, and then select the folder in the [Specify Using Library File dialog box](#).
 - Double click a row in [Placeholder].
- Remarks 3. Select the [Subfolders are automatically included] check box before clicking the [Browse...] button to add all paths under the specified one (down to 5 levels) to [Path(One path per one line)].

If you click the [OK] button, the entered library files are displayed as subproperties.

Figure 2.48 [Input library file] Property (After Setting Library Files)



To change the library files, you can use the [...] button or enter the path directly in the text box of the subproperty.

2.7.2 Preparation for the use of the overlay-section selection facility

The optimizing linkage editor (rlink) used by the CC-RX provides the start option, which allows two or more sections defined in a program to be allocated to a single address. Sections that have been allocated in this way are called overlay sections.

When the program is executed, only one of the sections allocated to the same address in the load module is executed.

The debug tool provides the "overlay-section selection facility" to select a particular overlay section to be debugged preferentially.

When overlay sections exist in a load module, the overlay-section selection facility allows you to choose the priority section to be debugged before the load module is executed.

To use this facility, the load module must be created in the following way.

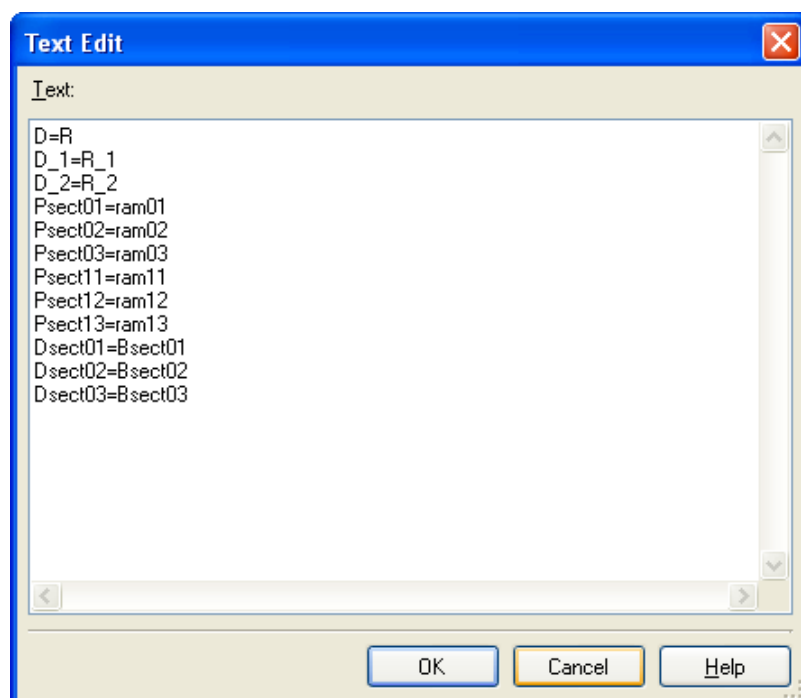
- (1) Copy the contents of the ROM into the RAM
Within the program, copy the contents of the ROM into the RAM. Code and data will be allocated to the RAM.
- (2) Set build options
In order to support the overlay-section selection facility, set up sections and overlay sections to be mapped from the ROM to the RAM. Select the build tool node on the project tree and select on the [\[Link Options\] tab](#) in the [Property panel](#).
 - (a) Set up sections to be mapped from the ROM to the RAM
This setting can be made through the [ROM to RAM mapped section] property under the [Output] category. Reserve a RAM section as large as the ROM section and relocate the symbols defined in the ROM to the corresponding addresses in the RAM section.

Figure 2.49 [ROM to RAM mapped section] Property

Output	
Output file type	Load module file (-FOrn=Absolute)
Outputs debugging information	Yes (Outputs to the output file) (-DEBug)
ROM to RAM mapped section	ROM to RAM mapped section[3]
[0]	D=R
[1]	D_1=R_1
[2]	D_2=R_2

Clicking on the [...] button opens the [Text Edit dialog box](#).

Figure 2.50 Text Edit Dialog Box



In the Text field, enter pairs of section names in the form of "ROM section name=RAM section name". Write one pair on each line. Up to 30 lines can be listed, each consisting of up to 256 characters. Then click on the [OK] button. The entered section names will be displayed as sub-properties.

Figure 2.51 [ROM to RAM mapped section] Property (after Setting up Sections to be Mapped from the ROM to the RAM)

Output	
Output file type	Load module file (-F0rm=Absolute)
Outputs debugging information	Yes (Outputs to the output file) (-DEBog)
ROM to RAM mapped section	ROM to RAM mapped section[12]
[00]	D=R
[01]	D_1=R_1
[02]	D_2=R_2
[03]	Psect01=ram01
[04]	Psect02=ram02
[05]	Psect03=ram03
[06]	Psect11=ram11
[07]	Psect12=ram12
[08]	Psect13=ram13
[09]	Dsect01=Bsect01
[10]	Dsect02=Bsect02
[11]	Dsect03=Bsect03

You can also change the name of each section by clicking on the [...] button or directly entering a new name in the text box for the sub-property.

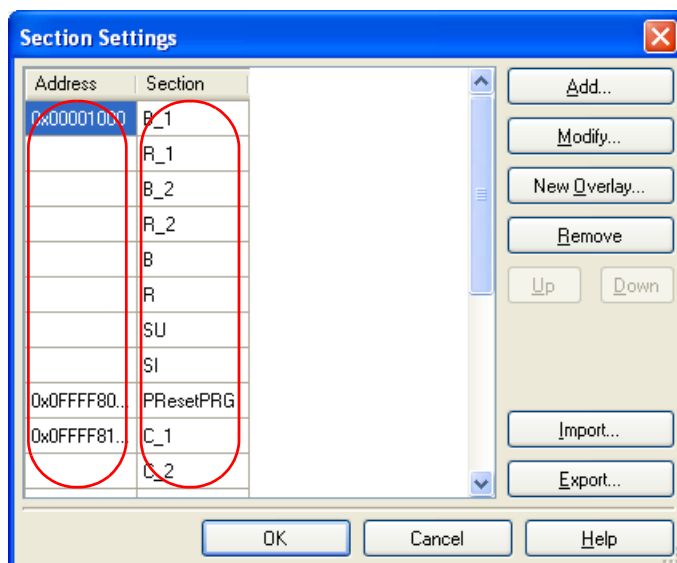
- (b) Set up ROM and RAM sections (overlay sections)
Sections can be set up through the [Section start address] property under the [Section] category.

Figure 2.52 [Section start address] Property

Section	
Section start address	B_1,R_1,B_2,R_2,B,R,SU,SI/01000,PRresetPRG/0FFFF8000,C_1,C_2,C,C\$,D_1,D_2,D,P,Pln

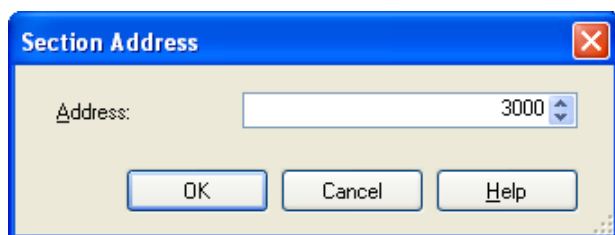
Clicking on the [...] button opens the [Section Settings dialog box](#). You can change the size of this dialog box by dragging the lower right corner.

Figure 2.53 Section Settings Dialog Box



- <1> To add the address of a section
Place the cursor over the [Section] area in the [Section Settings dialog box](#) and click on the [Add...] button. The [Section Address dialog box](#) will appear.

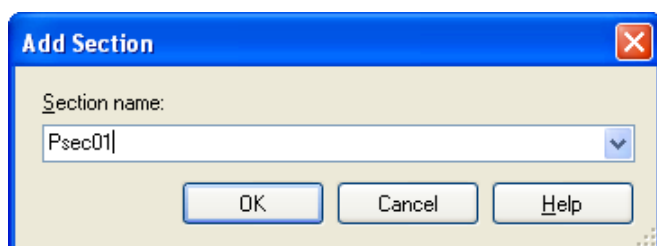
Figure 2.54 Section Address Dialog Box



Enter the address of the section that you wish to add in the [Address] field and click on the [OK] button. The added address will be displayed in the [Address] area of the [Section Settings dialog box](#).

- <2> To add a ROM section
Place the cursor over the [Address] area in the [Section Settings dialog box](#) and click on the [Add...] button. The [Add Section] dialog box will appear.

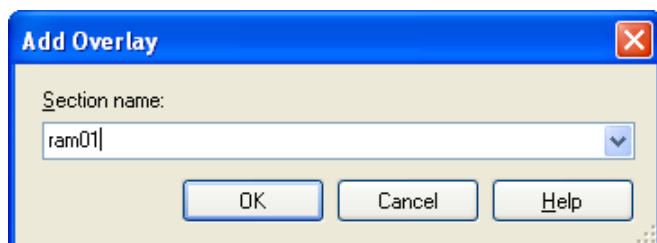
Figure 2.55 Add Section Dialog Box



Enter the name of the section that you wish to add in the [Section name] field and click on the [OK] button. The added section will be displayed in the [Section] area of the [Section Settings dialog box](#).

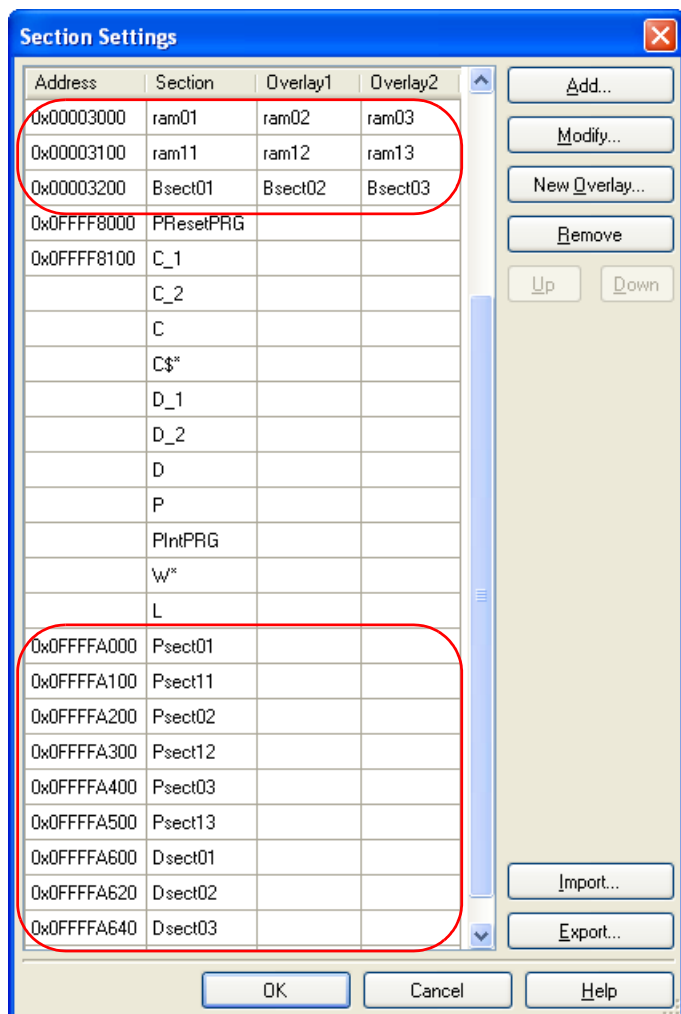
- <3> To add RAM sections (overlay sections)
Place the cursor over the [Address] area in the [Section Settings dialog box](#) and click on the [New Overlay...] button. The [Add Overlay] dialog box will appear.

Figure 2.56 Add Overlay Dialog Box



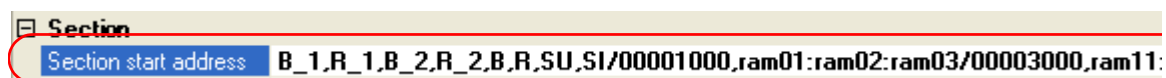
Enter the name of the section that you wish to add in the [Section name] field and click on the [OK] button. The added section will be displayed in the [Section] area of the [Section Settings dialog box](#). Repeat this procedure for the number of overlay sections that you wish to add. The added sections will be displayed in the [Overlay *n*] area (*n*: Number, beginning with 1).

Figure 2.57 After Setting up ROM and RAM Sections ([Section Settings] Dialog Box)



When the [OK] button is clicked on, the ROM and RAM sections (overlay sections) are displayed in the text box.

Figure 2.58 [Section start address] Property (after Sections Have been Set up)



- (3) Build the project
Start building of the project.
A load module that supports the overlay-section selection facility will be generated.

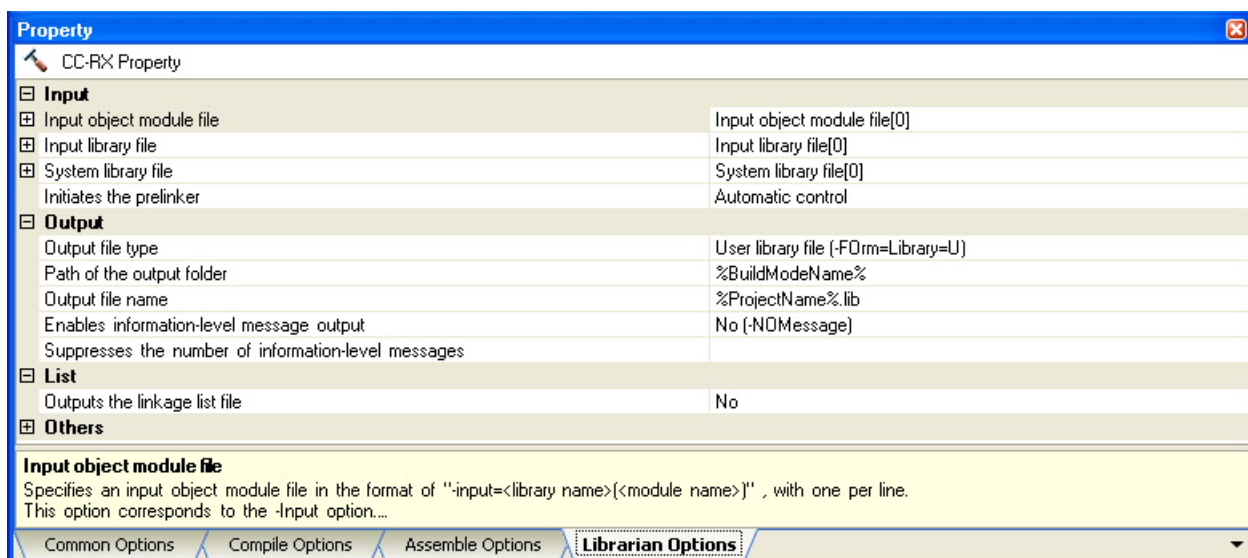
2.8 Set Librarian Options

To set options for the link phase, select the Build tool node on the project tree and select the [\[Librarian Options\] tab](#) on the [Property panel](#).

You can set the various librarian options by setting the necessary properties in this tab.

Caution This tab is not displayed for the application project.

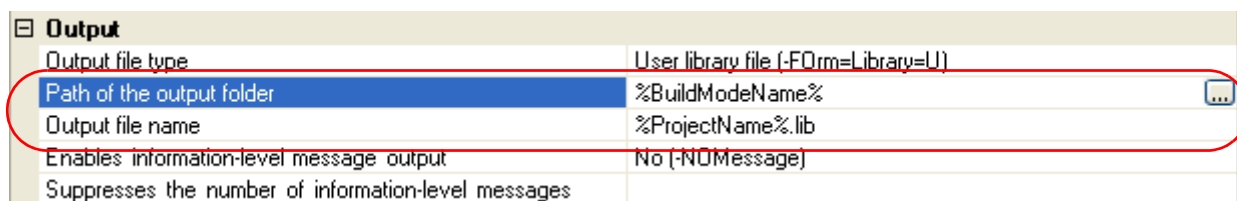
Figure 2.59 Property Panel: [Librarian Options] Tab



2.8.1 Set the output of a library file

Select the build tool node on the project tree and select the [\[Librarian Options\]](#) tab on the [Property panel](#). The setting to output a library file is made with the [Output] category.

Figure 2.60 [Output] Category



- (1) Set the output folder

Setting the output folder is made with the [Path of the output folder] property by directly entering to the text box or by the [...] button.

Up to 247 characters can be specified in the text box.

This property supports the following placeholder.

 - %BuildModeName%: Replaces with the build mode name.

"%BuildModeName%" is set by default.
- (2) Set the output file name

Setting the output file is made with the [Output file name] property by directly entering to the text box.

Up to 259 characters can be specified in the text box.

This property supports the following placeholders.

 - %ActiveProjectName%: Replaces with the active project name.
 - %MainProjectName%: Replaces with the main project name.
 - %ProjectName%: Replaces with the project name.

"%ProjectName%.lib" is set by default.

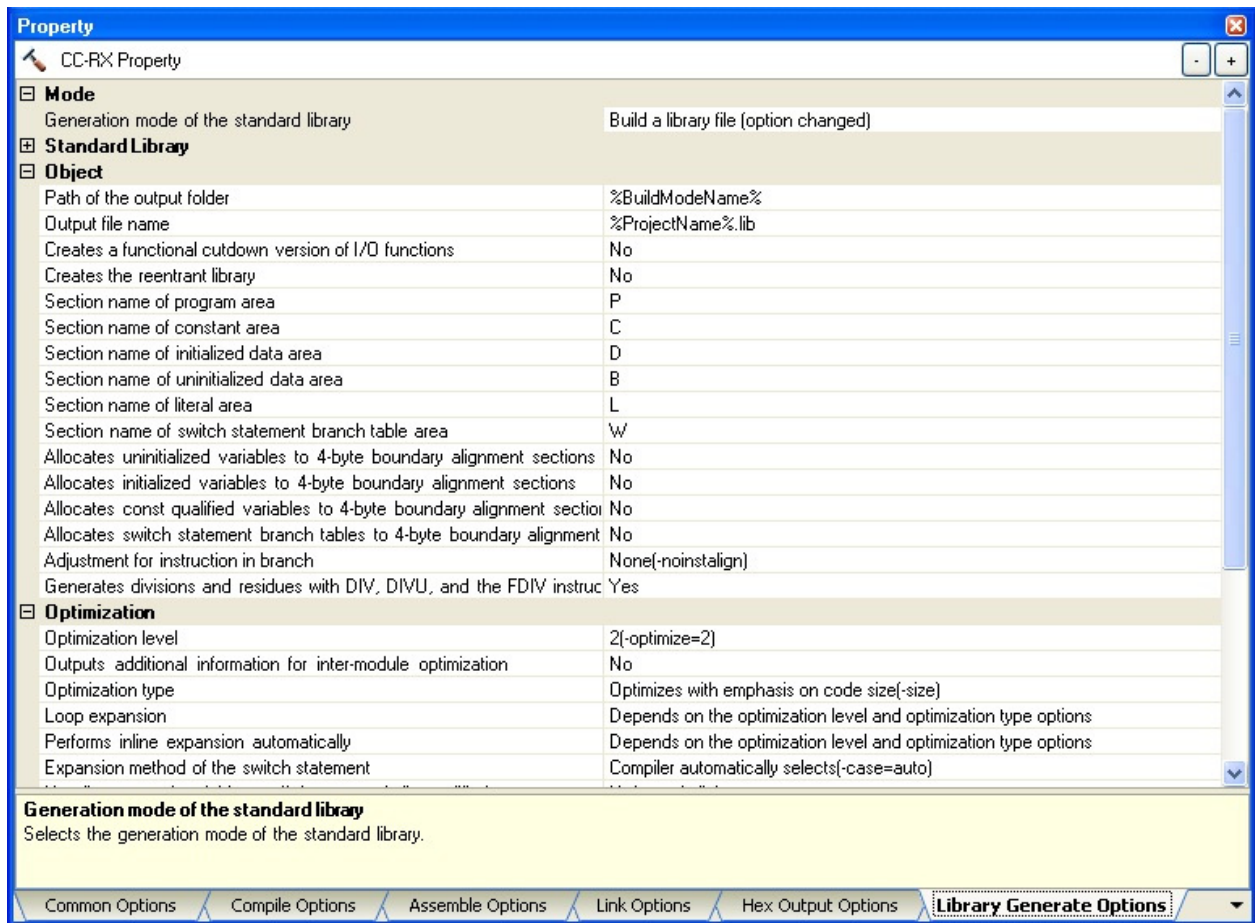
2.9 Set Library Generate Options

To set options for the library generator, select the Build tool node on the project tree and select the [\[Library Generate Options\]](#) tab on the [Property panel](#).

You can set the various create library options by setting the necessary properties in this tab.

Caution This tab is not displayed for the library project.

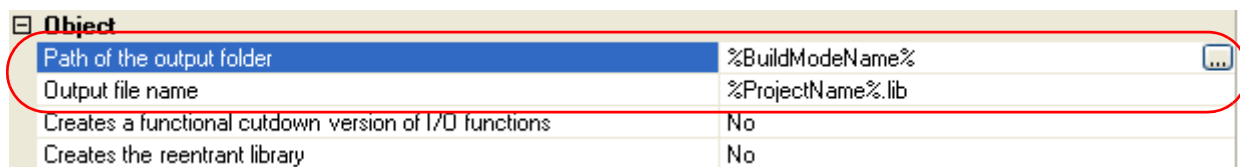
Figure 2.61 Property Panel: [Library Generate Options] Tab



2.9.1 Set the output of a standard library file

Select the build tool node on the project tree and select the [\[Library Generate Options\]](#) tab on the [Property panel](#). The setting to output a standard library file is made with the [\[Object\]](#) category.

Figure 2.62 [Object] Category



- (1) Set the output folder
Setting the output folder is made with the [\[Path of the output folder\]](#) property by directly entering to the text box or by the [...] button.
Up to 247 characters can be specified in the text box.
This property supports the following placeholder.
%BuildModeName%: Replaces with the build mode name.
"%BuildModeName%" is set by default.
- (2) Set the output file name
Setting the output file is made with the [\[Output file name\]](#) property by directly entering to the text box.
Up to 259 characters can be specified in the text box.
This property supports the following placeholders.
%ActiveProjectName%: Replaces with the active project name.
%MainProjectName%: Replaces with the main project name.
%ProjectName%: Replaces with the project name.
"%ProjectName%.hex" is set by default.

2.10 Preparation before Using the PIC/PID Function

In the PIC/PID function, a program whose code or data in the ROM has been converted into PIC or PID is called an application, and the program necessary to execute an application is called the master.

When the application and master are built, the option settings related to the PIC/PID function should be matched between the objects that compose the application and master.

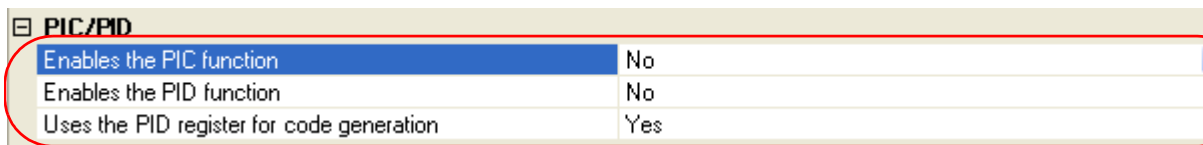
The procedure for setting build options for the application and master is given below.

Remark For details on the PIC/PID function, possible combinations of options, and how to create a startup program for the application or master, refer to Chapter 7, Startup, in "CubeSuite+: Coding".

(1) Setting build options

Build options related to the PIC/PID function can be set in the Project Tree panel. Select the build tool node for the master or application and set options in the [PIC/PID] category on the [\[Common Options\] tab](#) of the [Property panel](#).

Figure 2.63 [PIC/PID] Category



PIC/PID	
Enables the PIC function	No
Enables the PID function	No
Uses the PID register for code generation	Yes

(a) Setting build options for the master

Select [No] for the [Enables the PIC function] property (default).

Select [No] for the [Enables the PID function] property (default).

Select [Yes] for the [Uses the PID register for code generation] property (default).

(b) Setting build options for the application

Select [Yes(-pic)] for the [Enables the PIC function] property.

Select [The maximum bit width of the offset: 16 bits] (-pid=16) or [Yes (The maximum bit width of the offset: No limitation) (-pid=32)] for the [Enables the PID function] property.

2.11 Set Build Options Separately

Build options are set at the project or file level.

- Project level: See ["2.11.1 Set build options at the project level"](#)

- Project level: See ["2.11.2 Set build options at the file level"](#)

2.11.1 Set build options at the project level

To set options for build options for a project (main project or subproject), select the Build tool node on the project tree to display the [Property panel](#).

Select the component tabs, and set build options by setting the necessary properties.

Compile phase: [\[Compile Options\] tab](#)

Assemble phase: [\[Assemble Options\] tab](#)

Link phase (For the application project): [\[Link Options\] tab](#)

Link phase (For the library project): [\[Librarian Options\] tab](#)

Library Generate phase: [\[Library Generate Options\] tab](#)

2.11.2 Set build options at the file level

You can individually set compile and assemble options for each source file added to the project.

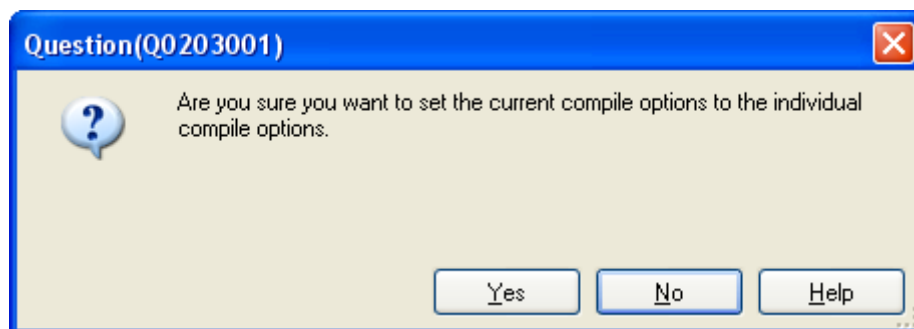
(1) When setting compile options for a C source file

Select a C source file on the project tree and select the [\[Build Settings\] tab](#) on the [Property panel](#). In the [Build] category, if you select [Yes] on the [Set individual compile option] property, the message dialog box ("[Figure 2.68 Message Dialog Box](#)") is displayed.

Figure 2.64 [Set individual compile option] Property

Build	
Set as build target	Yes
Set individual compile option	Yes
File type	C source

Figure 2.65 Message Dialog Box



If you click the [Yes] button in the dialog box, the [Individual Compile Options(C)] tab will be displayed.

Figure 2.66 Property Panel: [Individual Compile Options(C)] Tab

Property	
main.c Property	
Source	
Language of the C source file	C(C89)(-lang=c)
Additional include paths	Additional include paths[0]
Use whole include paths specified for build tool	Yes
Include files at the head of compiling units	Include files at the head of compiling units[0]
Macro definition	Macro definition[0]
Invalidates the predefined macro	
Enables information-level message output	No(-nomessage)
Suppresses the number of information-level messages	
Changes the warning-level messages to information-level messages	No
Changes the information-level messages to warning-level messages	No
Changes the information-level and warning-level messages to error-level	No
Permits comment (/ " /) nesting	No(-comment=nonest)
Checks the compatibility with an existing program	No
Character code of an input program	SJIS code(-sjis)
Object	
Output file type	Object module file(-output=obj)
Object file name	
Outputs debugging information	Yes(-debug)
Section name of program area	P
Section name of constant area	C
Section name of initialized data area	D
Section name of uninitialized data area	B
Section name of literal area	L
Section name of switch statement branch table area	W
Allocates uninitialized variables to 4-byte boundary alignment section	No
Allocates initialized variables to 4-byte boundary alignment sections	No
Allocates const qualified variables to 4-byte boundary alignment section	No
Allocates switch statement branch tables to 4-byte boundary alignment section	No
Adjustment for instruction in branch	None(-noinstalign)
Generates divisions and residues with DIV, DIVU, and the FDIV instruction	Yes
Character code of an output assembly-language file	SJIS code(-outcode=sjis)
List	
Outputs a source list file	No(-nolistfile)
Language of the C source file	
Selects language of the C source file. This corresponds to the -lang option of the compiler.	
Build Settings Individual Compile Options(C) File Information	

You can set compile options for the C source file by setting the necessary properties in this tab. Note that this tab takes over the settings of the [\[Compile Options\] tab](#) by default.

- (2) When setting compile options for a C++ source file
Select a C++ source file on the project tree and select the [\[Build Settings\] tab](#) on the [Property panel](#). In the [Build] category, if you select [Yes] on the [Set individual compile option] property, the message dialog box ("[Figure 2.68 Message Dialog Box](#)") is displayed.

Figure 2.67 [Set individual compile option] Property

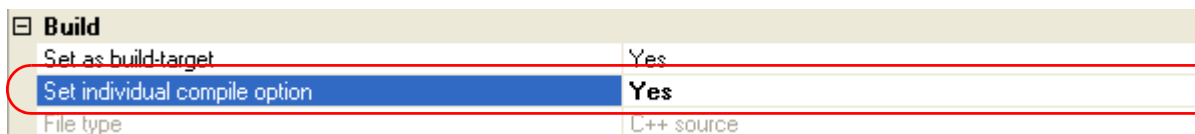
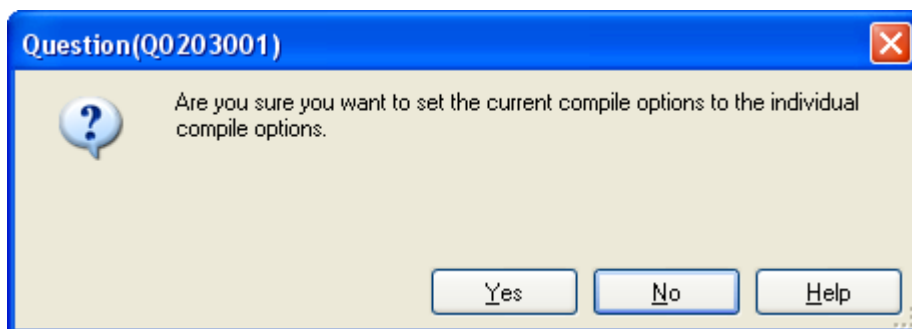


Figure 2.68 Message Dialog Box



If you click the [Yes] button in the dialog box, the [\[Individual Compile Options\(C++\)\] tab](#) will be displayed.

Figure 2.69 Property Panel: [Individual Compile Options(C++)] Tab

Property

main.cpp Property

Source

Language of the C++ source file	C++(-lang=cpp)
Additional include paths	Additional include paths[0]
Use whole include paths specified for build tool	Yes
Include files at the head of compiling units	Include files at the head of compiling units[0]
Macro definition	Macro definition[0]
Invalidates the predefined macro	
Enables information-level message output	No(-nomessage)
Suppresses the number of information-level messages	
Changes the warning-level messages to information-level message	No
Changes the information-level messages to warning-level message	No
Changes the information-level and warning-level messages to error	No
Permits comment (/* */) nesting	No(-comment=nonest)
Checks the compatibility with an existing program	No
Character code of an input program	SJIS code(-sjis)

Object

Output file type	Object module file(-output=obj)
Object file name	
Outputs debugging information	Yes(-debug)
Section name of program area	P
Section name of constant area	C
Section name of initialized data area	D
Section name of uninitialized data area	B
Section name of literal area	L
Section name of switch statement branch table area	W
Allocates uninitialized variables to 4-byte boundary alignment section	No
Allocates initialized variables to 4-byte boundary alignment sections	No
Allocates const qualified variables to 4-byte boundary alignment sections	No
Allocates switch statement branch tables to 4-byte boundary alignment sections	No
Adjustment for instruction in branch	None(-noinstalign)
Generates divisions and residues with DIV, DIVU, and the FDIV instructions	Yes
Character code of an output assembly-language file	SJIS code(-outcode=sjis)

List

Language of the C++ source file
Selects language of the C++ source file.
This option corresponds to the -lang option of the compiler.

Build Settings | **Individual Compile Options(C++)** | File Information

You can set compile options for the C++ source file by setting the necessary properties in this tab. Note that this tab takes over the settings of the [\[Compile Options\] tab](#) by default.

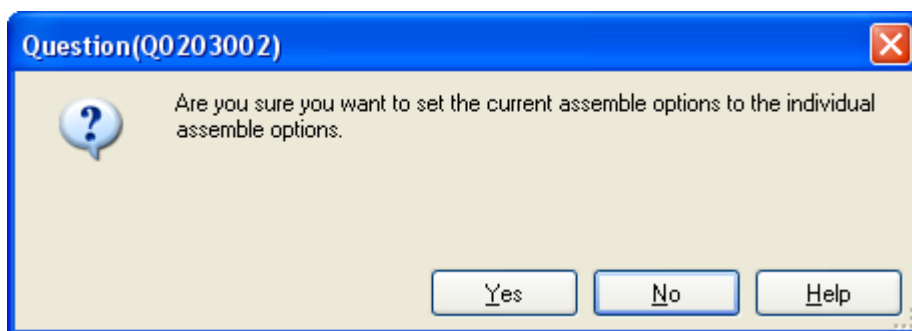
- (3) When setting assemble options for an assembler source file
Select an assembler source file on the project tree and select the [\[Build Settings\] tab](#) on the [Property panel](#). In the [Build] category, if you select [Yes] on the [Set individual assemble option] property, the message dialog box ("[Figure 2.71 Message Dialog Box](#)") is displayed.

Figure 2.70 [Set individual assemble option] Property

Build

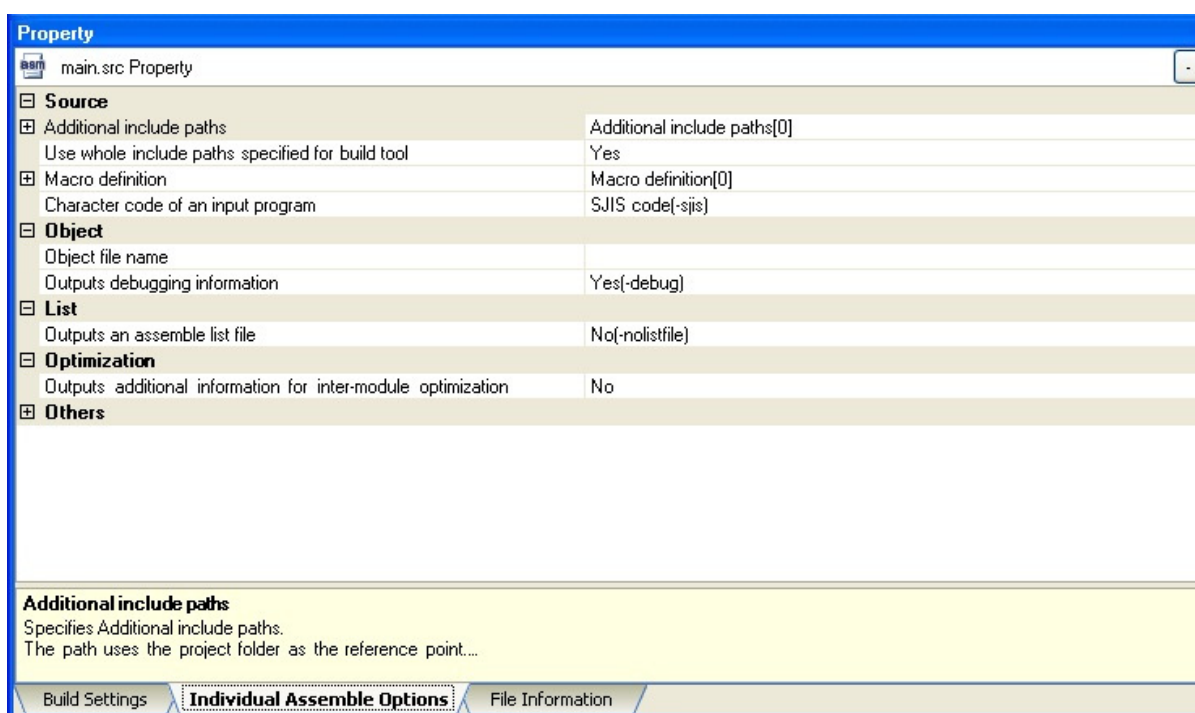
Set as build-target	Yes
Set individual assemble option	Yes
File type	Assembly source file

Figure 2.71 Message Dialog Box



If you click the [Yes] button in the dialog box, the [\[Individual Assemble Options\]](#) tab will be displayed.

Figure 2.72 Property Panel: [Individual Assemble Options] Tab



You can set assemble options for the assembler source file by setting the necessary properties in this tab. Note that this tab takes over the settings of the [\[Assemble Options\]](#) tab by default.

2.12 Make Settings for Build Operations

This section explains operations on a build.

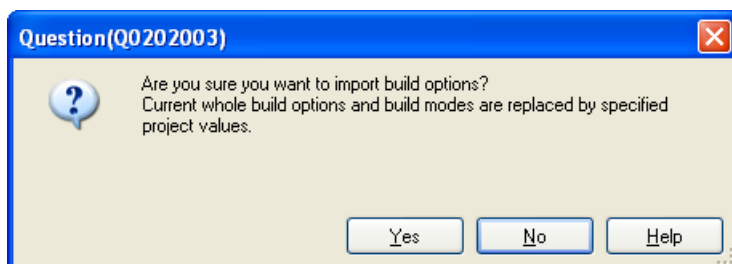
- [Import the build options of other project](#)
- [Set the link order of files](#)
- [Change the file build order of subprojects](#)
- [Display a list of build options](#)
- [Change the file build target project](#)
- [Add a build mode](#)
- [Change the build mode](#)
- [Delete a build mode](#)
- [Set the current build options as the standard for the project](#)

2.12.1 Import the build options of other project

You can import the build options of other project to the current project.

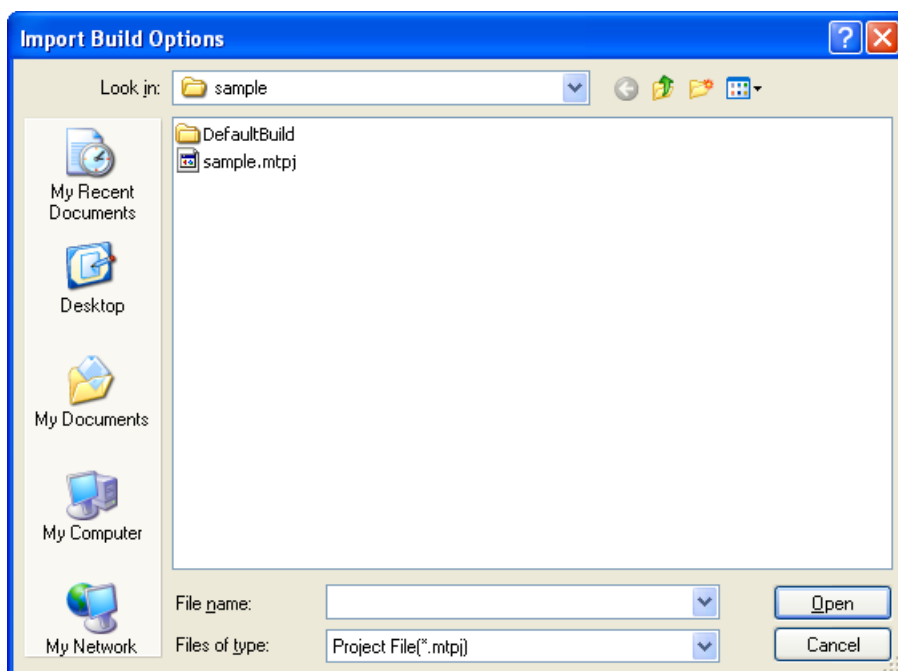
On the project tree, select the Build tool node, and then select [Import Build Options...] from the context menu. The following message dialog box will open.

Figure 2.73 Message Dialog Box



Click [Yes] in the dialog box. The [Select Import File dialog box](#) will open.

Figure 2.74 Import Build Options Dialog Box



In the dialog box, select the target project file for import the build options and click the [Open] button.
The build options of the selected project file are imported to the current project.

Remarks 1. The conditions of the project that is importable are shown below.

- The build tool is the same.
- The type of the project (application, library, etc.) is the same.
- The project has been created by CubeSuite+ with the same version.

Remarks 2. The target build options for importing are only the general options set in the properties of the build tool.
The setting of the standard build options (see "2.12.9 Set the current build options as the standard for the project") and the individual options are not imported.

Remarks 3. All the build modes of the import target are imported.
However, the build modes of the current project (other than DefaultBuild) are deleted.

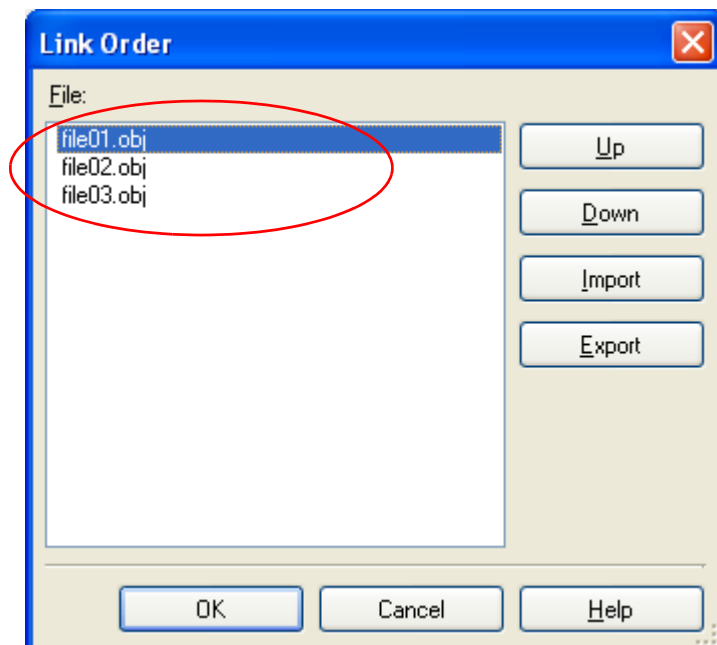
Remarks 4. The version of the build tool to be used is imported.

2.12.2 Set the link order of files

The link order of object module files and library files is decided automatically, but you can also set the order.
The procedures for performing this operation are described below.

- (1) Open the Link Order dialog box
On the project tree, select the Build tool node, and then select [Set Link Order...] from the context menu. The [Link Order dialog box](#) opens.

Figure 2.75 Link Order Dialog Box



The names of the following files are listed in [File] in the order that the files are input to the linker.

- Object module files generated from the source files added to the selected main project or subproject
- Object module files added directly to the project tree of the selected main project or subproject

Remark The default order is the order the files are added to the project.
Object module files created from newly added source files and newly added object module files are added after the last object module file in the list.

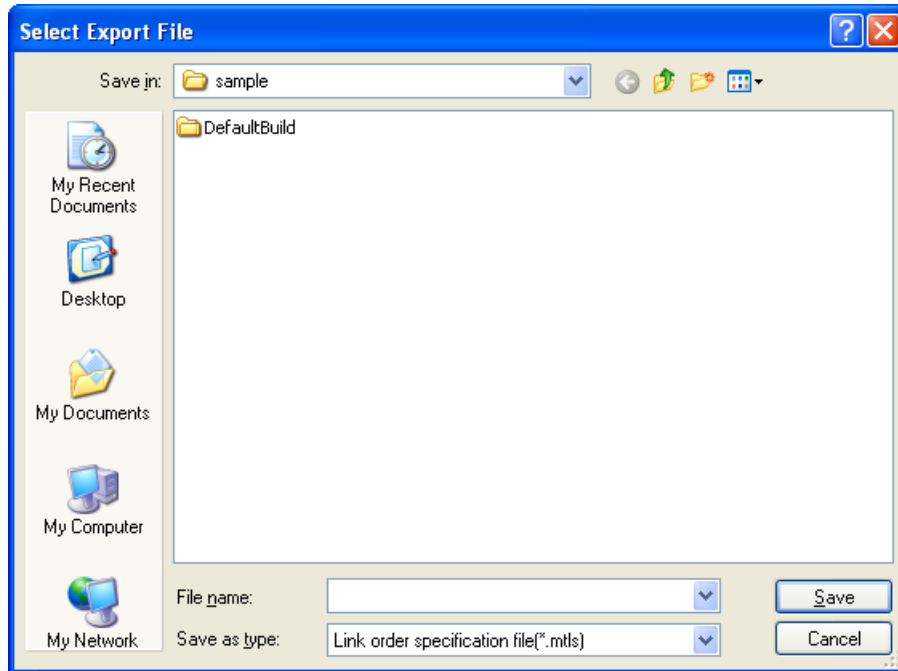
- (2) Change the file display order
By changing the display order of the files, you can set the input order of the files to the linker.
Change the file display order by one of the following procedures.
 - Move the file name by using the [Up] and [Down] buttons.

- Drag and drop the file name.
- Use a link order specification file.

Remark You can use a link order specification file to change the display order on a file basis.
The procedures for performing this operation are described below.

- (a) Generate a link order specification file
Click the [Export] button in the [Link Order dialog box](#) to open the [Select Export File dialog box](#).

Figure 2.76 Select Export File Dialog Box



In the dialog box, specify the file (link order specification file) that is output the list of the file names displayed in [File] in the [Link Order dialog box](#).

Click the [Save] button to generate the link order specification file.

Caution Only the file names are output to the link order specification file.
If a file with the same name exists, check the location of the file in the popup display after installing the link order specification file.

- (b) Edit the link order specification file
Open the link order specification file with a editor, and then change the description order of the file names.
The code example of the link order specification file is shown below.

```
# CubeSuite+ Vx.xx.xx Link order specification file
# SampleProject: XXXXXX xx, xxxx

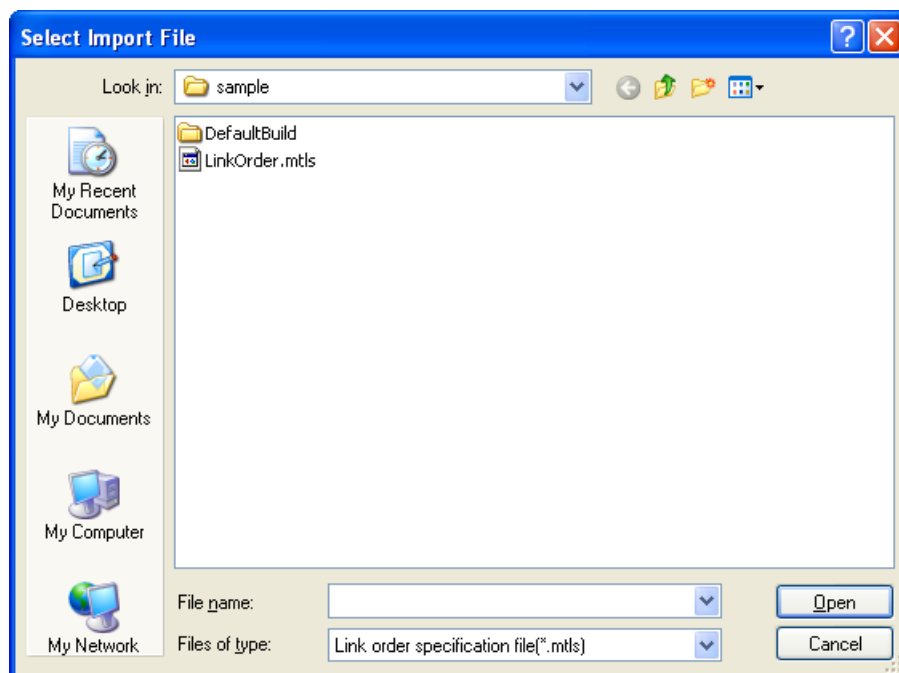
file01.obj
file03.obj
file02.obj
:
```

The following points should be noted when describing the link order specification file.

- Describe one file name on one line.
- Uppercase characters and lowercase characters are not distinguished for the file name.
- If the line begins with "#", the line is interpreted as a comment.
- A space or tab is ignored.

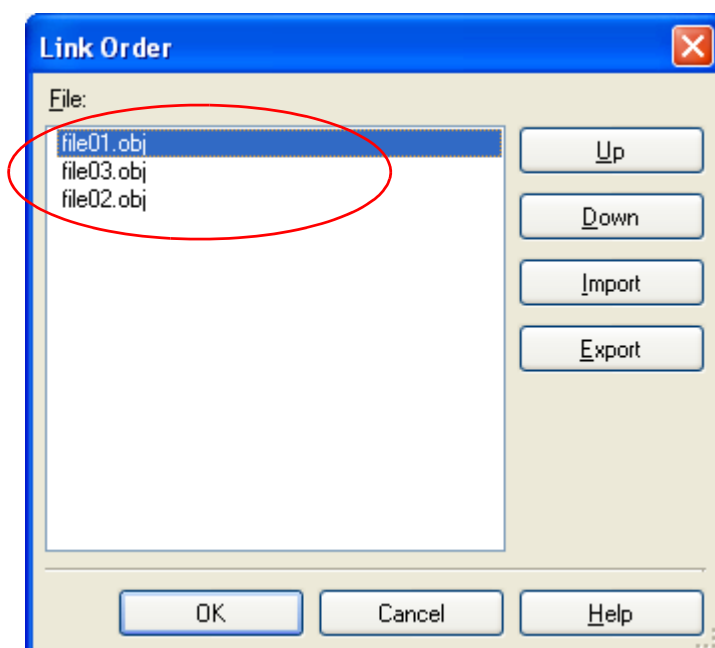
- (c) Import the link order specification file
Click the [Import] button in the [Link Order dialog box](#) to open the [Select Import File dialog box](#).

Figure 2.77 Select Import File Dialog Box



In the dialog box, select the link order specification file and click the [Open] button.
The description order of the file names are acquired from the selected link order specification file, and then they are reflected in [File] in the [Link Order dialog box](#).

Figure 2.78 Link Order Dialog Box (After Setting Link Order)



- Cautions 1.** The file that is described in the link order specification file and is not added to the project is not displayed.
If the corresponding file exists, the list of the file names will be displayed in the [Output panel](#).
- Cautions 2.** The file that is added to the project and is not described in the link order specification file is displayed in the end of [File].

Cautions 3. If a file with the same name exists, check the location of the file in the popup display (it will appear when you hover the mouse cursor over the file name).
To change the link order, use the [Up] and [Down] buttons, or drag and drop the file names.

- (3) Set the file link order
Click the [OK] button in the [Link Order dialog box](#) to set the input order of the files to the linker.

2.12.3 Change the file build order of subprojects

Builds are run in the order of subproject, main project, but when there are multiple subprojects added, the build order of subprojects is their display order on the project tree.

To change the display order of the subprojects on the project tree, drag the subproject to be moved and drop it on the desired location.

2.12.4 Display a list of build options

You can display the list of build options set currently on the [Property panel](#) for the project (main project and subproject). If you select [Build Options List] from the [Build] menu, the current settings of the options for the project are displayed on the [Build Tool] tab from the [Output panel](#) in the build order.

Remark You can change the display format of the build option list.
Select the build tool node on the project tree and select the [\[Common Options\] tab](#) on the [Property panel](#).
Set the [Format of build option list] property in the [Others] category.

Figure 2.79 [Format of build option list] Property

Others	
Output message format	%FileName%
Format of build option list	%FileName% : %Program% %Options%
Temporary folder	
Commands executed before build processing	Commands executed before build processing[0]
Commands executed after build processing	Commands executed after build processing[0]

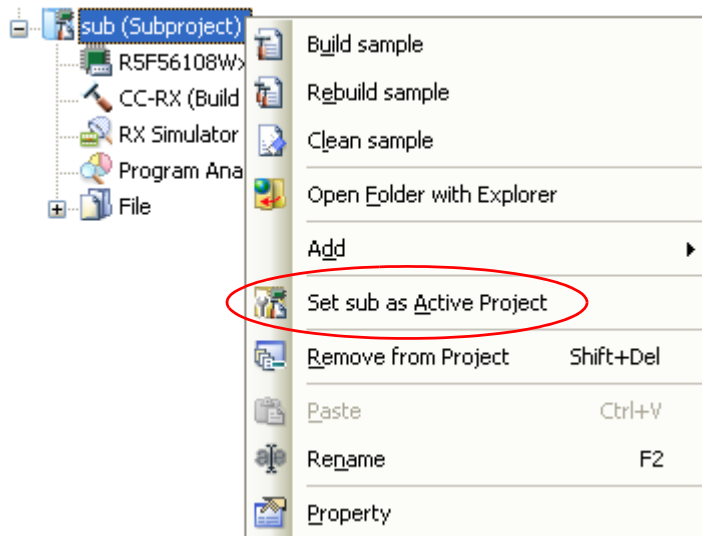
"%TargetFiles% : "%TargetFiles% : %Program% %Options%" is set by default.

Remark "%TargetFiles%", "%Program%", and "%Options%" are placeholders. They are replaced with the file name being built, program name under execution, and command line option under build execution.

2.12.5 Change the file build target project

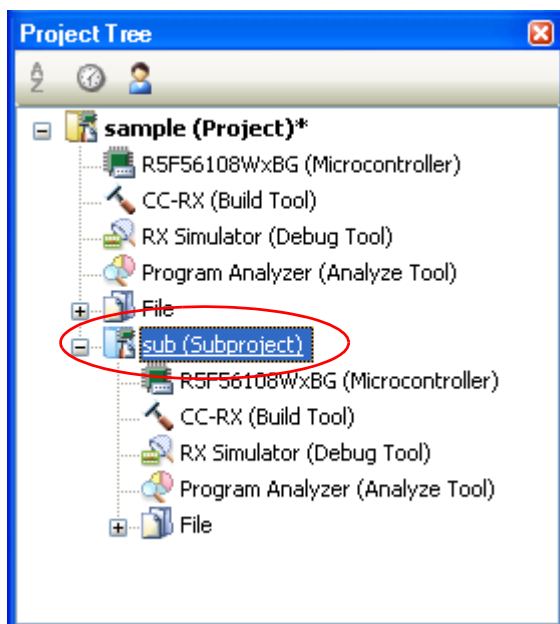
When running a build that targets a specific project (main project or subproject), you must set that project as the "active project".

To set the active project, select the main project or subproject to be set as the active project on the project tree and select [Set *selected subproject* as Active Project] from the context menu.

Figure 2.80 [Set *selected project* as Active Project] Item

When a project is set as the active project, that project is underlined.

Figure 2.81 Active Project



Remarks 1. Immediately after creating a project, the main project is the active project.

Remarks 2. When you remove a subproject that set as the active project from a project, the main project will be the active project.

2.12.6 Add a build mode

When you wish to change the build options and macro definitions according to the purpose of the build, you can collectively change those settings. Build options and macro definition settings are organized into what is called "build mode", and by changing the build mode, you eliminate the necessity of changing the build options and macro definition settings every time.

The build mode prepared by default is only "DefaultBuild". Add a build mode according to the purpose of the build.

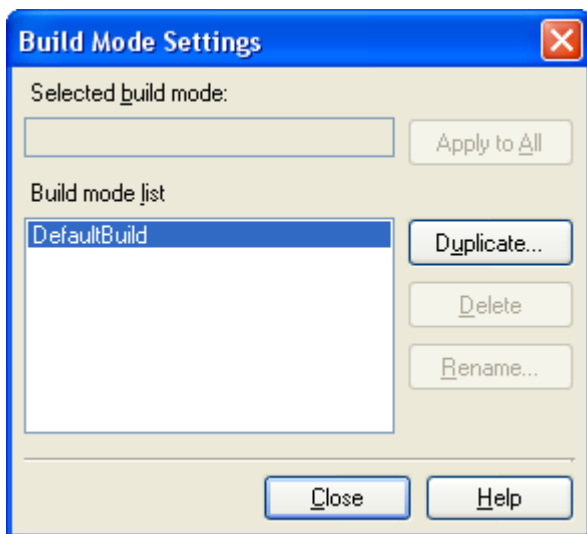
The method to add a build mode is shown below.

(1) Create a new build mode

Creating a new build mode is performed with duplicating an existing build mode.

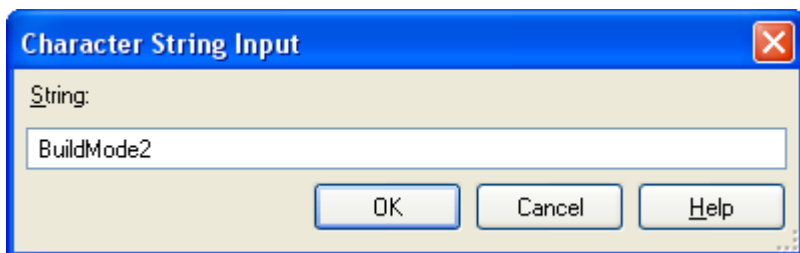
Select [Build Mode Settings...] from the [Build] menu. The [Build Mode Settings dialog box](#) opens.

Figure 2.82 Build Mode Settings Dialog Box



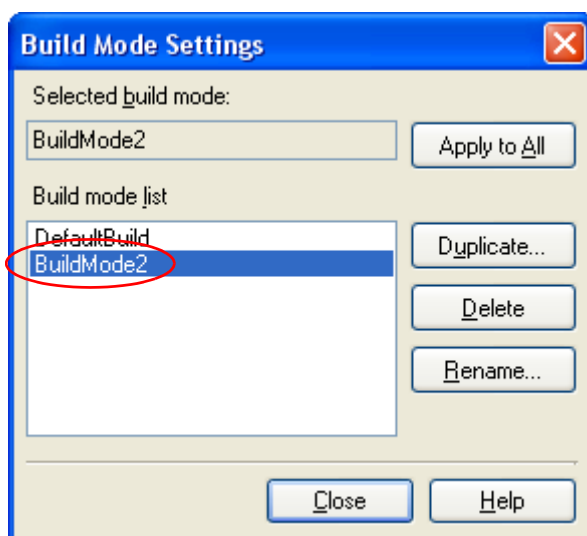
Select the build mode to be duplicated from the build mode list and click the [Duplicate...] button. The [Character String Input dialog box](#) opens.

Figure 2.83 Character String Input Dialog Box



In the dialog box, enter the name of the build mode to be created and then click the [OK] button. The build mode with that name will be duplicated. The created build mode is added to the build modes of the main project and all the subprojects which belong to the project.

Figure 2.84 Build Mode Settings Dialog Box (After Adding Build Mode)



- (2) Change the build mode
Change the build mode to the newly created build mode (see "[2.12.7 Change the build mode](#)").
- (3) Change the setting of the build mode

Select the build tool node on the project tree and change the build options and macro definition settings on the [Property panel](#).

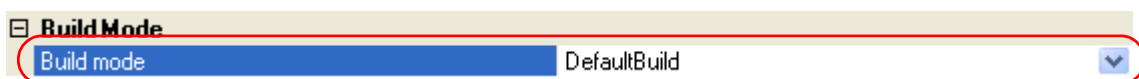
Remark Creating a build mode is regarded a project change. When closing the project, you will be asked to confirm whether or not to save the build mode.

2.12.7 Change the build mode

When you wish to change the build options and macro definitions according to the purpose of the build, you can collectively change those settings. Build options and macro definition settings are organized into what is called "build mode", and by changing the build mode, you eliminate the necessity of changing the build options and macro definition settings every time.

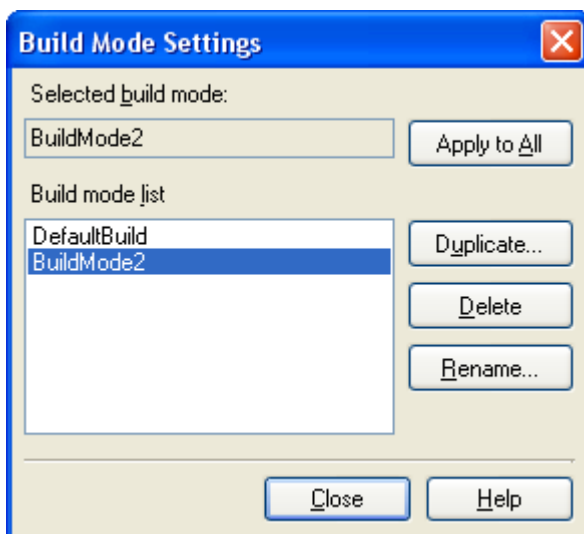
- (1) When changing the build mode for the main project or subprojects
Select the Build tool node of the target project on the project tree and select the [\[Common Options\] tab](#) on the [Property panel](#). Select the build mode to be changed to on the [Build mode] property in the [Build Mode] category.

Figure 2.85 [Build Mode] Property



- (2) When changing the build mode for the entire project
Select [Build Mode Settings...] from the [Build] menu. The [Build Mode Settings dialog box](#) opens.

Figure 2.86 Build Mode Settings Dialog Box



If you select the build mode to be changed from the build mode list, the selected build mode is displayed in [Selected build mode]. If you click the [Apply to All] button, the build mode for the main project and all the subprojects which belong to the project will be changed to the build mode selected in the dialog box.

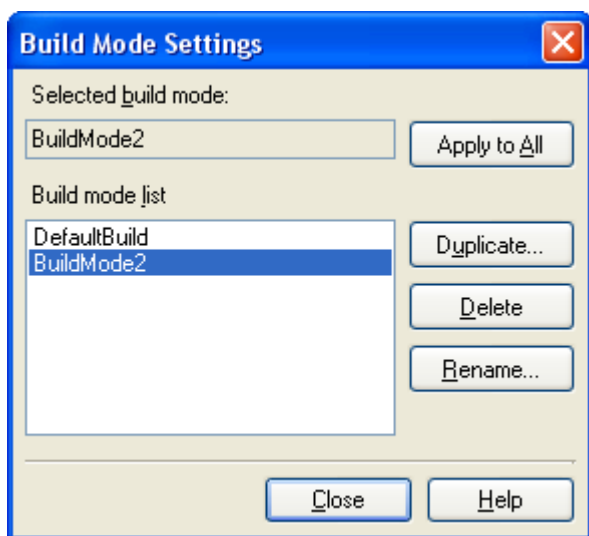
Caution For projects that the selected build mode does not exist, the build mode is duplicated from "DefaultBuild" with the selected build mode name, and the build mode is changed to the duplicated build mode.

- Remarks 1.** The build mode prepared by default is only "DefaultBuild". See ["2.12.6 Add a build mode"](#) for the method of adding a build mode.
- Remarks 2.** You can change the name of the build mode by selecting the build mode from the build mode list and clicking the [Rename...] button. However, you cannot change the name of "DefaultBuild".

2.12.8 Delete a build mode

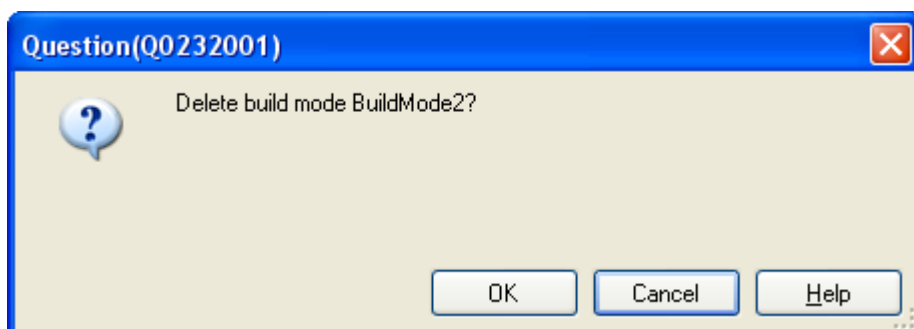
Deleting a build mode is performed with the [Build Mode Settings dialog box](#).
Select [Build Mode Settings...] from the [Build] menu. The dialog box opens.

Figure 2.87 Build Mode Settings Dialog Box



Select the build mode to be deleted from the build mode list and click the [Delete] button. The Message dialog box below opens.

Figure 2.88 Message Dialog Box



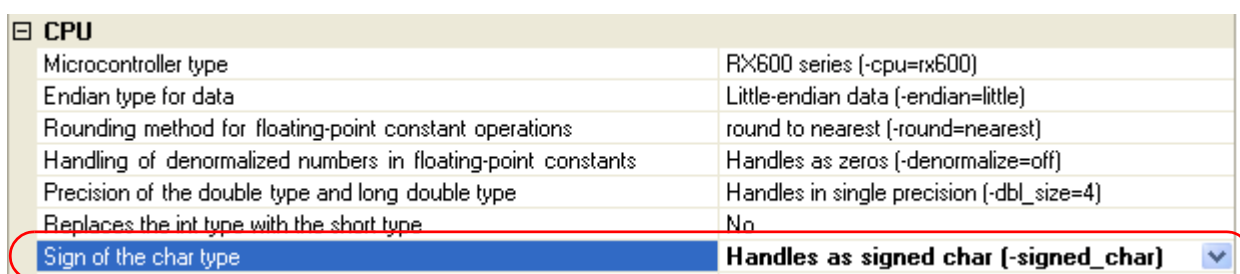
To continue with the operation, click the [OK] button in the dialog box. The selected build mode is deleted from the project.

Caution You cannot delete "DefaultBuild".

2.12.9 Set the current build options as the standard for the project

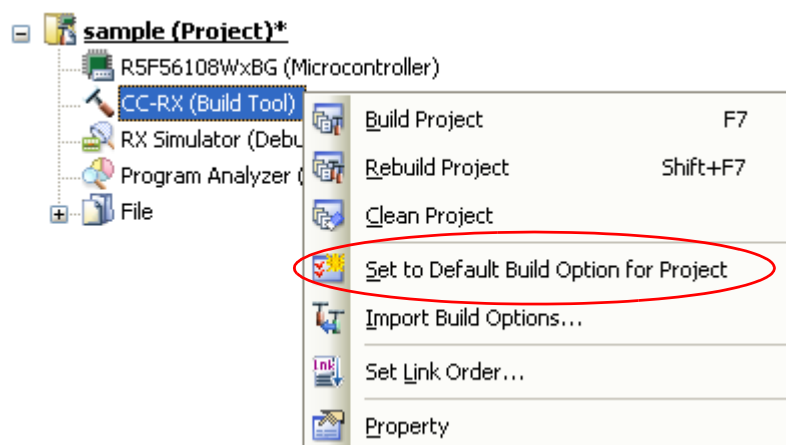
On the [Property panel](#), if you add a change to the settings for the standard build options, the value of the property will be displayed in boldface.

Figure 2.89 Property Panel (After Changing Standard Build Option)



To make the build options for the currently selected project (main project or subproject) the standard build options (remove the boldface), select the Build tool node on the project tree and select [Set to Default Build Option for Project] from the context menu.

Figure 2.90 [Set to Default Build Option for Project] Item



The value of the properties after setting them as the standard build option are as shown below.

Figure 2.91 Property Panel (After Setting Standard Build Option)

CPU	
Microcontroller type	RX600 series (-cpu=rx600)
Endian type for data	Little-endian data (-endian=little)
Rounding method for floating-point constant operations	round to nearest (-round=nearest)
Handling of denormalized numbers in floating-point constants	Handles as zeros (-denormalize=off)
Precision of the double type and long double type	Handles in single precision (-dbl_size=4)
Replaces the int type with the short type	No
Sign of the char type	Handles as signed char (-signed_char) ▼

Caution When the main project is selected, only the main project settings are made. Even if subprojects are added, their settings are not made.

2.13 Run a Build

This section explains operations related to running a build.

(1) Build types

The following types of builds are available.

Table 2.1 Build Types

Type	Description
Build	Out of build target files, runs a build of only updated files. See "2.13.1 Run a build of updated files".
Rebuild	Runs a build of all build target files. See "2.13.2 Run a build of all files".
Rapid build	Runs a build in parallel with other operations. See "2.13.3 Run a build in parallel with other operations".
Batch build	Runs builds in batch with the build modes that the project has. See "2.13.4 Run builds in batch with build modes".

Remarks 1. Builds are run in the order of subproject, main project.
Subprojects are built in the order that they are displayed on the project tree (see "2.12.3 Change the file build order of subprojects").

Remarks 2. If there are files being edited with the **Editor panel** when running a build, rebuild, or batch build, then all these files are saved.

(2) Display execution results

The execution results of the build (output messages of the build tool) are displayed in each tab on the [Output panel](#).

- Build, rebuild, or batch build: [All Messages] tab and [Build Tool] tab
- Rapid build: [Rapid Build] tab

Figure 2.92 Build Execution Results (Build, Rebuild, or Batch Build)

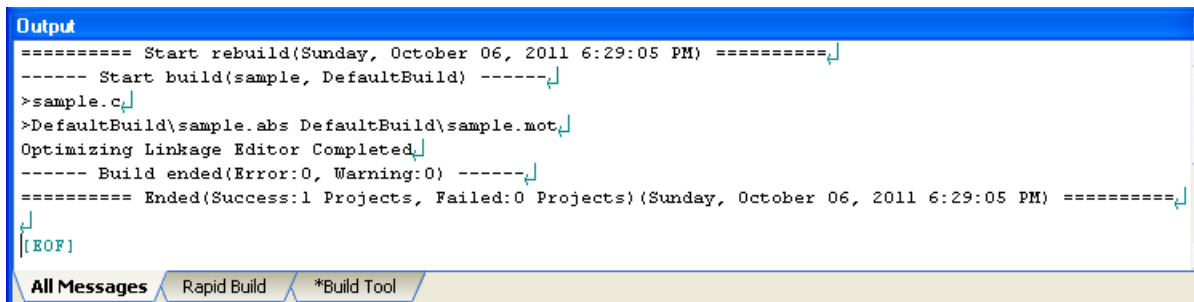
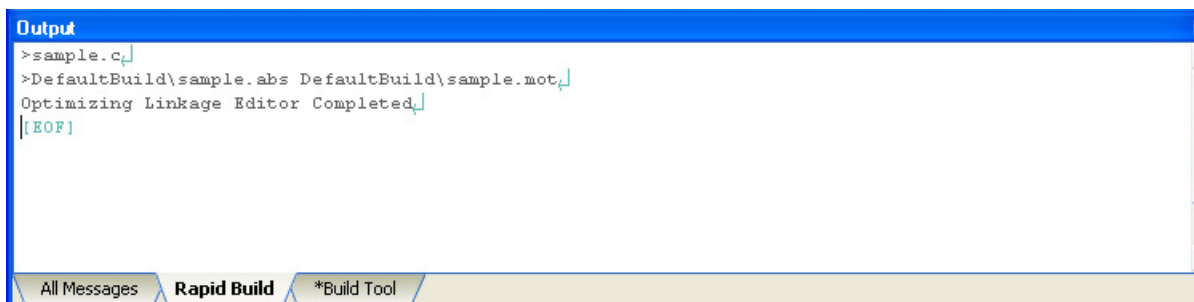


Figure 2.93 Build Execution Results (Rapid Build)



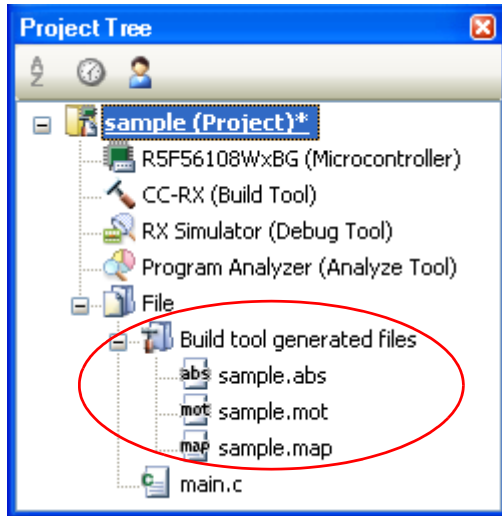
Remarks 1. The text in the [Rapid Build] tab becomes dimmed.

Remarks 2. When a file name or line number can be obtained from the output messages, if you double click on the message, you can jump to the relevant line in the file.

Remarks 3. If you press the [F1] key when the cursor is on a line displaying the warning or error message, you can display the help related to that line's message.

Files generated by the build tool appear on the [Project Tree panel](#), under the Build tool generated files node.

Figure 2.94 Build Tool Generated Files



Remark Files displayed under the Build tool generated files node are as follows.

- For application projects
 - Load module file (*.abs)
 - Hex file (*.hex)
 - S record file (*.mot)
 - Binary data file (*.bin)
 - Map file (*.map)
- For library projects
 - Library file (*.lib)
 - Relocatable module file (*.rel)
 - Map file (*.map, *.lbp)

Caution The Build tool generated files node is created during build.
This node will no longer appear if you reload the project after building.

2.13.1 Run a build of updated files

Out of build target files, run a build of only updated files (hereafter referred to as "build").

Running a build is performed for the entire project (main project and subprojects) or active project (see ["2.12.5 Change the file build target project"](#)).


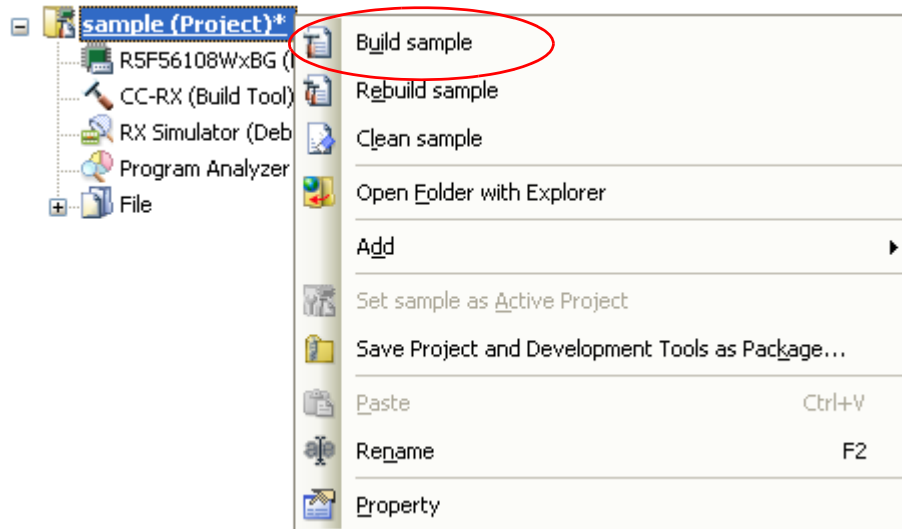
- (1) When running a build of the entire project
Click  on the toolbar.
- (2) When running a build of the active project
Select the project, and then select [Build *active project*] from the context menu.

Figure 2.95 [Build active project] Item



Remark If the included source files are not built after editing the header file and running the build, update the file dependencies (see ["2.3.6 Update file dependencies"](#)).

2.13.2 Run a build of all files

Run a build of all build target files (hereafter referred to as "rebuild").

Running a rebuild is performed for the entire project (main project and subprojects) or active project (see ["2.12.5 Change the file build target project"](#)).


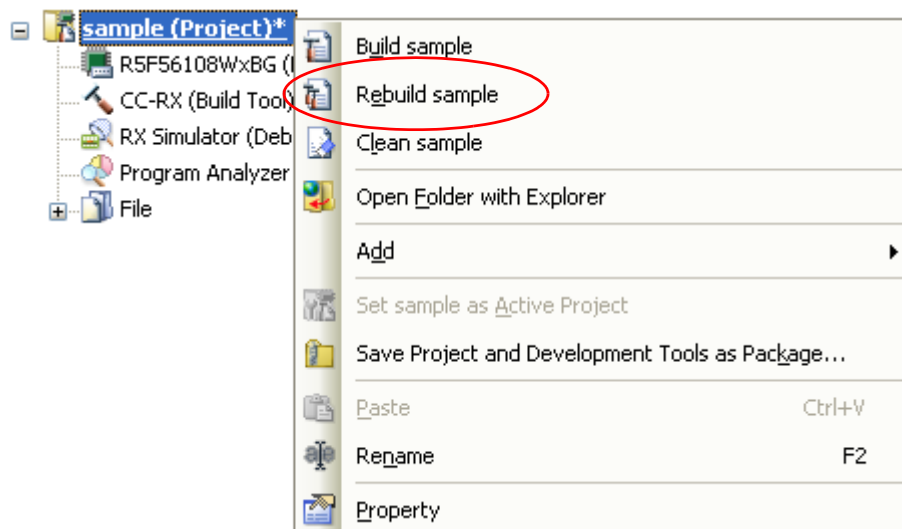
- (1) When running a rebuild of the entire project
Click  on the toolbar.
- (2) When running a rebuild of the active project
Select the project, and then select [Rebuild active project] from the context menu.

Figure 2.96 [Rebuild active project] Item



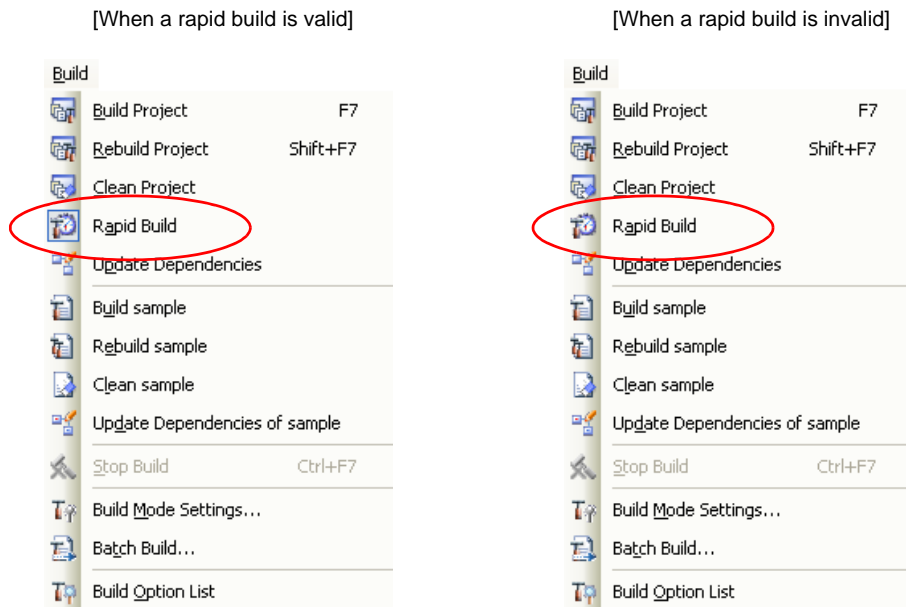
2.13.3 Run a build in parallel with other operations

CubeSuite+ has a function that a build is started automatically when one of the following events occurs (hereafter referred to as "rapid build").

- When any one of the following files that are added to the project is updated:
(C source file, C++ source file, assembler source file, header file, object module file, relocatable module file, and library file)
- When a build target file has been added to or removed from the project
- When the link order of object module files has changed
- When the properties of the build tool or build target files are changed

If a rapid build is enabled, it is possible to perform a build in parallel with the above operations.
To enable/disable a rapid build, select [Rapid Build] from the [Build] menu. A rapid build is enabled by default.

Figure 2.97 [Rapid Build] Item



Remarks 1. After editing source files, it is recommend to save frequently by pressing the [Ctrl] + [S] key.

Remarks 2. Enabling/disabling a rapid build is set for the entire project (main project and subprojects).

Remarks 3. If you disable a rapid build while it is running, it will be stopped at that time.

Caution This function will also be enabled in an external text editor if the [Observe registered files changing] checkbox is selected in the [Build/Debug] category of the [Option dialog box](#).

2.13.4 Run builds in batch with build modes

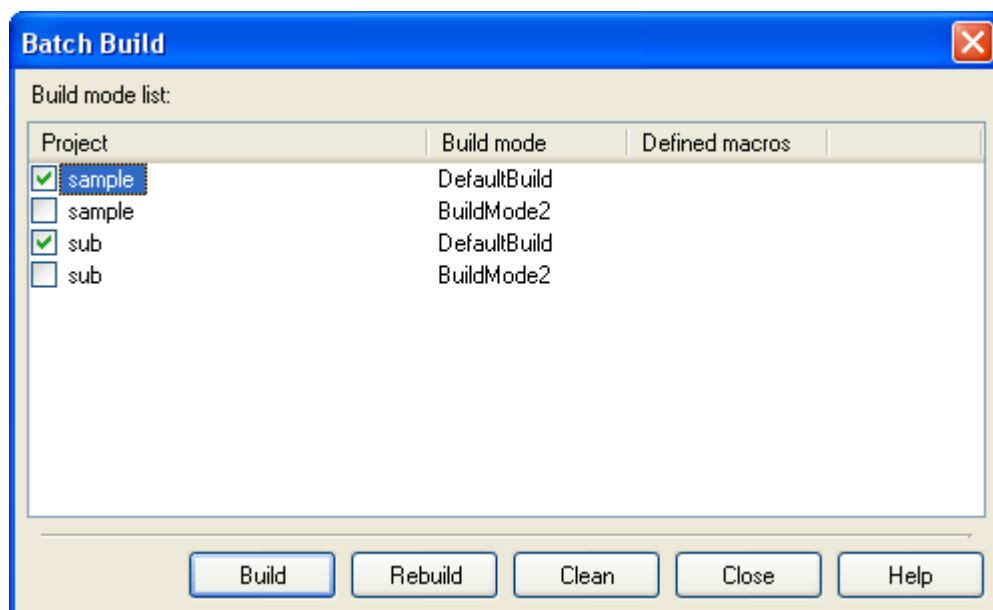
You can run builds, rebuilds and cleans in batch with the build modes that the project (main project and subproject) has (hereafter referred to as "batch build").

Remark See the sections below for a build, rebuild, and clean.

- Build: See ["2.13.1 Run a build of updated files"](#).
- Rebuild: See ["2.13.2 Run a build of all files"](#).
- Clean: See ["2.13.8 Delete intermediate files and generated files"](#).

Select [Batch Build] from the [Build] menu. The [Batch Build dialog box](#) opens.

Figure 2.98 Batch Build Dialog Box



In the dialog box, the list of the combinations of the names of the main project and subprojects in the currently opened project and their build modes is displayed.

Select the check boxes for the combinations of the main project and subprojects and build modes that you wish to run a batch build, and then click the [Build], [Rebuild], or [Clean] button.

Caution This build tool does not support the display of [Defined macros].

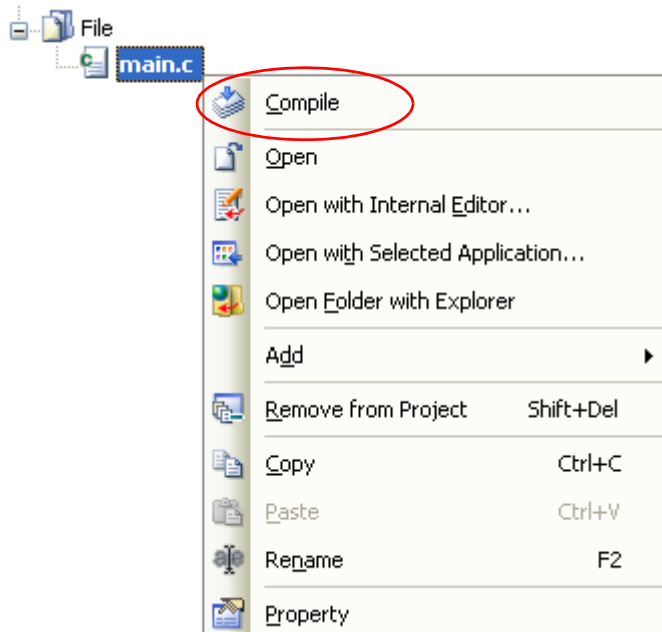
Remark The batch build order follows the project build order, the order of the subprojects, main project. When multiple build modes are selected for a single main project or subproject, after running builds of the subproject with all the selected build modes, the build of the next subproject or main project is run.

2.13.5 Compile/assemble individual files

You can just compile or assemble for each source file added to the project.

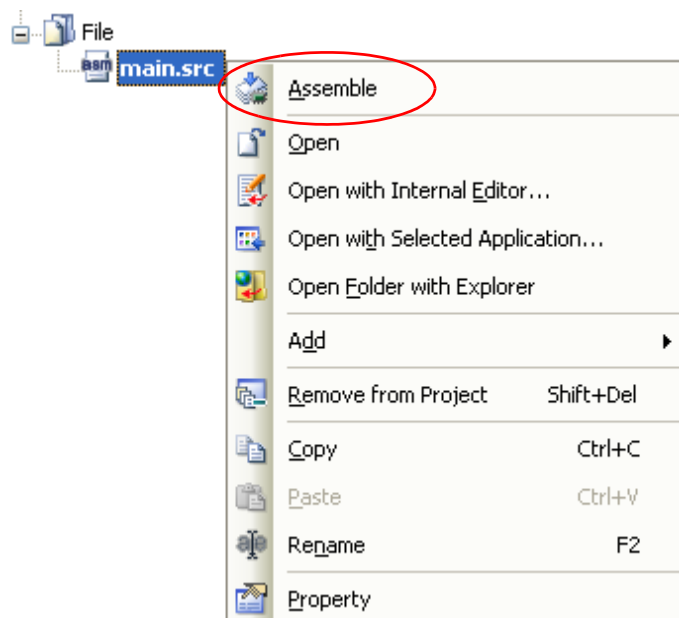
- (1) When compiling a C source file
Select a C source file on the project tree and select the [Compile] from the context menu.

Figure 2.99 [Compile] Item




- (2) When assembling an assembler source file
Select an assembler source file on the project tree and select the [Assemble] from the context menu.

Figure 2.100 [Assemble] Item



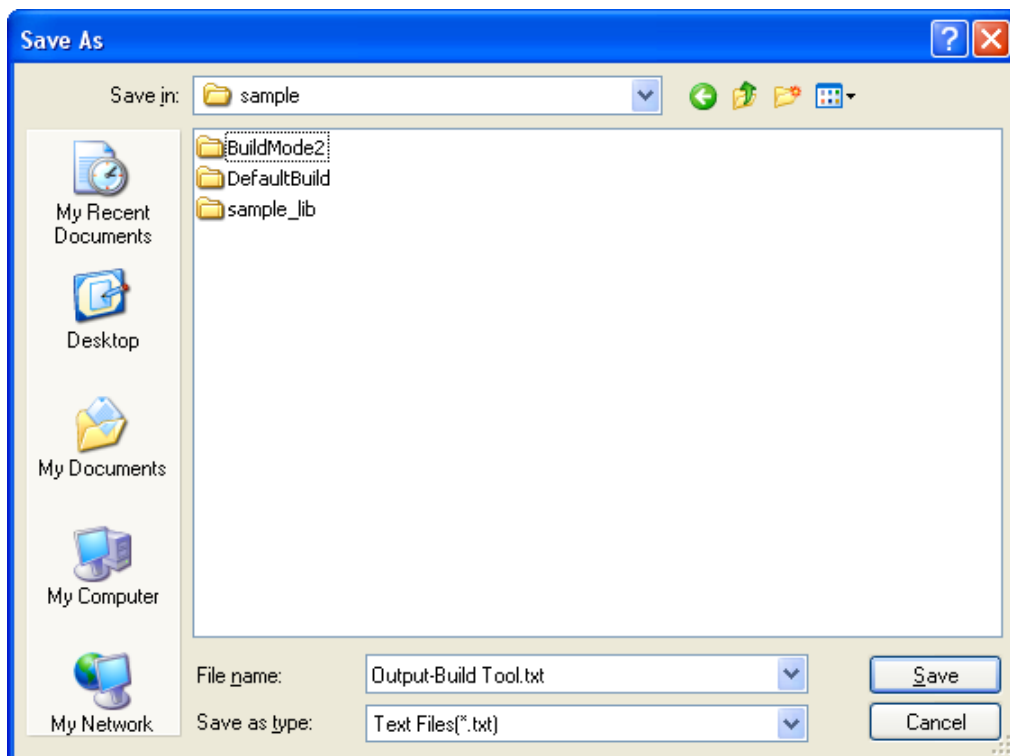
2.13.6 Stop running a build

To stop running a build, rebuild, or batch build, click  on the toolbar.

2.13.7 Save the build results to a file

You can save the execution results of the build (output messages of the build tool) that displayed on the [Output panel](#). Select the [Build Tool] tab on the panel, and then select [Save Output - Build Tool As...] from the [File] menu. The [Save As dialog box](#) opens.

Figure 2.101 Save As Dialog Box



In the dialog box, specify the file to be saved and then click the [Save] button.

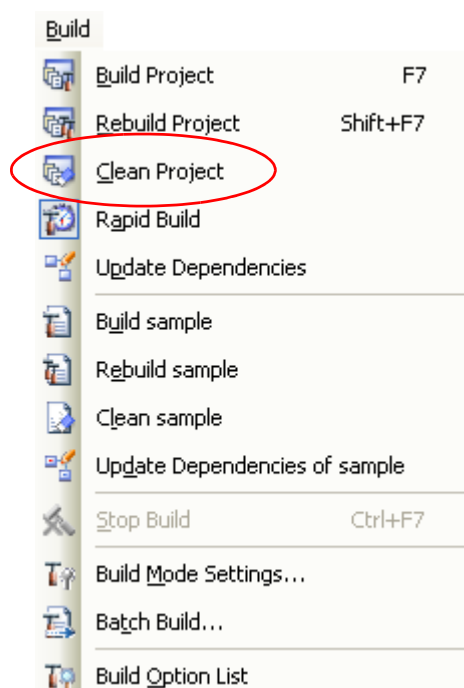
2.13.8 Delete intermediate files and generated files

You can delete all the intermediate files and generated files output by running a build (hereafter referred to as "clean").

Running a clean is performed for the entire project (main project and subprojects) or active project (see ["2.12.5 Change the file build target project"](#)).

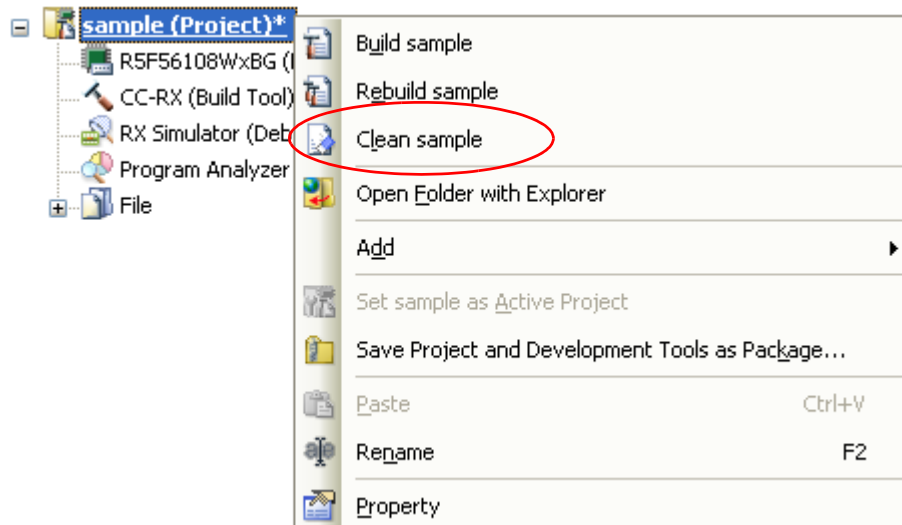
- (1) When running a clean of the entire project
From the [Build] menu, select [Clean Project].

Figure 2.102 [Clean Project] Item



- (2) When running a clean of the active project
Select the project, and then select [Clean *active project*] from the context menu.

Figure 2.103 [Clean active project] Item



2.14 Estimate the Stack Capacity

To estimate the stack capacity, use the Call Walker.

The Call Walker performs a static analysis, and displays the symbols called by a symbol in a tree format, as well as stack information for each symbol (symbol name, attributes, address, size, stack size, and file name) in list format.

To start the Call Walker, select Windows® [start] >> [All Programs] >> [Renesas Electronics CubeSuite+] >> [Utilities] >> [Call Walker].

To exit the Call Walker, from the Call Walker, select the [File] menu >> [Exit].

See the Call Walker help for details on the procedure for using it, from the Call Walker, select the [Help] menu >> [Help Topics].

3. BUILD OUTPUT LISTS

This chapter describes the format and other aspects of files output by a build via each command.

3.1 Assemble List File

This section covers the contents and format of the assemble list file output by the assembler.

The source list file contains the compilation and assembly results. Table 3.1 shows the structure and contents of the source list.

Table 3.1 Structure and Contents of Source List

No	Output Information	Contents	Suboption ^{Note}	When -show Option is not Specified
1	Source information	C/C++ source code corresponding to assembly source code	-show=source	Not output
2	Object information	Machine code used in object programs and the assembly source code	None	Output
3	Statistics information	Total number of errors, number of source program lines, and size of each section	None	Output
4	Command specification information	File names and options specified by the command	None	Output

Note Valid when the **-listfile** option is specified.

3.1.1 Source Information

The source information is included in the object information when the **-show=source** option is specified. For an example of source information, refer to the next section, Object Information.

3.1.2 Object Information

Figure 3.1 shows an example of object information output.

```
* RX FAMILY ASSEMBLER V2.00.00 [15 Feb 2013] * SOURCE LIST Mon Feb 18 20:15:19 2013
(1)      (2)      (3)      (4)
LOC.      OBJ.      OXMDA SOURCE STATEMENT

;RX Family C/C++ Compiler (V2.00.00 [15 Feb 2013]) 18-
Feb-2013 20:15:19

;*** CPU TYPE ***

;-ISA=RXV1

;*** COMMAND PARAMETER ***

;-output=src=sample.src
;-listfile
;-show=source
;sample.c

.glob_x
.glob_y
.glob_func02
.glob_func03
```

```

        .glb_func01
        (5)      (6)
;LineNo. C-SOURCE STATEMENT

        .SECTIONNP,CODE
00000000  _func02:
        .STACK_func02=12
        ;      1 #include "include.h"
        ;      2 int func01(int);
        ;      3 int func03(int);
        ;      4
        ;      5 int func02(int z)
00000000 6E67  PUSHM R6-R7
00000002 EF16  MOV.L R1, R6
        ;      6 {
        ;      7      x = func01(z);
00000004 05rrrrrrr A  BSR _func01
00000008 FB72rrrrrrrr MOV.L #_x, R7
0000000E E371  MOV.L R1, [R7]
        ;      8      if (z == 2) {
00000010 6126  CMP #02H, R6
00000012 18      S  BNE L12
00000013      L11:: bb3
        ;      9      x++;
00000013 6211  ADD #01H, R1
00000015 08      S  BRA L13
00000016      L12:: bb6
        ;      10      } else {
        ;      11      x = func03(x + 2);
00000016 6221  ADD #02H, R1
00000018 39rrrrr W  BSR _func03
0000001B      L13:: bb13
0000001B E371  MOV.L R1, [R7]
        ;      12      }
        ;      13      return x;
        ;      14 }
0000001D 3F6702  RTSD #08H, R6-R7
00000020  _func03:
        .STACK_func03=4
        ;      15
        ;      16 int func03(int p)
        ;      17 {
        ;      18      return p+1;
00000020 6211  ADD #01H, R1
        ;      19 }
00000022 02      RTS
        .SECTIONND,ROMDATA,ALIGN=4
00000000  _y:
00000000 01000000 .lword00000001H
        .END

```

Item Num-ber	Description
(1)	Location information (LOC.) Location address of the object code that can be determined at assembly.
(2)	Object code information (OBJ.) Object code corresponding to the mnemonic of the source code.

Item Number	Description					
(3)	Line information (0XMDA) Results of source code processing by the assembler. The following shows the meaning of each symbol.					
	0	X	M	D	A	Description
	0-30					Shows the nesting level of include files.
		X				Shows the line where the condition is false in conditional assembly when -show=conditions is specified.
			M			Shows the line expanded from a macro instruction when -show=expansions is specified.
			D			Shows the line that defines a macro instruction when -show=definitions is specified.
				S		Shows that branch distance specifier S is selected.
				B		Shows that branch distance specifier B is selected.
				W		Shows that branch distance specifier W is selected.
				A		Shows that branch distance specifier A is selected.
					*	Shows that a substitute instruction is selected for a conditional branch instruction.
(4)	Source information (SOURCE STATEMENT) Contents of the assembly-language source file.					
(5)	C/C++ source line number (LineNo.)					
(6)	C/C++ source statement (C-SOURCE STATEMENT) C/C++ source statement output when the -show=source option is specified.					

3.1.3 Statistics Information

The following figure shows an example of statistics information output.

Information List (1)			
TOTAL ERROR(S)	00000		
TOTAL WARNING(S)	00000		
TOTAL LINE(S)	00071	LINES	
Section List (2)			
Attr	Size		Name
CODE	0000000047(0000002FH)	P	
ROMDATA	0000000004(00000004H)	D	

Item Number	Description
(1)	Numbers of error messages and warning messages, and total number of source lines
(2)	Section information (section attribute, size, and section name)

3.1.4 Compiler Command Specification Information

The file names and options specified on the command line when the compiler is invoked are output. The compiler command specification information is output at the beginning of the list file. The following figure shows an example of command specification information output.

```
;*** CPU TYPE *** (1)
;-ISA=RXV1
;*** COMMAND PARAMETER *** (2)
;-output=src=C:\tmp\elp1894\sample.src
;-nologo
;-show=source
;sample.c
```

Item Number	Description
(1)	Selected microcomputer
(2)	File names and options specified for the compiler

3.1.5 Assembler Command Specification Information

The file names and options specified on the command line when the assembler is invoked are output. The assembler command specification information is output at the end of the list file. The following figure shows an example of command specification information output.

```
Cpu Type (1)
-ISA=RXV1
Command Parameter (2)
-output=sample.obj
-nologo
-listfile=sample.lst
```

Item Number	Description
(1)	Microcomputer selected for the assembler
(2)	File names and options specified for the assembler

3.2 Link Map File

This section explains the link map file.

The link map has information of the link result. It can be referenced for information such as the section's allocation addresses.

3.2.1 Structure of Linkage List

Table 3-3 shows the structure and contents of the linkage list.

Table 3.2 Structure and Contents of Linkage List

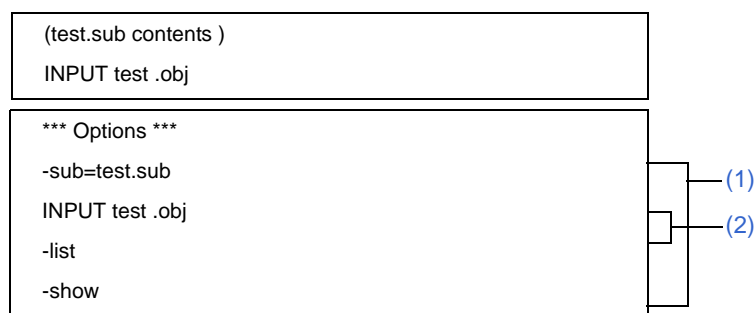
No	Output Information	Contents	When -show Option- Note is Specified	When -show Option is not Specified
1	Option information	Option strings specified by a command line or subcommand	None	Output

No	Output Information	Contents	When -show Option- Note is Specified	When -show Option is not Specified
2	Error information	Error messages	None	Output
3	Linkage map information	Section name, start/end addresses, size, and type	None	Output
4	Symbol information	Static definition symbol name, address, size, and type in the order of address	-show =symbol	Not output
		When -show =reference is specified: Symbol reference count and optimization information in addition to the above information	-show =reference	Not output
5	Symbol deletion optimization information	Symbols deleted by optimization	-show =symbol	Not output
6	Cross-reference information	Symbol reference information	-show =xreference	Not output
7	Total section size	Total sizes of RAM, ROM, and program sections	-show =total_size	Not output
8	Vector information	Vector numbers and address information	-show =vector	Not output
9	CRC information	CRC calculation result and output addresses	None	Always output when the CRC option is specified

Note The **-show** option is valid when the **list** option is specified.

3.2.2 Option Information

The option strings specified by a command line or a subcommand file are output. The following figure shows an example of option information output when **rlink -subcommand=test.sub -list -show** is specified.



Item Number	Description
(1)	Outputs option strings specified by a command line or a subcommand in the specified order.
(2)	Subcommand in the test.sub subcommand file.

3.2.3 Error Information

Error messages are output. The following figure shows an example of error information output.

```
*** Error Information ***
** E0562310 (E) Undefined external symbol "strcmp" referred to in "test.obj" (1)
```

Item Number	Description
(1)	Outputs an error message.

3.2.4 Linkage Map Information

The start and end addresses, size, and type of each section are output in the order of address. The following figure shows an example of linkage map information output.

```
*** Mapping List ***
(1)          (2)      (3)      (4)  (5)
SECTION      START    END      SIZE  ALIGN
P
              00001000 00001000      1    1
C
              00001004 00001007      4    4
D_2
              00001008 000014dd     4d6    2
B_2
              000014de 000050b3     3bd6    2
```

Item Number	Description
(1)	Section name
(2)	Start address
(3)	End address
(4)	Section size
(5)	Section boundary alignment value

3.2.5 Symbol Information

When **-show=symbol** is specified, the addresses, sizes, and types of externally defined symbols or static internally defined symbols are output in the order of address. When **-show=reference** is specified, the symbol reference counts and optimization information are also output. The following figure shows an example of symbol information output.

```

*** Symbol List ***
SECTION=(1)
FILE=(2)      (3)      (4)      (5)
              START    END      SIZE
              (6)      (7)      (8)      (9)      (10)  (11)
              SYMBOL   ADDR     SIZE    INFO    COUNTS  OPT
SECTION=P
FILE=test.obj
      00000000      00000428      428
      _main
      00000000          2      func ,g          0
      _malloc
      00000000      32      func ,l          0
FILE=mvn3
      00000428      00000490      68
      $MVN#3
      00000428          0      none ,g          0

```

Item Number	Description
(1)	Section name
(2)	File name
(3)	Start address of a section included in the file indicated by (2) above
(4)	End address of a section included in the file indicated by (2) above
(5)	Section size of a section included in the file indicated by (2) above
(6)	Symbol name
(7)	Symbol address
(8)	Symbol size
(9)	Symbol type as shown below Data type: func: Function name data: Variable name entry: Entry function name none: Undefined (label, assembler symbol) Declaration type g: External definition l: Internal definition
(10)	Symbol reference count only when <code>-show=reference</code> is specified. * is output when <code>show=reference</code> is not specified.
(11)	Optimization information as shown below. ch: Symbol modified by optimization cr: Symbol created by optimization mv: Symbol moved by optimization

3.2.6 Symbol Deletion Optimization Information

The size and type of symbols deleted by symbol deletion optimization (`-optimize=symbol_delete`) are output. The following figure shows an example of symbol deletion optimization information output.

```
*** Delete Symbols ***
```

```
(1)          (2)    (3)
SYMBOL          SIZE    INFO
  _Version
                4    data ,g
```

Item Number	Description
(1)	Deleted symbol name
(2)	Deleted symbol size
(3)	Deleted symbol type as shown below Data type func: Function name data: Variable name Declaration type g: External definition l: Internal definition

3.2.7 Cross-Reference Information

The symbol reference information (cross-reference information) is output when **-show=xreference** is specified. The following figure shows an example of cross-reference information output.

```
*** Cross Reference List ***
(1)  (2)    (3)          (4)    (5)
No   Unit Name  Global.Symbol  Location  External Information
0001 a
      SECTION=P  _func
                        00000100
                        _func1
                        00000116
                        _main
                        0000012c
                        _g
                        00000136
      SECTION=B
                        _a
                        00000190  0001(00000140:P)
                                0002(00000178:P)
                                0003(0000018c:P)
0002 b
      SECTION=P
                        _func01
                        00000154  0001(00000148:P)
                        _func02
                        00000166  0001(00000150:P)
0003 c
      SECTION=P
                        _func03
                        00000184
```

Item Number	Description
(1)	Unit number, which is an identification number in object units

Item Number	Description
(2)	Object name, which specifies the input order at linkage
(3)	Symbol name output in ascending order of allocation addresses for every section
(4)	Symbol allocation address, which is a relative value from the beginning of the section when <code>-form=relocate</code> is specified
(5)	Address of an external symbol that has been referenced Output format: <Unit number> (<address or offset in section>:<section name>)

3.2.8 Total Section Size

The total sizes of ROM, RAM, and program sections are output. The following figure shows an example of total section size output.

```
*** Total Section Size ***
RAMDATA SECTION :    00000660 Byte(s)  (1)
ROMDATA SECTION  :    00000174 Byte(s)  (2)
PROGRAM SECTION  :    000016d6 Byte(s)  (3)
```

Item Number	Description
(1)	Total size of RAM data sections
(2)	Total size of ROM data sections
(3)	Total size of program sections

3.2.9 Vector Information

The contents of the variable vector table are output when `-show=vector` is specified. The following figure shows an example of vector information output.

```
*** Variable Vector Table List ***
(1)      (2)
NO.      SYMBOL/ADDRESS
0        $fdummy
1        $fa
2        00ff8800
3        $fdummy
:
<Omitted>
```

Item Number	Description
(1)	Vector number
(2)	Symbol. When no symbol is defined for the vector number, the address is output.

3.2.10 CRC Information

The CRC calculation result and output address are output when the CRC option is specified.

```

*** CRC Code ***
CODE      : cb0b      (1)
ADDRESS   : 00007ffe  (2)

```

Item Number	Description
(1)	CRC calculation result
(2)	Address where the CRC calculation result is output

3.3 Library List

This section covers the contents and format of the library list output by the optimizing linkage editor.

3.3.1 Structure of Library List

Table 3.4 shows the structure and contents of the library list.

Table 3.3 Structure and Contents of Library List

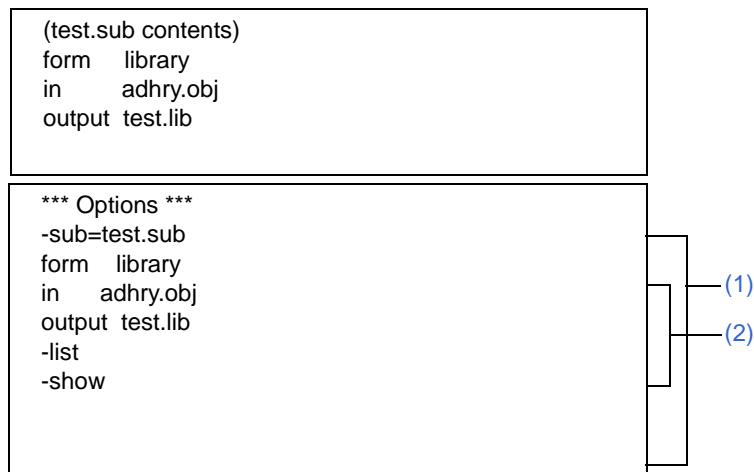
No	Output Information	Contents	Suboption ^{Note}	When -show Option is not Specified
1	Option information	Option strings specified by a command line or subcommand	-	Output
2	Error information	Error messages	-	Output
3	Library information	Library information	-	Output
4	Information of modules, sections, and symbols within library	Module within the library	-	Output
		When show=symbol is specified: List of symbol names in a module within the library	-show=symbol	Not output
		When show=section is specified: Lists of section names and symbol names in a module within the library	-show=section	Not output

Note All options are valid when the **-list** option is specified.

3.3.2 Option Information

The option strings specified by a command line or a subcommand file are output.

The following figure shows an example of option information output when **rlink -subcommand = test.sub -list -show** is specified.



Item Number	Description
(1)	Outputs option strings specified by a command line or a subcommand in the specified order.
(2)	Subcommand in the test.sub subcommand file.

3.3.3 Error Information

Messages for errors or warnings are output.
The following figure shows an example of error information output.

*** Error Information ***
** W0561200 (W) Backed up file "main.lib" into "main.lbk" (1)

Item Number	Description
(1)	Outputs a warning message.

3.3.4 Library Information

The library type is output.
The following figure shows an example of library information output.

*** Library Information ***
LIBRARY NAME =test.lib (1)
CPU=RX610 (2)
ENDIAN=Big (3)
ATTRIBUTE=system (4)
NUMBER OF MODULE =1 (5)

Item Number	Description
(1)	Library name
(2)	CPU name
(3)	Endian type

Item Number	Description
(4)	Library file attribute: either system library or user library
(5)	Number of modules within the library

3.3.5 Module, Section, and Symbol Information within Library

A list of modules within the library is output.

When **-show=symbol** is specified, the symbol names in a module within the library are listed. When **-show=section** is specified, the section names and symbol names in a module within the library are listed.

The following figure shows an output example of module, section, and symbol information within a library.

*** Library List ***	
(1)	(2)
MODULE	LAST UPDATE
(3)	
SECTION	
(4)	
SYMBOL	
adhry	29-Feb-2000 12:34:56
P	
_main	
_Proc0	
_Proc1	
C	
D	
_Version	
B	
_IntGlob	
_CharGlob	

Item Number	Description
(1)	Module name
(2)	Module definition date If the module is updated, the latest module update date is displayed.
(3)	Section name within a module
(4)	Symbol within a section

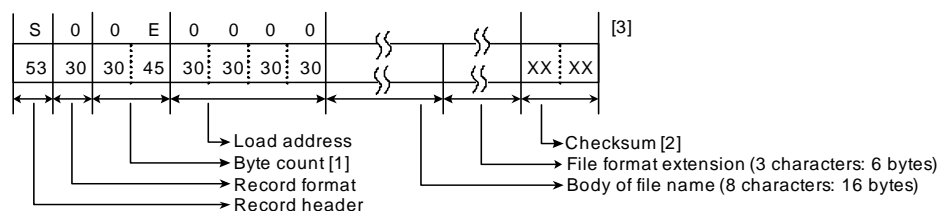
3.4 S-Type and HEX File Formats

This section describes the S-type files and HEX files that are output by the optimizing linkage editor.

3.4.1 S-Type File Format

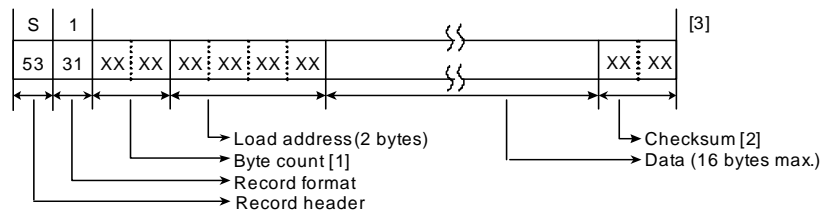
Figure 3.1 S-Type File Format

(a) Header record (S0 record)



(b) Data record (S1, S2, and S3 records)

(i) When the load address is 0 to FFFF



(ii) When the load address is 10000 to FFFFFF

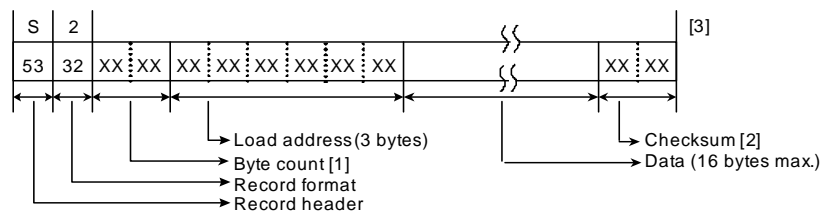
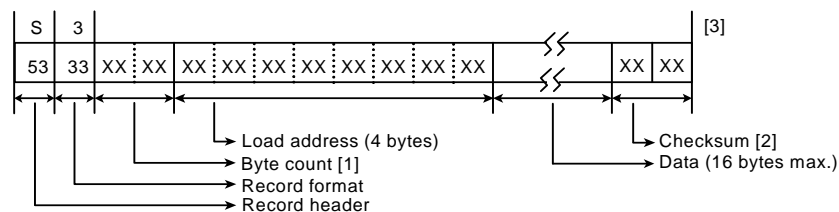


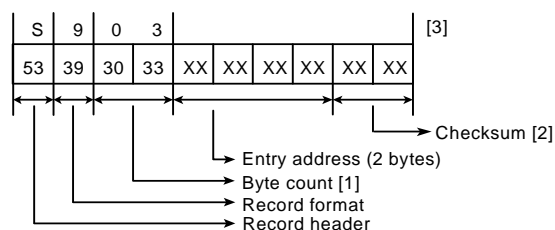
Figure 3.2 S-Type File Format (cont)

(iii) When the load address is 1000000 to FFFFFFFF

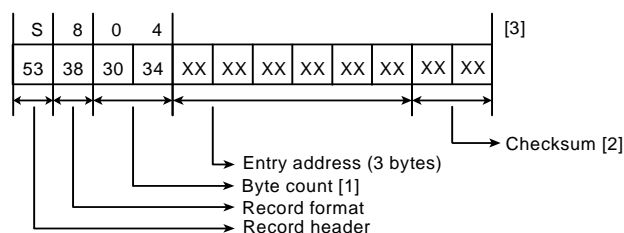


(c) End record (S9, S8, and S7 records)

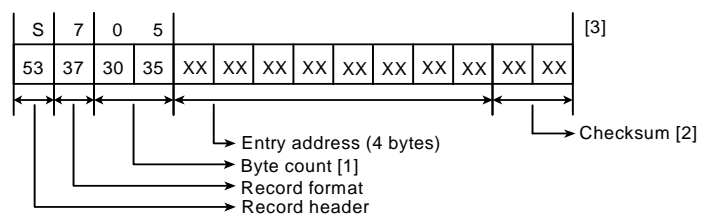
(i) When the entry address is 0 to FFFF



(ii) When the entry address is 10000 to FFFFFF



(iii) When the entry address is 1000000 to FFFFFFFF



Notes: [1] The number of bytes from the load address (or the entry address) to the checksum.

[2] 1's complement of the sum of the byte count and the data between the checksum and the byte count, in byte units.

[3] A new-line character is added immediately after the checksum.

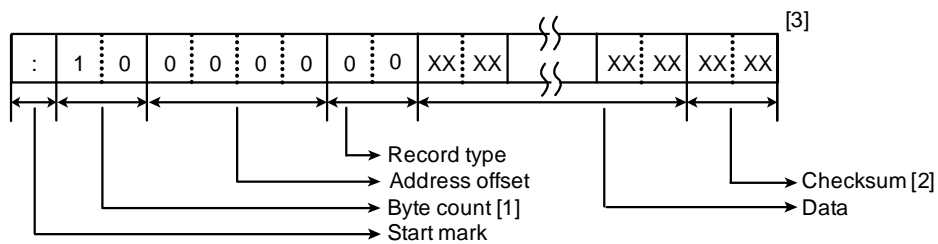
3.4.2 HEX File Format

The execution address of each data record is obtained as described below.

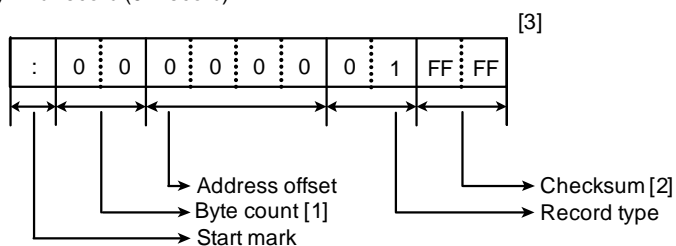
- (1) Segment address
(Segment base address << 4) + (Address offset of the data record)
- (2) Linear address
(Linear base address << 16) + (Address offset of the data record)

Figure 3.3 HEX File Format

(a) Data record (00 record)



(b) End record (01 record)



(c) Expansion segment address record(02 record)

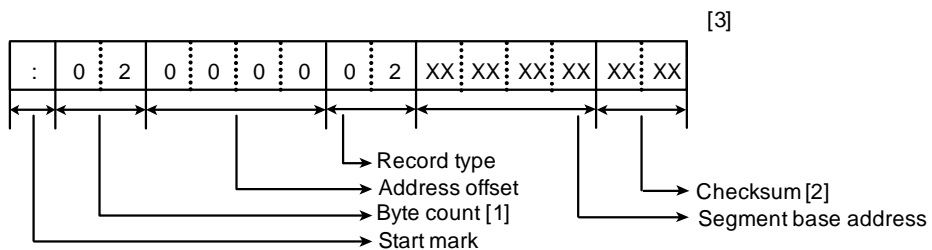
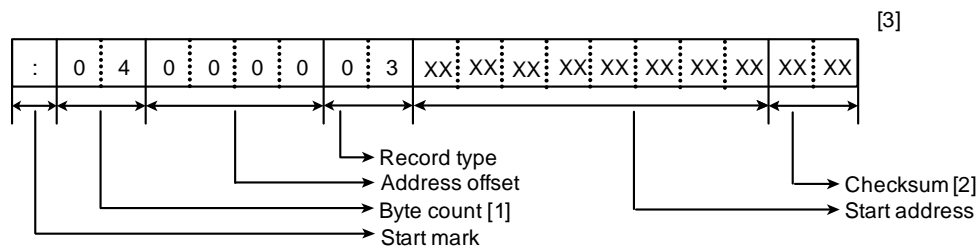
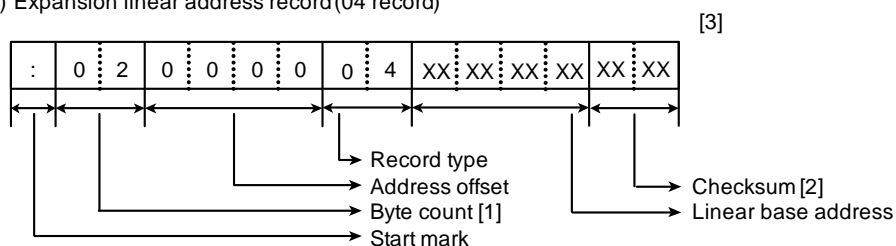


Figure 3.4 HEX File Format (cont)

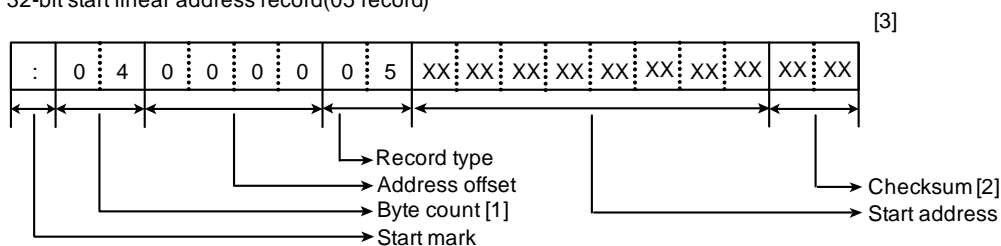
(d) Start address record (03 record)



(e) Expansion linear address record (04 record)



(f) 32-bit start linear address record (05 record)



Notes: [1] The number of bytes from the byte following the record type to the previous byte of the checksum.

[2] 2's complement of the sum of the byte count and the data between the byte count and checksum, in hexadecimal (lower 8 bits are valid).

[3] A new-line character is added immediately after the checksum

A. WINDOW REFERENCE

This appendix explains windows/panels/dialog boxes used in build process.

A.1 Description

The following lists the windows/panels/dialog boxes used in build process.

Table A.1 List of Windows/Panels/Dialog Boxes

Window/Panel/Dialog Box Name	Function Description
Main window	This is the first window to be open when CubeSuite+ is launched.
Project Tree panel	This panel is used to display the project components in tree view.
Property panel	This panel is used to display the detailed information on the build tool, file, or category that is selected on the Project Tree panel and change the settings of the information.
Editor panel	This panel is used to display/edit text files/source files.
Output panel	This panel is used to display the message that is output from the build tool.
Add File dialog box	This dialog box is used to create a new file and add it to the project.
Add Folder and File dialog box	This dialog box is used to add existing files and folder hierarchies to the project.
Character String Input dialog box	This dialog box is used to input and edit characters in one line.
Text Edit dialog box	This dialog box is used to input and edit texts in multiple lines.
Path Edit dialog box	This dialog box is used to edit or add the path.
System Include Path Order dialog box	This dialog box is used to refer the system include paths specified for the compiler and set their specified sequence.
Save Settings dialog box	This dialog box is used to set the encoding and newline code of the file that is editing on the Editor panel .
Link Order dialog box	This dialog box is used to display object module files to input to the linker and configure these link order.
Build Mode Settings dialog box	This dialog box is used to add and delete build modes and configure the current build mode in batch.
Batch Build dialog box	This dialog box is used to do build, rebuild and clean process in batch with the build mode that each project has.
Progress Status dialog box	This dialog box is used to show how the process has been progressed.
Option dialog box	This dialog box is used to configure the CubeSuite+ environment.
Add Existing File dialog box	This dialog box is used to select existing files to add to projects.
Import Build Options dialog box	This dialog box is used to select the target project file for import the build options.
Browse For Folder dialog box	This dialog box is used to select a folder and retrieve it for the caller.
Add Excluding File dialog box	This dialog box is used to select the file that will not be checked against the MISRA-C:2004 rules and set it to the area that this dialog box is called from.
Section Settings dialog box	This dialog box is used to add, change, or delete sections.

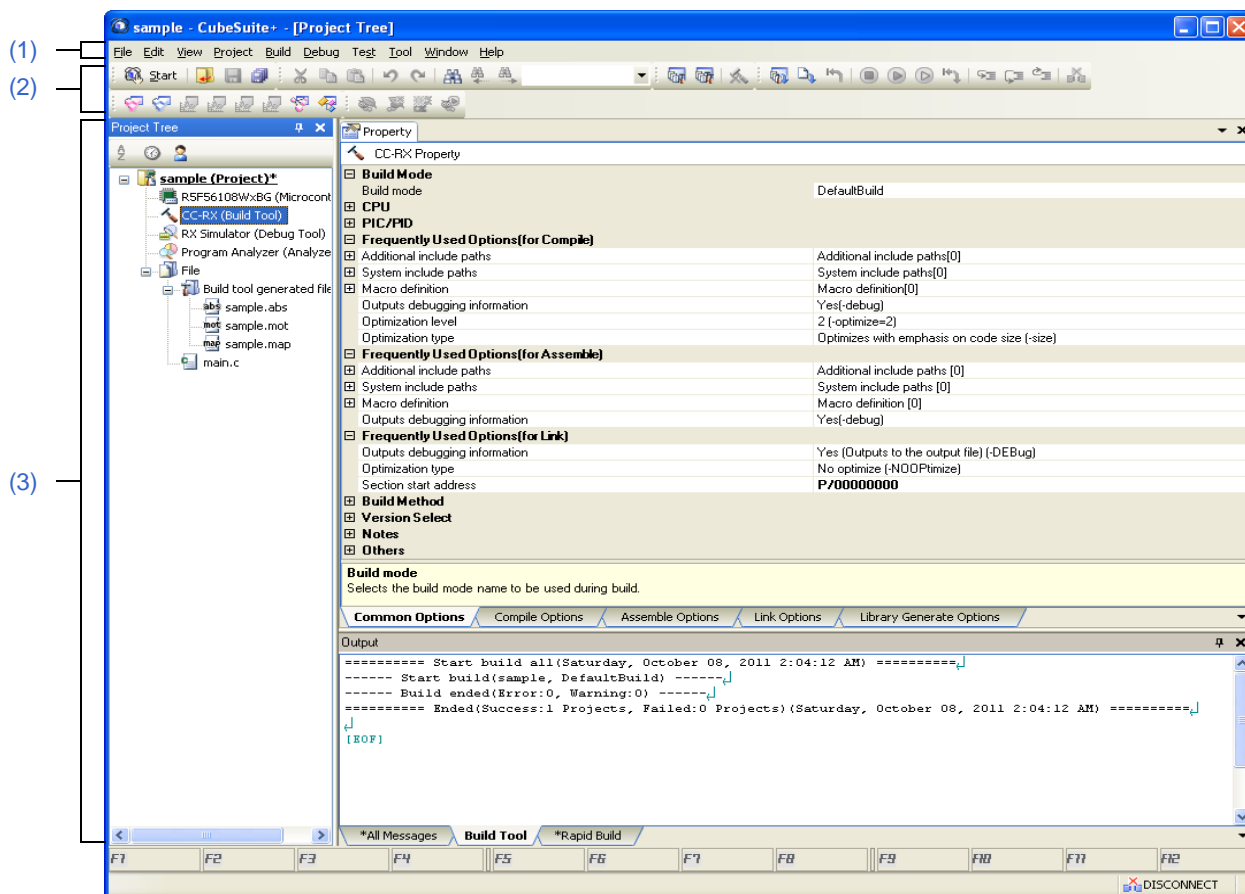
Window/Panel/Dialog Box Name	Function Description
Add Section, Modify Section, and Add Overlay dialog boxes	These dialog boxes are used to enter a new section name when adding, modifying, or overlaying a section, respectively.
Section Address dialog box	This dialog box is used to enter an address.
Unassigned Section dialog box	This dialog box is used to select sections to be deleted.
Specify The Predefined Macro dialog box	This dialog box is used to select the predefined macros to disable and specify it to the area that this dialog box is called from.
Specify the rule number dialog box	This dialog box is used to select numbers of the MISRA-C: 2004 rules and specify it to the area that this dialog box is called from.
Specify Misra2004 Rule File dialog box	This dialog box is used to select the MISRA-C: 2004 rule file and set it to the area that this dialog box is called from.
Add Excluding File dialog box	This dialog box is used to select the files that will not be checked against the MISRA-C: 2004 rules and set it to the area that this dialog box is called from.
Specify Using Library File dialog box	This dialog box is used to select the library file and set it to the area that this dialog box is called from.
Save As dialog box	This dialog box is used to save the editing file or contents of each panel to a file with a name.
Open with Program dialog box	This dialog box is used to select the application to open the file.
Select Import File dialog box	This dialog box is used to select a link order specification file to import to the Link Order dialog box .
Select Export File dialog box	This dialog box is used to generate a link order specification file.

Main window

This is the first window to be open when CubeSuite+ is launched.

This window is used to control the user program execution and open panels for the build process.

Figure A.1 Main Window



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)

[How to open]

- Select Windows® [start] >> [All programs] >> [Renesas Electronics CubeSuite+] >> [CubeSuite+]

[Description of each area]

- (1) Menu bar
Displays the menu relates to build.
- (a) [Project]
The [Project] menu shows menu items to operate the project and others.

Add New Subproject...	Closes the current project and opens the Create Project dialog box to create a new project. If the currently open project or file has been modified but it has not been saved yet, a confirmation message is displayed to ask you whether you want to save it.
-----------------------	---

Open Project...	Closes the current project and opens the Open Project dialog box to open the existing project. If the currently open project or file has been modified but it has not been saved yet, a confirmation message is displayed to ask you whether you want to save it.
Favorite Projects	Displays a cascading menu to use to open or save your favorite project.
1 <i>path</i>	[Opens your favorite project registered with [Favorite Projects] >> [1 Register to Favorite Project]. If no project has been registered, "Favorite Project" is displayed.
2 <i>path</i>	[Opens your favorite project registered with [Favorite Projects] >> [2 Register to Favorite Project]. If no project has been registered, "Favorite Project" is displayed.
3 <i>path</i>	[Opens your favorite project registered with [Favorite Projects] >> [3 Register to Favorite Project]. If no project has been registered, "Favorite Project" is displayed.
4 <i>path</i>	[Opens your favorite project registered with [Favorite Projects] >> [4 Register to Favorite Project]. If no project has been registered, "Favorite Project" is displayed.
1 Register to Favorite Project	The current project path is added to [1 <i>path</i>] in [Favorite Projects].
2 Register to Favorite Project	The current project path is added to [2 <i>path</i>] in [Favorite Projects].
3 Register to Favorite Project	The current project path is added to [3 <i>path</i>] in [Favorite Projects].
4 Register to Favorite Project	The current project path is added to [4 <i>path</i>] in [Favorite Projects].
Add	Shows the cascading menu to add subprojects to the project.
Add Subproject...	Opens the Add Existing Subproject dialog box to add an existing subproject to the project.
Add New Subproject...	Opens the Create Project dialog box to add a new subproject to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the file to the project. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new category node to the root of the File node. This allows the category name to be changed. The default category name is "New category". The new category name can be changed to the same name as the existing category node. Note that this menu is disabled when the build tool is in operation.
Sets <i>selected project or sub-project</i> as Active Project.	Set the selected project or subproject as an active project.
Close Project	Closes the current project. If the currently open project or file has been modified but it has not been saved yet, a confirmation message is displayed to ask you whether you want to save it.
Save Project	Saves the configuration information of the current project to the project file.

Save Project As...	Opens the Save Project As dialog box to save the configuration information of the current project to the project file with another name.
Remove from Project	Removes the selected project or subproject from the project. The subproject files or the file themselves are not deleted from the file system.
Save Project and Development Tools as Package...	Saves a set of this product and the project by copying them in a folder.

(b) [Build]

The [Build] menu shows menu items for the build process and others.



Build Project	Builds the project. The subproject is also built when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Rebuild Project	Rebuilds the project. The subproject is also rebuilt when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Clean Project	Cleans the project. The subproject is also cleaned when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Rapid Build	Toggles the rapid build function between enabled (default) and disabled.
Update Dependencies	Updates the dependency of the file in the project to build. The dependency of the file in the subproject to build is also updated when the subproject is added to the project.
Build <i>active project</i>	Builds the active project. If the active project is the main project, its subproject is not built. Note that this menu is disabled when the build tool is in operation.
Rebuild <i>active project</i>	Rebuilds the active project. If the active project is the main project, its subproject is not rebuilt. Note that this menu is disabled when the build tool is in operation.
Clean <i>active project</i>	Cleans the active project. If the active project is the main project, its subproject is not cleaned. Note that this menu is disabled when the build tool is in operation.
Update Dependencies of <i>active project</i>	Updates the dependency of the file in the active project to build.
Stop Build	Cancels the build, rebuild, batch build and clean operation.
Build Mode Settings...	Opens the Build Mode Settings dialog box to modify and add to the build mode.
Batch Build...	Opens the Batch Build dialog box to batch build.
Build Option List	Lists the currently set build option in the Output panel .


(2) Toolbar

Buttons used in build process are displayed.

(a) Build toolbar

Build toolbar shows buttons used in build process.

	Builds projects. The subproject is also built when it is added in the project. Note that this button is disabled when the build tool is in operation.
	Rebuilds projects. The subproject is also rebuilt when it is added in the project. Note that this button is disabled when the build tool is in operation.

	Cancels the build, rebuild, batch build and clean in operation.
---	---

- (3) Panel display area
The following panels are displayed in this area.

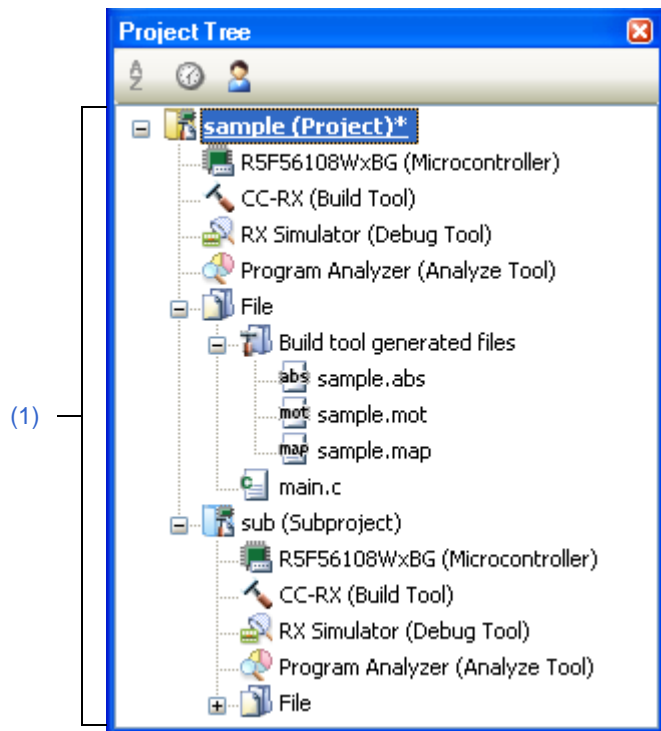
- [Project Tree panel](#)
- [Property panel](#)
- [Editor panel](#)
- [Output panel](#)

See the each panel section for details of the contents of the display.

Project Tree panel

This panel is used to display the project components such as the build tool, source files, etc. in tree view.

Figure A.2 Project Tree Panel



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[\[Edit\] menu \(only available for the Project Tree panel\)\]](#)
- [\[Context menu\]](#)

[How to open]

- From the [View] menu, select [Project Tree].

[Description of each area]

- (1) Project tree area
Project components are displayed in tree view with the following given node.

Node	Description
<i>Project name</i> (Project) (hereafter referred to as "Project node")	Project name.
<i>Build tool name</i> (Build tool) (hereafter referred to as "Build tool node")	The build tool (compiler, assembler, etc.) used in the project.

Node	Description
File (hereafter referred to as "File node")	<p>The following files that are added to the project are displayed under the root of this node.</p> <ul style="list-style-type: none"> - C source file (*.c) - C++ source file (*.cpp, *.cc, *.cp) - Assembler source file (*.src) - Header file (*.h, *.hpp, *.inc) - Object module file (*.obj) - Library file (*.lib) - Other file (doc, xml, etc.)
Build tool generated files (hereafter referred to as "Build tool generated files node")	<p>The following files generated by the build tool appear directly below the node generated during the build.</p> <ul style="list-style-type: none"> - For other than library projects <ul style="list-style-type: none"> Load module file (*.abs) Hex file (*.hex) S record file (*.mot) Binary data file (*.bin) Map file (*.map) - For library projects <ul style="list-style-type: none"> Library file (*.lib) Relocatable module file (*.rel) Map file (*.map, *.lbp) <p>Files displayed under this node cannot be renamed, deleted, or moved. This node is always placed lower than the File node. This node will no longer appear if you reload the project after building.</p>
Startup (hereafter referred to as "Startup node")	<p>This is a node for adding other than standard startup files to the project. This node is always placed lower than the File node.</p>
Category name (hereafter referred to as "category node")	<p>Categories that the user created to categorize files (see "2.3.4 Classify a file into a category"). This node is always placed lower than the File node.</p>
<i>Subproject name</i> (Subproject) (hereafter referred to as "Subproject node")	Subprojects added to the project.

When each component (the node or file) is selected, the detailed information (property) is displayed in the [Property panel](#). You can change the settings.

Remark When more than one components are selected, only the tab that is common to all the components is displayed.
When multiple files are selected and the values of their common properties are different, then the corresponding value fields are displayed blank.

This area has the following functions.

- (a) Add files
You can add files by one of the following procedure.
The files are added under the File node.

<1> Add existing files

- Select either one of the Project node, Subproject node, File node or a file. Then select [Add] >> [Add File...] from the [File] menu. The [Add Existing File dialog box](#) appears. Select files to add.
- Select either one of the Project node, Subproject node, File node or a file. Then select [Add] >> [Add File...] from the context menu. The [Add Existing File dialog box](#) appears. Select files to add.
- Copy the file using windows explorer and the like and then point the mouse to this area. Select [Paste] from the [Edit] menu.
- Drag files using windows explorer and the like and then drop them at the location in this area where you want to add the files to.

Remark If the files are dragged from the windows explorer and the like and then dropped in the blank space under the lower project tree, it is regarded as dropped in the Main project.

<2> When new files are added

- Select either one of the Project node, Subproject node, File node or a file. Then select [Add] >> [Add New File...] from the [File] menu. The [Add File dialog box](#) appears. Designate the file to create.
- Select either one of the Project node, Subproject node, File node or a file. Then select [Add] >> [Add New File...] from the context menu. The [Add File dialog box](#) appears. Designate the file to create.

Remark A blank file is created at the location designated in the [Add File dialog box](#).

(b) Remove the file from a project

You can remove files from the project by one of the following procedure.
The removed files are not deleted from the file system in this operation.

- Select the file you want to remove from the project. Then select [Remove from Project] from the [Project] menu.
- Select the file you want to remove from the project. Then select [Remove from Project] from the context menu.

(c) Move files

You can move files by the following procedure.
The file are moved under the File node.

- Drag the file you want to move and then drop it in the destination.

Remarks 1. Individual option is retained when the file is dropped in the main project or subproject.

Remarks 2. The file is copied, not moved when the file is dropped between the different project, or in the main project or subproject in same project. Note that this operation does not retain the individual option set in each file.

(d) Add categories

You can add the category node by one of the following procedure.
The category node are added under the File node.

- Select [Add New Category] from the [Project] menu.
- Select [Add New Category] from the context menu of either one of the Project node, Subproject node, or File node.

Remarks 1. The default category name is "New category".

Remarks 2. The new category name can be changed to the same name as the existing category node.

(e) Move categories

You can move the category node by the following procedure.
The category node are moved under the File node.

- Drag the category node you want to move and then drop it in the destination.

Remarks 1. Individual option set in the file in the category node is retained when the category node is dropped in the main project or subproject.

Remarks 2. The category node is copied, not moved when the it is dropped between the different project, or in the main project or subproject in same project. Note that the individual option set in each file contained in the category node is not retained.

(f) Add folders

You can add folders from Explorer or the like by the following procedure.

The folders are added under the File node.

The folders are added as categories.

- Drag the folder from Explorer or the like, and drop it over its destination. The [Add Folder and File dialog box](#) opens. Specify the file types and subdirectory levels in the folder to add.

Caution You cannot drag and drop folders and files into this area simultaneously.

(g) Modify the display order of the subprojects placed in order of build

The subproject is displayed in order of build from the top. Therefore, the order of build can be changed by changing the display order of the subprojects.

The project must be built from the subproject then the main project.

(h) Configure the standard build option

When the standard build option is changed, the property is displayed in boldface in the [Property panel](#).

You can change the standard build option to the current setting (cancel boldface) by the following procedure.

- Select the Build tool node and then select [Set to Default Build Option for Project] in the context menu.


Remark The configuration of the standard build option takes effect to the whole project (main project and subproject).










(i) Sort files and categories

You can sort files and category nodes in order of the file name, time stamp, or the user definition by the following procedure.

- Select one of the buttons in the toolbar.



The following table explains the buttons.

 is selected default by default.

Button	Description
	Sorts files and category nodes in order of their names.  : Ascending order  : Descending order  : Ascending order
	Sorts files and category nodes in order of their time stamp.  : Descending order  : Ascending order  : Descending order
	Sorts files and category nodes in order of the user definition (default). You can change the display order by dragging and dropping the file and category node.



(j) Display the file while editing

When the file added to the project is edited in the [Editor panel](#) and the file is not saved once, the file name is followed by "**". When the file is saved, "**" is deleted.

The file that is saved	 main.c
The file that is not saved after editing	 main.c*

(k) Display the source file in boldface that the individual build option is set


The source file icon whose option is different from the project general option (individual compile option, individual assemble option) is changed to a different one from the normal icon.

The file with project general option	 main.c
The file with individual build option	 main.c

(l) Highlight the file with read-only attribute

The read-only file added to the project is displayed in italic.

The file without read-only attribute	 main.c
--------------------------------------	--




The file with read-only attribute	 main.c
-----------------------------------	--

- (m) Highlight the file that does not exist
The file that is added to the project but does not exist is grayed out and its icon is dimmed.

The file that exists	 main.c
The file that does not exist	 main.c

- (n) Highlight the build-target file

- <1> The file which the error occurred during building (rapid building), rebuilding, compiling or assembling is highlighted as the example below.

The file without errors or warnings	 main.c
The file with error	 main.c
The file with warning	 main.c

Remarks 1. The file with both the error and the warning is highlighted in red.



Remarks 2. The highlight is canceled when the build option (general option or individual option) or the build mode is changed.

- <2> The names of the following files are displayed in boldface.



- The source files that have not been compiled after edited
- The source files after cleaning has been executed
- The source files after build tool options have been changed
- The source files after any build mode has been changed

Remark The file names are all displayed in boldface right after the project is opened. The boldface display is canceled after building is executed.

- (o) Highlight non build-target file
The file that is set as non build-target is highlighted as shown in the example below.

Build-target file	 main.c
Non build-target file	 main.c

- (p) Highlight the project that has been changed
The file component that is added to the project and the property of the project component are changed, the project name is followed by "*" and is displayed in boldface.
The boldface is canceled when the project is saved.

The project that has not been changed	 sample (Project)
The project that has been changed	 sample (Project)*

- (q) Highlight the active project
The active projects is underlined.

Non-active project	 sample (Project)*
Active project	 <u>sample (Project)*</u>

- (r) Run the editor
Open the file with the specific extension in the [Editor panel](#). When an external text editor is specified to use in the [Option dialog box](#), open the file with the external text editor. Other files are opened with the application associated with the OS.

Caution The files with the extensions that are not associated with the OS are not displayed.

You can open the editor by one of the following procedure.

- Double click the file.
- Select the file and then select [Open] from the context menu.
- Select the file and then press the [Enter] key.

The files that can be opened in the [Editor panel](#) are as follows.

- C source file (*.c)
- C++ source file (*.cpp, *.cc, *.cp)
- Header file (*.h, *.hpp, *.inc)
- Assembler source file (*.src)
- Map file (*.map, *.lbp)
- Hex file (*.hex)
- Assemble list file (*.lst)
- S record file (*.mot)
- Text file (*.txt)

Remark You can use one of the methods below to open files other than those listed above in the [Editor panel](#).

- Drag the file and drop it into the [Editor panel](#).
- Select the file and then select [Open with Internal Editor...] from the context menu.

[[Edit] menu (only available for the Project Tree panel)]

Copy	Copies the selected file or category node to the clipboard. While editing the file name or the category name, the characters of the selection are copied to the clipboard. Note that this menu is only enabled when the file or category node is selected.
Paste	Inserts the contents of the clipboard directly below the selected node on the project tree. While editing the file name or the category name, insert the contents of the clipboard. Note that this menu is disabled when the contents of the clipboard exist in the same project, when multiple files and category nodes are selected, and when the build tool is in operation.
Rename	You can rename the selected project, subproject, file, and category node. Press the [Enter] key to confirm the rename. Press the [ESC] key to cancel. When the file is selected, the actual file name is also changed. When the selected file is added to other project, those file names are also changed. Note that this menu is only enabled when the project, subproject, file, and category node is selected. Note that rename is disabled when the build tool is in operation.

[Context menu]

- (1) When the Project node is selected

Build <i>active project</i>	Builds the active project. If the active project is the main project, its subproject is not built. Note that this menu is disabled when the build tool is in operation.
Rebuild <i>active project</i>	Rebuilds the active project. If the active project is the main project, its subproject is not rebuilt. Note that this menu is disabled when the build tool is in operation.
Clean <i>active project</i>	Cleans the active project. If the active project is the main project, its subproject is not cleaned. Note that this menu is disabled when the build tool is in operation.
Open Folder with Explorer	Opens the folder that contains the project file of the selected project with Explorer.
Add	Shows the cascading menu to add subprojects and files to the project.
Add Subproject...	Opens the Add Existing Subproject dialog box to add the selected subproject to the project.
Add New Subproject...	Opens the Create Project dialog box to add the created subproject to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the project. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new category node to the root of the File node. This allows the category name to be changed. Up to 200 characters can be specified. The default category name is "New category". The new category name can be changed to the same name as the existing category node. This menu is disabled while the build tool is running, and if categories are nested 20 levels.
Set <i>selected project</i> as Active Project	Sets the selected project to an active project.
Save Project and Development Tools as Package...	Saves a set of the this product and the project by copying them in a folder.
Paste	This menu is always disabled.
Rename	You can rename the selected project.
Property	Displays the selected project's property on the Property panel .

(2) When the Subproject node is selected

Build <i>active project</i>	Builds the active project. Note that this menu is disabled when the build tool is in operation.
Rebuild <i>active project</i>	Rebuilds the active project. Note that this menu is disabled when the build tool is in operation.
Clean <i>active project</i>	Cleans the active project. Note that this menu is disabled when the build tool is in operation.
Open Folder with Explorer	Opens the folder that contains the subproject file of the selected subproject with Explorer.
Add	Shows the cascading menu to add subprojects, files, and category nodes to the project.
Add Subproject...	Opens the Add Existing Subproject dialog box to add the selected subproject to the project. The subproject cannot be added to another subproject.
Add New Subproject...	Opens the Create Project dialog box to add the created subproject to the project. The subproject cannot be added to another subproject.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the project. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new category node to the root of the File node. This allows the category name to be changed. Up to 200 characters can be specified. The default category name is "New category". The new category name can be changed to the same name as the existing category node. This menu is disabled while the build tool is running, and if categories are nested 20 levels.
Set <i>selected subproject</i> as Active Project	Sets the selected subproject to an active project.
Remove from Project	Removes the selected subproject from the project. The subproject file itself is not deleted from the file system with this operation. When the selected subproject is the active project, it cannot be removed from the project. Note that this menu is disabled when the build tool is in operation.
Paste	This menu is always disabled.
Rename	You can rename the selected subproject.
Property	Displays the selected subproject's property on the Property panel .

(3) When the Build tool node is selected

Build Project	Builds the selected project (main project or subproject). The subproject is also built when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Rebuild Project	Rebuilds the selected project (main project or subproject). The subproject is also rebuilt when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Clean Project	Cleans the selected project (main project or subproject). The subproject is also cleaned when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Set to Default Build Option for Project	Sets the current build option to the standard option for the selected project. When the subproject is added, it is not set. When the build option that is different from the standard option is set, its property is displayed in boldface.
Import Build Options...	Opens the Import Build Options dialog box to import the build options from the selected project file. ^{Note}
Set Link Order...	Opens the Link Order dialog box to display object module files and library files and to setup their link order. Note that this menu is disabled when the build tool is in operation.
Property	Displays the selected build tool's property on the Property panel .

Note See "2.12.1 [Import the build options of other project](#)" for details about the import function of the build options.

(4) When the File node is selected

Add	Shows the cascading menu to add files and category nodes to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project. The file is added directly below this node. The added file can be opened with the application corresponds to the file extension. The file is added directly below this node.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the project. The file is added directly below this node. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new category node to the root of this node. You can rename the category. Up to 200 characters can be specified. The default category name is "New category". The new category name can be changed to the same name as the existing category node. This menu is disabled while the build tool is running, and if categories are nested 20 levels.
Remove from Project	This menu is always disabled.
Copy	This menu is always disabled.
Paste	Inserts the contents of the clipboard directly below this node. However, this menu is disabled when the contents of the clipboard exist in the same project.
Rename	This menu is always disabled.
Property	Displays the selected category node's property on the Property panel .

(5) When a file is selected

Compile	Compiles the selected C source file. Note that this menu is only displayed when a C source file (except for non build-target file) is selected. Note that this menu is disabled when the build tool is in operation.
Assemble	Assembles the selected assembler source file. Note that this menu is only displayed when an assembler source file (except for non build-target file) is selected. Note that this menu is disabled when the build tool is in operation.
Open	Opens the selected file with the application corresponds to the file extension (see "(r) Run the editor "). Note that this menu is disabled when multiple files are selected.
Open with Internal Editor...	Opens the selected file with the Editor panel . Note that this menu is disabled when multiple files are selected.
Open with Selected Application...	Opens the Open with Program dialog box to open the selected file with the designated application. Note that this menu is disabled when multiple files are selected.
Open Folder with Explorer	Opens the folder that contains the selected file with Explorer.
Add	Shows the cascading menu to add files and category nodes to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project. The file is added to the same level as the selected file.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the project. The file is added to the same level as the selected file. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new category node at the same level as the selected file. You can rename the category. Up to 200 characters can be specified. The default category name is "New category". The new category name can be changed to the same name as the existing category node. This menu is disabled while the build tool is running, and if categories are nested 20 levels.
Remove from Project	Removes the selected file from the project. The removed file is not deleted from the file system in this operation. Note that this menu is disabled when the build tool is in operation.
Copy	Copies the selected file to the clipboard. When the file name is in editing, the characters of the selection are copied to the clipboard.
Paste	This menu is always disabled.
Rename	You can rename the selected file. The actual file is also renamed. When the selected file is added to another projects, it is also renamed.
Property	Displays the selected file's property on the Property panel .

- (6) When the Build tool generated files node is selected

Property	Displays this node 's property on the Property panel .
----------	--

- (7) When the Startup node is selected

Add	Shows the cascading menu to add files and category nodes to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project. The file is added directly below this node. The added file can be opened with the application corresponds to the file extension.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the project. The file is added directly below this node. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new category node to the root of this node. You can rename the category. Up to 200 characters can be specified. The default category name is "New category". The new category name can be changed to the same name as the existing category node. This menu is disabled while the build tool is running, and if categories are nested 20 levels.
Remove from Project	This menu is always disabled.
Copy	This menu is always disabled.
Paste	Inserts the contents of the clipboard directly below this node. However, this menu is disabled when the contents of the clipboard exist in the same project.
Rename	This menu is always disabled.
Property	Displays this node 's property on the Property panel .

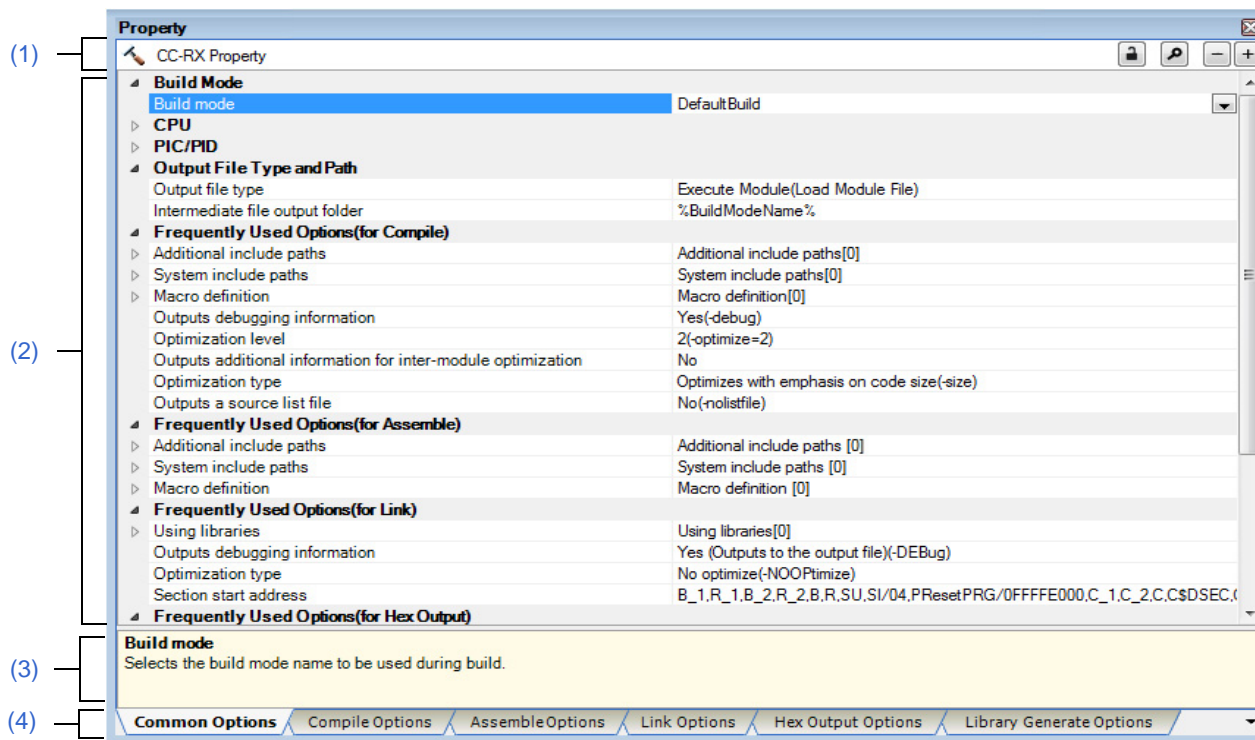
(8) When a category node is selected

Add	Shows the cascading menu to add files and category nodes to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project. The file is added directly below this node. The added file can be opened with the application corresponds to the file extension.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the project. The file is added directly below this node. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new category node to the root of this node. You can rename the category. Up to 200 characters can be specified. The default category name is "New category". The new category name can be changed to the same name as the existing category node. This menu is disabled while the build tool is running, and if categories are nested 20 levels.
Remove from Project	Removes the selected category node from the project. Note that this menu is disabled when the build tool is in operation.
Copy	Copies the selected category node to the clipboard. When the category name is in editing, the characters of the selection are copied to the clipboard.
Paste	Inserts the contents of the clipboard directly below this node. However, this menu is disabled when the contents of the clipboard exist in the same project. When the category name is in editing, the contents of the clipboard are inserted.
Rename	You can rename the selected category node.
Property	Displays the selected category node's property on the Property panel .

Property panel

In this panel, the detailed information on the node that is selected in the [Project Tree panel](#) is displayed categorized. Also, the settings of the selected node can be changed. This also shows the type of the [Generate Code] button clicked in the Code Generator panel and information about the file selected in the Code Generator Preview panel, and changes settings.

Figure A.3 Property Panel



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Dialog boxes opened from the Property panel\]](#)
- [\[\[Edit\] menu \(Property panel-dedicated items\)\]](#)
- [\[\[Help\] menu \(Property panel-dedicated items\)\]](#)
- [\[Context menu\]](#)

[How to open]

- Select either one of the Project node, Subproject node, Microcontroller node, Design Tool node, Build Tool node, Debug Tool node, Analyze Tool node, File node, or Category node in [Project Tree panel](#). Then select [Property] in [View] menu, or in the context menu.
- On the Code Generator panel, click [Generate Code] button, and then select [Property] from the [View] menu or context menu.
- On the Code Generator Preview panel, select file, and then select [Property] from the [View] menu or context menu.

Remark When either one of the Project node, Subproject node, Microcontroller node, Design Tool node, Build Tool node, Debug Tool node, Analyze Tool node, File node, or Category node is selected in [Project Tree panel](#) while the Property panel is open, the detailed information of the selected node is displayed.








[Description of each area]

(1) Name for the selected node area and button group

(a) Name for the selected node area



In this area, the name of the selected node in [Project Tree panel](#) is displayed.
This area is left blank when multiple nodes are selected.


(b) button group

 	<p>Locks properties to prevent the property value from being modified by wrong operation of a mouse or a keyboard.</p> <p>: locks properties. : unlocks properties.</p> <p>This button is only on the following tabs.</p> <ul style="list-style-type: none"> - Tabs displayed when the Build Tool node is selected - Tabs displayed when the C source file, C++ source file [CC-RX], assembly source file [CC-RH], assembler source file [CC-RX][CA850][CX][NC30][CA78K0R][CA78K0], object file [CC-RH], object module file [CC-RX][CA850][CX][NC30][CA78K0R][CA78K0], link directive file [CA850][CX][CA78K0R][CA78K0], symbol information file [CX], and library file is selected (except for [File Information] tab)
	<p>Opens the Find and Replace dialog box with the [Quick Find] tab target. Property names, property values, and active-help character strings can be found.</p>
	<p>Collapses the entire currently displayed detailed information display/change area.</p>
	<p>Expands the entire currently displayed detailed information display/change area.</p>

(2) Detailed information display/change area

In this area, the detailed information on the node that is selected in the [Project Tree panel](#) is displayed categorized in the list. Also, you can directly change the settings of the selected node. The node includes; Project node, Subproject node, Microcontroller node, Design Tool node, Build Tool node, Debug Tool node, Analyze Tool node, File node, and Category node.

The  mark indicates all the items in the category are expanded. The  mark indicates all the items are shrunk. You can expand/shrink the items by clicking these marks or double clicking the category name.

If the  mark is displayed, only the hex number is allowed in the text box.

Please see the information on each tab for the details of the display/setting in the category and its contents.

(3) Property description area

In this area, brief description of the categories and their contents selected in the Detailed information display/change area is shown.

(4) Tab selection area

Categories for the display of the detailed information are changed when a tab is selected.

In this panel, the following tabs are contained (see the section explaining each tab for the details on the contents of the display or the method of the setting on the tab).

Remark When more than two components are selected in [Project Tree panel](#), only the tab that is common to all the components is displayed. If the property is modified, that is taken effect to the selected components all of which are common to all.

[Dialog boxes opened from the Property panel]

The following dialog boxes are opened from the Property panel.

- Character String Input dialog box

See "[Character String Input dialog box](#)", "CubeSuite+ Integrated Development Environment User's Manual: Build", or "CubeSuite+ Integrated Development Environment User's Manual: Debug" for details.

- Text Edit dialog box

See "[Text Edit dialog box](#)", "CubeSuite+ Integrated Development Environment User's Manual: Build", or "CubeSuite+ Integrated Development Environment User's Manual: Debug" for details.

- Path Edit dialog box

See the "CubeSuite+ Integrated Development Environment User's Manual: Build" for details.

[[Edit] menu (Property panel-dedicated items)]

Undo	Undoes any property changes being done.
Cut	Cuts the selected text to the clip board while editing the property.
Copy	Copies the selected text in the property to the clip board.
Paste	Pastes the contents of the clip board to the property while editing the property.
Delete	Deletes the selected text while editing the property.
Select All	Selects all the text in the selected property while editing the property.
Find...	Opens the Find and Replace dialog box with the [Quick Find] tab target.

[[Help] menu (Property panel-dedicated items)]

Open Help for Property Panel	Displays the help of this panel.
------------------------------	----------------------------------

[Context menu]

Undo	Undoes any property changes being done.
Cut	Cuts the selected text to the clip board while editing the property.
Copy	Copies the selected text in the property to the clip board.
Paste	Pastes the contents of the clip board to the property while editing the property.
Delete	Deletes the selected text while editing the property.
Select All	Selects all the text in the selected property while editing the property.
Reset to Default	Restores the configuration of the selected item to default of the project default configuration. For [Individual Compile Options] tab and [Individual Assemble Options] tab, restores to the configuration of the general option.
Reset All to Default	Restores the configuration of the current tab to default of the project default configuration. For [Individual Compile Options] tab and [Individual Assemble Options] tab, restores to the configuration of the general option.

[Common Options] tab

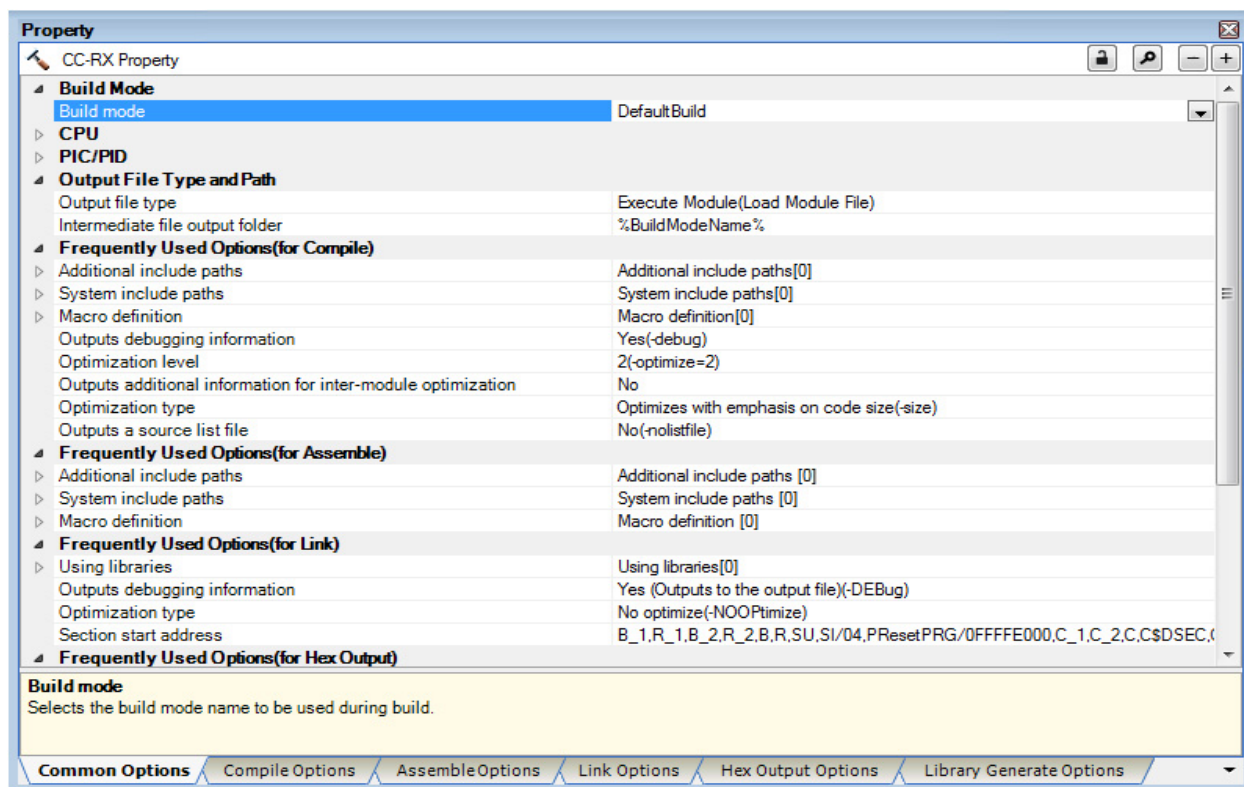
This tab shows the detailed information on the build tool categorized by the following and the configuration can be changed.

- (1) [Build Mode]
- (2) [CPU]
- (3) [PIC/PID]
- (4) [Output File Type and Path]
- (5) [Frequently Used Options(for Compile)]
- (6) [Frequently Used Options(for Assemble)]
- (7) [Frequently Used Options(for Link)]
- (8) [Frequently Used Options(for Hex Output)]
- (9) [Frequently Used Options(for Librarian)]
- (10) [Build Method]
- (11) [Version Select]
- (12) [Notes]
- (13) [Others]

Remark If the property in the [Frequently Used Options] category is changed, the value of the property having the same name contained in the corresponding tab will be changed accordingly.

Category from [Common Options] Tab	Corresponding Tab
[Frequently Used Options(for Compile)] category	[Compile Options] tab
[Frequently Used Options(for Assemble)] category	[Assemble Options] tab
[Frequently Used Options(for Link)] category	[Link Options] tab
[Frequently Used Options(for Hex Output)] category	[Hex Output Options] tab
[Frequently Used Options(for Librarian)] category	[Librarian Options] tab

Figure A.4 Property Panel: [Common options] Tab



[Description of each category]

(1) [Build Mode]

The detailed information on the build mode is displayed and the configuration can be changed.

Build mode	Selects the build mode to be used during build. Note that this property is not applied to [Reset All to Default] from the context menu.		
	Default	DefaultBuild	
	How to change	Select from the drop-down list.	
	Restriction	DefaultBuild	Builds with the default build mode that is set when a new project is created.
		Build mode that is added to the project (other than Default-Build)	Builds with the build mode that is added to the project (other than DefaultBuild).

(2) [CPU]

The detailed information on the CPU is displayed and the configuration can be changed.

Instruction-set architecture	Selects the instruction-set architecture. This corresponds to the -isa option of the compiler and library generator, -isa option of the assembler.		
	Default	The default value is set by selected device on creating project.	
	How to change	Select from the drop-down list.	
	Restriction	RXv1 architecture (-isa=rxv1)	Generates an instruction code for the RXv1 architecture.
		RXv2 architecture (-isa=rxv2)	Generates an instruction code for the RXv2 architecture.
		None	It allows the Microcontroller type setting.
Microcontroller type	Selects the microcontroller type. This corresponds to the -cpu option of the compiler and library generator, -cpu option of the assembler.		
	Default	RX600 series (-cpu=rx600)	
	How to change	Select from the drop-down list.	
	Restriction	RX600 series (-cpu=rx600)	Generates an instruction code for the RX600 Series.
		RX200 or RX100 series (-cpu=rx200)	Generates an instruction code for the RX200 Series.

Uses floating-point operation instructions	Selects whether to use floating-point operation instructions. This corresponds to the -fpu and -nofpu option of the compiler, and to the -fpu and -nofpu option of the assembler.		
	Default	<ul style="list-style-type: none"> - When [None] in the [Instruction-set architecture] property is selected: Depends on the Microcontroller type option - Other than above: The peculiar value for the target device 	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the Microcontroller type option	Depends on the Microcontroller type option. This item is not available when other than [None] in the [Instruction-set architecture] property is selected.
		Yes (-fpu)	Outputs an object that uses FPU instructions. This item is not available when [RX200 series (-cpu=rx200)] in the [Microcontroller type] property is selected.
		No (-nofpu)	Outputs an object that does not use FPU instructions.
Endian type for data	Selects endian type for data. This corresponds to the -endian option of the compiler and library generator, -endian option of assembler.		
	Default	Little-endian data (-endian=little)	
	How to change	Select from the drop-down list.	
	Restriction	Big-endian data (-endian=big)	Arranges data bytes in big endian.
		Little-endian data (-endian=little)	Arranges data bytes in little endian.
Rounding method for floating-point constant operations	Selects rounding method for floating-point constant operations. This option does not affect the method of rounding for floating-point operations during program execution. This corresponds to the -round option of the compiler and library generator.		
	Default	round to nearest (-round=nearest)	
	How to change	Select from the drop-down list.	
	Restriction	round to zero (-round=zero)	Rounds values to zero.
		round to nearest (-round=nearest)	Rounds values to the nearest value.

Handling of denormalized numbers in floating-point constants	Selects handling of denormalized numbers in floating-point constants. This option does not affect the method of rounding for floating-point operations during program execution. This corresponds to the -denormalize option of the compiler and library generator.		
	Default	Handles as zeros (-denormalize=off)	
	How to change	Select from the drop-down list.	
	Restriction	Handles as zeros (-denormalize=off)	Handles denormalized numbers as zero.
		Handles as they are (-denormalize=on)	Handles denormalized numbers as they are.
Precision of the double type and long double type	Selects precision of the double type and long double type. This corresponds to the -dbl_size option of the compiler and library generator.		
	Default	Handles in single precision (-dbl_size=4)	
	How to change	Select from the drop-down list.	
	Restriction	Handles in single precision (-dbl_size=4)	Handles the double type and long double type in single precision.
		Handles in double precision (-dbl_size=8)	Handles the double type and long double type in double precision.
Replaces the int type with the short type	Selects whether to replace the int type with the short type. This corresponds to the -int_to_short option of the compiler and library generator.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-int_to_short)	Replaces the int type with the short type and the unsigned int type with the unsigned short type.
		No	Does not replace the int type with the short type and the unsigned int type with the unsigned short type.
Sign of the char type	Selects sign of the char type with no sign specification. This corresponds to the -signed_char and -unsigned_char options of the compiler and library generator.		
	Default	Handles as unsigned char (-unsigned_char)	
	How to change	Select from the drop-down list.	
	Restriction	Handles as signed char (-signed_char)	Handles the char type as signed char.
		Handles as unsigned char (-unsigned_char)	Handles the char type as unsigned char.

Sign of the bit-field type	Selects sign of the bit-field type with no sign specification. This corresponds to the <code>-signed_bitfield</code> and <code>-unsigned_bitfield</code> options of the compiler and library generator.		
	Default	Handles as unsigned (<code>-unsigned_bitfield</code>)	
	How to change	Select from the drop-down list.	
	Restriction	Handles as signed (<code>-signed_bitfield</code>)	Handles the sign of a bit-field as signed.
		Handles as unsigned (<code>-unsigned_bitfield</code>)	Handles the sign of a bit-field as unsigned.
Selects the enumeration type size automatically	Selects whether to automatically select the enumeration type size. This corresponds to the <code>-auto_enum</code> option of the compiler and library generator.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (<code>-auto_enum</code>)	Processes the enumerated data qualified by enum as the minimum data type with which the enumeration value can fit in.
		No	Processes the enumeration type size as the signed long type.
Order of bit-field members	Selects order of bit-field members. This corresponds to the <code>-bit_order</code> option of the compiler and library generator.		
	Default	Allocates from right (<code>-bit_order=right</code>)	
	How to change	Select from the drop-down list.	
	Restriction	Allocates from left (<code>-bit_order=left</code>)	Allocates members from the upper bit.
		Allocates from right (<code>-bit_order=right</code>)	Allocates members from the lower bit.
Assumes the boundary alignment value for structure members is 1	Selects whether to assume the boundary alignment value for structure members is 1. This corresponds to the <code>-pack</code> and <code>-unpack</code> options of the compiler and library generator.		
	Default	No (<code>-unpack</code>)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (<code>-pack</code>)	Assumes the boundary alignment value for structure members is 1.
		No (<code>-unpack</code>)	Follows the boundary alignment.

Enables C++ exceptional handling function (try, catch and throw)	Selects whether to enable C++ exceptional handling function (try, catch and throw). This corresponds to the -exception and -noexception options of the compiler and library generator.		
	Default	No (-noexception)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-exception)	Enables the exception handling function.
		No (-noexception)	Disables the exception handling function.
Enables the C++ exceptional handling function (dynamic_cast and typeid)	Selects whether to enable the C++ exceptional handling function (dynamic_cast and typeid). This corresponds to the -rtti option of the compiler and library generator.		
	Default	No (-rtti=off)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-rtti=on)	Enables dynamic_cast and typeid.
		No (-rtti=off)	Disables dynamic_cast and typeid.
General registers used only in fast interrupt functions	Selects registers used only for fast interrupts. This corresponds to the -fint_register option of the compiler and library generator, -fint_register option of the assembler.		
	Default	None (-fint_register=0)	
	How to change	Select from the drop-down list.	
	Restriction	None (-fint_register=0)	No registers are used only for fast interrupts.
		R13 (-fint_register=1)	R13 is used only for fast interrupts.
		R12, R13 (-fint_register=2)	R13 and R12 are used only for fast interrupts.
		R11 to R13 (-fint_register=3)	R13 to R11 are used only for fast interrupts.
		R10 to R13 (-fint_register=4)	R13 to R10 are used only for fast interrupts.
Branch width size	Selects branch width size. This corresponds to the -branch option of the compiler and library generator.		
	Default	Compiles within 24 bits (-branch=24)	
	How to change	Select from the drop-down list.	
	Restriction	Compiles within 16 bits (-branch=16)	Compiles the program with a branch width within 16 bits.
		Compiles within 24 bits (-branch=24)	Compiles the program with a branch width within 24 bits.
		No specified (-branch=32)	Does not specify the branch width.

Base register for ROM	Specifies the general register used as a fixed base address throughout the program. When "base=rom=register A" is specified, accesses to const variables are all performed relative to the specified "register A". Note that the total size of the constant area section must be within 64 Kbytes to 256 Kbytes. This corresponds to the -base option of the compiler and library generator, -base option of the assembler. This property is displayed only when [No] in the [Enables the PID function] property in the [PIC/PID] category is selected.		
	Default	None	
	How to change	Select from the drop-down list.	
	Restriction	None	Does not specify the base register for ROM.
		R8 (-base=rom=R8)	Specifies R8 as the base register for ROM.
		R9 (-base=rom=R9)	Specifies R9 as the base register for ROM.
		R10 (-base=rom=R10)	Specifies R10 as the base register for ROM.
		R11 (-base=rom=R11)	Specifies R11 as the base register for ROM.
R12 (-base=rom=R12)		Specifies R12 as the base register for ROM.	
	R13 (-base=rom=R13)	Specifies R13 as the base register for ROM.	

Base register for RAM	Specifies the general register used as a fixed base address throughout the program. When "base=ram=register B" is specified, accesses to initialized variables and uninitialized variables are all performed relative to the specified "register B". Note that the total RAM data size must be within 64 Kbytes to 256 Kbytes. This corresponds to the -base option of the compiler and library generator, -base option of the assembler.	
	Default	None
	How to change	Select from the drop-down list.
	Restriction	None Does not specify the base register for RAM.
		R8 (-base=ram=R8) Specifies R8 as the base register for RAM.
		R9 (-base=ram=R9) Specifies R9 as the base register for RAM.
		R10 (-base=ram=R10) Specifies R10 as the base register for RAM.
		R11 (-base=ram=R11) Specifies R11 as the base register for RAM.
		R12 (-base=ram=R12) Specifies R12 as the base register for RAM.
		R13 (-base=ram=R13) Specifies R13 as the base register for RAM.
Address value of base register that sets the address value	Specifies the Address value of base register that sets the address value. This corresponds to the -base option of the compiler and library generator, -base option of the assembler.	
	Default	00000000 (hexadecimal number)
	How to change	Directly enter in the text box.
	Restriction	0 to FFFFFFFF (hexadecimal number)

Register of base register that sets the address value	Specifies the general register used as a fixed base address throughout the program. When " <i>address value=register C</i> " is specified, accesses to an area within 64 Kbytes to 256 Kbytes from the address value, among the areas whose addresses are already determined at the time of compilation, are performed relative to the specified " <i>register C</i> ". This corresponds to the -base option of the compiler and library generator, -base option of the assembler.	
	Default	None
	How to change	Select from the drop-down list.
	Restriction	None Does not specify the base register that sets the address value.
		R8 (-base=<address value>=R8) Specifies R8 as the base register that sets the address value.
		R9 (-base=<address value>=R9) Specifies R9 as the base register that sets the address value.
		R10 (-base=<address value>=R10) Specifies R10 as the base register that sets the address value.
		R11 (-base=<address value>=R11) Specifies R11 as the base register that sets the address value.
		R12 (-base=<address value>=R12) Specifies R12 as the base register that sets the address value.
		R13 (-base=<address value>=R13) Specifies R13 as the base register that sets the address value.
Avoids a problem specific to the CPU type	Selects avoid a problem specific to the CPU type. This corresponds to the -patch option of the compiler and library generator, -patch option of the assembler.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	No The code generated in response to the call by the intrinsic function <code>set_ipl</code> will contain the MVTIPL instruction.
		Yes(for RX610 Group) (-patch=rx610) Does not use the MVTIPL instruction in the generated code.
Saves and restores ACC using the interrupt function	Selects whether to save and restore Accumulator(ACC) using the interrupt function. The generated saved and restored code is the same code generated when acc is selected in #pragma interrupt. This corresponds to the -save_acc option of the compiler and library generator.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-save_acc) Saves and restores ACC using the interrupt function.
		No Does not save and restore ACC using the interrupt function.

(3) [PIC/PID]

The detailed information on the PIC/PID function is displayed and the configuration can be changed.

Enables the PIC function	Selects whether to enable the PIC(position independent code) function. In PIC, all function calls are performed with BSR or BRA instructions. When acquiring the address of a function, a relative address from the PC should be used. This allows PIC to be located at a desired address after linkage. This option corresponds to the -pic option of the compiler and library generator, -pic option of the assembler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-pic)	Enables the PIC function.
	No	Disables the PIC function.	
Enables the PID function	Selects whether to enable the PID(position independent data) function. The constant area sections C, C_2, and C_1, the literal section L, and the switch statement branch table sections W, W_2, and W_1 are handled as PID (position independent data). PID can be accessed through a relative address from the PID register. This corresponds to the -pid option of the compiler and library generator, the -pid option of the assembler. This property is displayed only when [None] in the [Base register for ROM] property in the [CPU] category, and [Yes] in the [Uses the PID register for code generation] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (The maximum bit width of the offset: 16 bits) (-pid=16)	Enables the PID function. 16-bit (64 Kbytes to 256 Kbytes) addressing mode is supported.
		Yes (The maximum bit width of the offset: No limitation) (-pid=32)	Enables the PID function. 32-bit (4 Gbytes) addressing mode is supported.
		No	Disables the PID function.
Uses the PID register for code generation	Selects whether to use the PID register for code generation. A master program called by an application program in which the PID function is enabled needs to be compiled/assembled with this option. This corresponds to the -nouse_pid_register option of the compiler and library generator, the -nouse_pid_register option of the assembler. This property is displayed only when [No] in the [Enables the PID function] property is selected.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Uses the PID register for code generation.
		No (-nouse_pid_register)	Does not use the PID register for code generation.

(4) [Output File Type and Path]

The detailed information on output file types and paths is displayed and the configuration can be changed.

Output file type	<p>Select the type of the file to be generated during a build. The file type set here will be the debug target. For other than the library project, only [Execute Module(Load Module File)] and [Execute Module(Hex File)] are displayed. For the library project, only [Library] is displayed.</p>		
	Default	<p>- For other than the library project Execute Module(Load Module File) - For the library project Library</p>	
	How to change	Select from the drop-down list.	
	Restriction	Execute Module(Load Module File)	Generates a load module file and hex file during a build. The load module file will be the debug target.
		Execute Module(Hex File)	Generates a load module file and hex file during a build. The hex file will be the debug target.
		Library	Generates a library file during a build.
Intermediate file output folder	<p>Specify the folder which the intermediate file is output. If a relative path is specified, the reference point of the path is the main project or sub-project folder. If an absolute path is specified, the reference point of the path is the main project or subproject folder (unless the drives are different). The following placeholder is supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. If this is blank, it is assumed that the project folder has been specified.</p>		
	Default	%BuildModeName%	
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.	
	Restriction	Up to 247 characters	

(5) [Frequently Used Options(for Compile)]

The detailed information on frequently used options during compilation is displayed and the configuration can be changed.

Additional include paths	<p>Specifies the name of the path to the folder that stores the include file.</p> <p>The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder.</p> <p>%ActiveProjectName%: Replaces with the active project name.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%MainProjectDir%: Replaces with the absolute path of the main project folder.</p> <p>%MainProjectName%: Replaces with the main project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.</p> <p>%ProjectDir%: Replaces with the absolute path of the project folder.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%TempDir%: Replaces with the absolute path of the temporary folder.</p> <p>%WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>The reference point of the path is the project folder.</p> <p>This corresponds to the -include option of the compiler.</p> <p>The specified include path is displayed as the subproperty.</p>	
	Default	Additional include paths[<i>number of defined items</i>]
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 247 characters. Up to 65536 items can be specified.
System include paths	<p>Changes the specified order of the include paths which the system set during compiling.</p> <p>This corresponds to the -include option of the compiler.</p>	
	Default	System include paths[<i>number of defined items</i>]
	How to change	Edit by the System Include Path Order dialog box which appears when clicking the [...] button.
	Restriction	Changes not allowed (Only the specified order of the include paths can be changed.)
Macro definition	<p>Specifies the macro name to be defined.</p> <p>Specify in the format of "<i>macro name=string</i>", with one macro name per line.</p> <p>The "<i>=string</i>" part can be omitted, and in this case, the macro name is assumed to be defined.</p> <p>This corresponds to the -define option of the compiler.</p> <p>The specified macro is displayed as the subproperty.</p>	
	Default	Macro definition[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.

Outputs debugging information	Selects whether to output debugging information to object module files. This corresponds to the -debug and -nodebug options of the compiler.		
	Default	No (-nodebug)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-debug)	Outputs debugging information to object module files.
		No (-nodebug)	Does not output debugging information to object module files.
Optimization level	Selects optimization level. This corresponds to the -optimize option of the compiler.		
	Default	2 (-optimize=2)	
	How to change	Select from the drop-down list.	
	Restriction	0 (-optimize=0)	Does not optimize the program.
		1 (-optimize=1)	Partially optimizes the program by automatically allocating variables to registers, integrating the function exit blocks, integrating multiple instructions which can be integrated, etc.
		2 (-optimize=2)	Performs overall optimization.
		Max (-optimize=max)	Performs optimization as much as possible.
Outputs additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. At linkage, inter-module optimization is applied to files for which this option has been specified. This corresponds to the -goptimize option of the compiler. This property is displayed only when [Object module file (-output=obj)] in the [Output file type] property in the [Object] category.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.
		No	Does not outputs additional information for inter-module optimization.
Optimization type	Selects optimization type. This corresponds to the -speed and -size option of the compiler.		
	Default	Optimizes with emphasis on code size (-size)	
	How to change	Select from the drop-down list.	
	Restriction	Optimizes with emphasis on execution performance (-speed)	Optimizes with emphasis on execution performance.
		Optimizes with emphasis on code size (-size)	Optimizes with emphasis on code size.

Outputs a source list file	Selects whether to output a source list file. This corresponds to the -listfile and -nolistfile option of the compiler.		
	Default	No (-nolistfile)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-lisfile)	Outputs a source list file.
		No (-nolistfile)	Disable output of a source list file.
Outputs the C/C++ source file	Specifies the contents of the source list file. Selects whether to output the C/C++ source file. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=source)	Outputs the C/C++ source file.
		No	Does not output the C/C++ source file.
Outputs the statements unsatisfied in conditional assembly	Specifies the contents of the source list file. Selects whether to output the statements unsatisfied in conditional assembly. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=conditionals)	Outputs the statements unsatisfied in conditional assembly.
		No	Does not output the statements unsatisfied in conditional assembly.
Outputs the information before .DEFINE replacement	Specifies the contents of the source list file. Selects whether to output the information before .DEFINE replacement. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=definitions)	Outputs the information before .DEFINE replacement.
		No	Does not output the information before .DEFINE replacement.

Outputs the assembler macro expansion statements	Specifies the contents of the source list file. Selects whether to output the assembler macro expansion statements. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=expansions)	Outputs the assembler macro expansion statements.
	No	Does not output the assembler macro expansion statements.	

(6) [Frequently Used Options(for Assemble)]

The detailed information on frequently used options during assembling is displayed and the configuration can be changed.

Additional include paths	<p>Specifies the name of the path to the folder that stores the include file. The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. The reference point of the path is the project folder. This corresponds to the -include option of the assembler. The specified include path is displayed as the subproperty.</p>	
	Default	Additional include paths[<i>number of defined items</i>]
	How to change	<p>Edit by the Path Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.</p>
	Restriction	<p>Up to 247 characters Up to 65536 items can be specified.</p>
System include paths	<p>Changes the specified order of the include paths which the system set during assembling. This corresponds to the -include option of the assembler.</p>	
	Default	System include paths[<i>number of defined items</i>]
	How to change	Edit by the System Include Path Order dialog box which appears when clicking the [...] button.
	Restriction	Changes not allowed (Only the specified order of the include paths can be changed.)

Macro definition	Specifies the macro name to be defined. Specifies in the format of " <i>macro name=string</i> ", with one macro name per line. This corresponds to the -define option of the assembler. The specified macro is displayed as the subproperty.		
	Default	Macro definition[<i>number of defined items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 32767 characters Up to 65536 items can be specified.	
Outputs debugging information	Selects whether to output debugging information to object module files. This corresponds to the -debug and -nodebug options of the assembler. This property is not displayed when [No] in the [Build simultaneously] property is selected.		
	Default	No (-nodebug)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-debug)	Outputs debugging information to object module files.
No (-nodebug)		Does not output debugging information to object module files.	
Output additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. At linkage, inter-module optimization is applied to files for which this option has been specified. This corresponds to the -goptimize option of the assembler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.
No		Does not output additional information for inter-module optimization.	
Outputs a assemble list file	Selects whether to output an assemble list file. This corresponds to the -listfile and -nolistfile option of the assembler. This property is displayed only when [No] in the [Build simultaneously] property is selected.		
	Default	No (-nolistfile)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-listfile)	Outputs an assemble list file.
No (-nolistfile)		Does not output an assemble list file.	

Outputs the state-ments unsatisfied in conditional assembly	Specifies the contents of the assemble list file. Selects whether to output the statements unsatisfied in conditional assembly. This corresponds to the -show option of the assembler. This property is displayed only when [Yes (-listfile)] in the [Output a assemble list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=con- ditionals)	Outputs the statements unsatisfied in conditional assembly.
	No	Does not output the statements unsatisfied in conditional assembly.	
Outputs the informa-tion before .DEFINE replacement	Specifies the contents of the assemble list file. Selects whether to output the information before .DEFINE replacement. This corresponds to the -show option of the assembler. This property is displayed only when [Yes (-listfile)] in the [Output a assemble list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=defi- nitions)	Outputs the information before replacement specified with .DEFINE.
	No	Does not output the information before replace-ment specified with .DEFINE.	
Outputs the assem- bler macro expansion statements	Specifies the contents of the assemble list file. Selects whether to output the assembler macro expansion statements. This corresponds to the -show option of the assembler. This property is displayed only when [Yes (-listfile)] in the [Output a assemble list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (- show=expan- sions)	Outputs the macro expansion statements.
	No	Does not output the macro expansion state-ments.	

(7) [Frequently Used Options(for Link)]

The detailed information on frequently used options during linking is displayed and the configuration can be changed.

This category is not displayed for the library project.

Using libraries	Specifies an input library file. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This corresponds to the -library option of the linker. The library file name is displayed as the subproperty.		
	Default	Input object module file[number of defined items]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 32767 characters Up to 65536 items can be specified.	
Outputs debugging information	Specifies whether debugging information is output. This corresponds to the -nodebug, -sdebug, and -debug options of the linker.		
	Default	Yes (Outputs to the output file) (-DEBug)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (Outputs to the output file) (-DEBug)	Outputs a debugging information to the output file.
		Yes (Outputs to <output file name>.dbg file) (-SDebug)	Outputs a debugging information to <output file name>.dbg file.
		No (-NODEBug)	Does not output a debugging information.
Optimization type	Specifies optimization type. This corresponds to the -nooptimize and -optimize options of the linker.		
	Default	No optimize (-NOOPTimize)	
	How to change	Select from the drop-down list.	
	Restriction	No optimize (-NOOPTimize)	Does not execute optimization for a module.
		All(-OPTimize)	Provides all optimizations.
		Speed-oriented optimization (-OPTimize=SPeed)	Provides optimization for speed.
		Safe optimization (-OPTimize=SAFE)	Provides safe optimization.
		Custom	Performs optimization for the specified options.

Deletes variables/ functions that are not referenced	Selects whether to delete variables/functions that are not referenced. This corresponds to the -optimize option of the linker. This property is displayed only when [Custom] in the [Optimization type] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-OPTi- mize=SYmbol_delete)	Deletes variables/functions that are not referenced.
		No	Does not delete variables/functions that are not referenced.
Creates a subroutine for the same instruc- tion sequence	Selects whether to create a subroutine for the same instruction sequence. This corresponds to the -optimize option of the linker. This property is displayed only when [Custom] in the [Optimization type] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-OPTi- mize=SAMe_code)	Creates a subroutine for the same instruction sequence.
		No	Does not create a subroutine for the same instruction sequence.
Minimum code size	Specifies the minimum code size for the optimization. This corresponds to the -samesize option of the linker. This property is displayed only when [Yes (-OPTimize=SAMe_code)] in the [Creates a subroutine for the same instruction sequence] property is specified.		
	Default	1E (hexadecimal number)	
	How to change	Directly enter in the text box.	
	Restriction	8 to 7FFF (hexadecimal number)	
Replaces an instruc- tion with a smaller-size instruction	Selects whether to replace an instruction with a smaller-size instruction. This corresponds to the -optimize option of the linker. This property is displayed only when [Custom] in the [Optimization type] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-OPTi- mize=SHort_format)	Replaces an instruction with a smaller- size instruction.
		No	Does not replace an instruction with a smaller-size instruction.

Optimizes branch instruction size	Selects whether to optimize branch instruction size. This corresponds to the -optimize option of the linker. This property is displayed only when [Custom] in the [Optimization type] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-OPTimize=Branch)	Optimizes branch instruction size according to program allocation information.
		No	Does not optimize branch instruction size.
Section start address	Specifies the section start address. This corresponds to the -start option of the linker.		
	Default	The peculiar value for the target device	
	How to change	Directly enter in the text box or edit by the Section Settings dialog box which appears when clicking the [...] button.	
	Restriction	Up to 1022 characters	

(8) [Frequently Used Options(for Hex Output)]

The detailed information on frequently used options during hex output is displayed and the configuration can be changed.

This category is not displayed for the application project.

Output hex file	Specifies whether hex file is output.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Outputs a hex file.
		No	Does not output a hex file.
Load module file convert format	Selects the load module file convert format. This corresponds to the -form option of the linker.		
	Default	S record file (-FOrm=Stype)	
	How to change	Select from the drop-down list.	
	Restriction	Intel expanded hex file (-FOrm=Hexadecimal)	Outputs a Intel expanded hex file.
		Motorola S type file (-FOrm=Stype)	Outputs a Motorola S-type file.
		Binary file (-FOrm=Binary)	Outputs a binary file.

Output folder	<p>Specifies path of the output folder.</p> <p>The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder.</p> <p>%ActiveProjectName%: Replaces with the active project name.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%MainProjectDir%: Replaces with the absolute path of the main project folder.</p> <p>%MainProjectName%: Replaces with the main project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.</p> <p>%OutputDir%: Replaces with the absolute path of the output folder.</p> <p>%OutputFile%: Replaces with the absolute path of the output file.</p> <p>%ProjectDir%: Replaces with the absolute path of the project folder.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%TempDir%: Replaces with the absolute path of the temporary folder.</p> <p>%WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>If this is blank, it is assumed that the project folder has been specified.</p> <p>This corresponds to the -output option of the linker.</p> <p>This property is displayed only when [Yes] in the [Output hex file] property is selected.</p>	
	Default	%BuildModeName%
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.
	Restriction	Up to 247 characters
Output file name	<p>Specifies the output file name.</p> <p>The default extensions depends on [Load module file convert format] property when extension omitted.</p> <p>The default extensions are as follows:</p> <p>"Hex file (-Form=Hexadecimal)" : .hex</p> <p>"S record file (-Form=Stype)" : .mot</p> <p>"Binary data file (-Form=Binary)" : .bin.</p> <p>The following placeholders are supported.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.</p> <p>This corresponds to the -output option of the linker.</p> <p>This property is displayed only when [Yes] in the [Output hex file] property is selected.</p>	
	Default	<p>- When [Hex file (-Form=Hexadecimal)] in the [Load module file convert format] property is selected</p> <p>%ProjectName%.hex</p> <p>- When [S record file (-Form=Stype)] in the [Load module file convert format] property is selected</p> <p>%ProjectName%.mot</p> <p>- When [Binary data file (-Form=Binary)] in the [Load module file convert format] property is selected</p> <p>%ProjectName%.bin</p>
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

Division output file	<p>Specifies the division conversion file.</p> <p>Specifies in the format of "file name=start address-end address" or "file name=section name", with one file name per line.</p> <p>Specifies an address in the hexadecimal notation (example: file2.mot=400-ffff).</p> <p>To define multiple sections, use a colon to separate each entry written, as in "file name=section name:section name" (example: file1.mot=stack:istack).</p> <p>The default extensions depends on [Load module file convert format] property when extension omitted.</p> <p>The default extensions are as follows:</p> <p>"Hex file (-Form=Hexadecimal)" : .hex</p> <p>"S record file (-Form=Stype)" : .mot</p> <p>"Binary data file (-Form=Binary)" : .bin.</p> <p>The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder.</p> <p>%ActiveProjectName%: Replaces with the active project name.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%MainProjectDir%: Replaces with the absolute path of the main project folder.</p> <p>%MainProjectName%: Replaces with the main project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.</p> <p>%OutputDir%: Replaces with the absolute path of the output folder.</p> <p>%OutputFile%: Replaces with the absolute path of the output file.</p> <p>%ProjectDir%: Replaces with the absolute path of the project folder.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%TempDir%: Replaces with the absolute path of the temporary folder.</p> <p>%WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>This corresponds to the -output option of the linker.</p> <p>This property is displayed only when [Yes] in the [Output hex file] property is selected.</p>	
	Default	Division conversion file[number of defined items]
	How to change	<p>Edit by the Text Edit dialog box which appears when clicking the [...] button.</p> <p>For the subproperty, you can use the text box directly enter the text.</p>
	Restriction	<p>Up to 255 characters</p> <p>Up to 65536 items can be specified.</p>

(9) [Frequently Used Options(for Librarian)]

The detailed information on frequently used options during library generating is displayed and the configuration can be changed.

This category is not displayed for the application project.

Outputs debugging information	Specifies whether debugging information is output. This corresponds to the -nodebug and -debug options of the linker. This property is displayed only when [Relocatble module file (-FOrm=Relocate)] in the [Output file type] property is selected.				
	Default	Yes (Outputs to the output file) (-DEBbug)			
	How to change	Select from the drop-down list.			
	Restriction	<table><tr><td>Yes (Outputs to the output file) (-DEBbug)</td><td>Outputs a debugging information to the output file.</td></tr><tr><td>No (-NODEBbug)</td><td>Does not output a debugging information.</td></tr></table>	Yes (Outputs to the output file) (-DEBbug)	Outputs a debugging information to the output file.	No (-NODEBbug)
Yes (Outputs to the output file) (-DEBbug)	Outputs a debugging information to the output file.				
No (-NODEBbug)	Does not output a debugging information.				

(10) [Build Method]

The detailed information on the build method is displayed and the configuration can be changed.

Build simultaneously	Select whether to generate the load module file by compiling/assembling/linking multiple files simultaneously. The files with the individual build options and files to be executed prior to the build are excluded from running a build simultaneously. See "2.16.5 Compile/assemble/link multiple files simultaneously" for details about running a build simultaneously.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Compiles, assembles, and links multiple files simultaneously.
		No	Compiles, assembles, and links for each file.
Handling the source file includes non-existing file	Selects whether to recompile/assemble the source file if there are no files that include it.		
	Default	Re-compile/assemble the source file	
	How to change	Select from the drop-down list.	
	Restriction	Re-compile/assemble the source file	Recompiles/assembles the source file if there are no files that include it.
		Ignore re-compiling/assembling the source file	Does not recompile/assemble the source file if there are no files that include it.
Ensure compatibility of paths and linkage order	Selects compatibility with High-performance embedded workshop about paths and linkage order.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Accords to the High-performance Embedded Workshop's linkage order.
		No	Accords to this product's path and linkage order.

(11) [Version Select]

The detailed information on the build tool version is displayed and the configuration can be changed.

Using compiler package install folder	The folder in which the compiler package to be used is installed is displayed.	
	Default	Install folder name
	How to change	Changes not allowed

Using compiler package version	Selects the version of the compiler package to be used. This setting is common to all the build modes.		
	Default	Always latest version which was installed	
	How to change	Select from the drop-down list.	
	Restriction	Always latest version which was installed	Uses the latest version in the installed compiler packages.
		Versions of the installed compiler packages	Uses the selected version in the compiler package.
Latest compiler package version which was installed	The version of the compiler package to be used when [Always latest version which was installed] is selected in the [Using compiler package version] property is displayed. This setting is common to all the build modes. This property is displayed only when [Always latest version which was installed] in the [Using compiler package version] property is selected.		
	Default	The latest version of the installed compiler packages	
	How to change	Changes not allowed	

(12) [Notes]

The detailed information on notes is displayed and the configuration can be changed.

Memo	Adds memos to the build tool. Adds one item in one line. This setting is common to all the build modes. The added memos are displayed as the subproperty.		
	Default	Memo[<i>number-of-items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 256 characters Up to 256 items can be specified.	

(13) [Others]

Other detailed information on the build tool are displayed and the configuration can be changed.

Output message format	<p>Specifies the format of the message being built. This applies to the messages output by the build tool to be used, and commands added by plugins. It does not apply to the output messages of commands specified in the [Commands executed before build processing] or [Commands executed after build processing] property. The following placeholders are supported. %Program%: Replaces with the program name under execution. %Options%: Replaces with the command line option under build execution. %TargetFiles%: Replaces with the file name being built. If this is blank, it is assumed that "%Program% %Options%" will be set automatically.</p>	
	Default	%TargetFiles%
	How to change	Directly enter to the text box (up to 256 characters) or select from the drop-down list.
	Restriction	%TargetFiles% Displays the file name in the output message.
		%TargetFiles%: %Options% Displays the file name and command line options in the output message.
		%Program% %Options% Displays the program name and command line options in the output message.
Format of build option list	<p>Specifies the display format of the build option list (see "2.12.4 Display a list of build options"). This applies to the options of the build tool to be used, and commands added by plugins. It does not apply to the options of commands specified in the [Commands executed before build processing] or [Commands executed after build processing] property. The following placeholders are supported. %Program%: Replaces with the program name under execution. %Options%: Replaces with the command line option under build execution. %TargetFiles%: Replaces with the file name being built. If this is blank, it is assumed that "%TargetFiles% : %Program% %Options%" will be set automatically.</p>	
	Default	%TargetFiles% : %Program% %Options%
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 256 characters

Commands executed before build processing	<p>Specifies the command to be executed before build processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before build processing.</p> <p>The placeholders can be described in the script. The specified command is displayed as the subproperty.</p> <table border="1"> <tr> <td data-bbox="505 864 671 943">Default</td><td data-bbox="671 864 1434 943">Commands executed before build processing[<i>number of defined items</i>]</td></tr> <tr> <td data-bbox="505 943 671 1048">How to change</td><td data-bbox="671 943 1434 1048">Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.</td></tr> <tr> <td data-bbox="505 1048 671 1128">Restriction</td><td data-bbox="671 1048 1434 1128">Up to 1023 characters Up to 64 items can be specified.</td></tr> </table>	Default	Commands executed before build processing[<i>number of defined items</i>]	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	Restriction	Up to 1023 characters Up to 64 items can be specified.
Default	Commands executed before build processing[<i>number of defined items</i>]						
How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.						
Restriction	Up to 1023 characters Up to 64 items can be specified.						
Commands executed after build processing	<p>Specifies the command to be executed after build processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after build processing.</p> <p>The placeholders can be described in the script. The specified command is displayed as the subproperty.</p> <table border="1"> <tr> <td data-bbox="505 1767 671 1823">Default</td><td data-bbox="671 1767 1434 1823">Commands executed after build processing[<i>number of defined items</i>]</td></tr> <tr> <td data-bbox="505 1823 671 1928">How to change</td><td data-bbox="671 1823 1434 1928">Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.</td></tr> <tr> <td data-bbox="505 1928 671 1998">Restriction</td><td data-bbox="671 1928 1434 1998">Up to 1023 characters Up to 64 items can be specified.</td></tr> </table>	Default	Commands executed after build processing[<i>number of defined items</i>]	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	Restriction	Up to 1023 characters Up to 64 items can be specified.
Default	Commands executed after build processing[<i>number of defined items</i>]						
How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.						
Restriction	Up to 1023 characters Up to 64 items can be specified.						

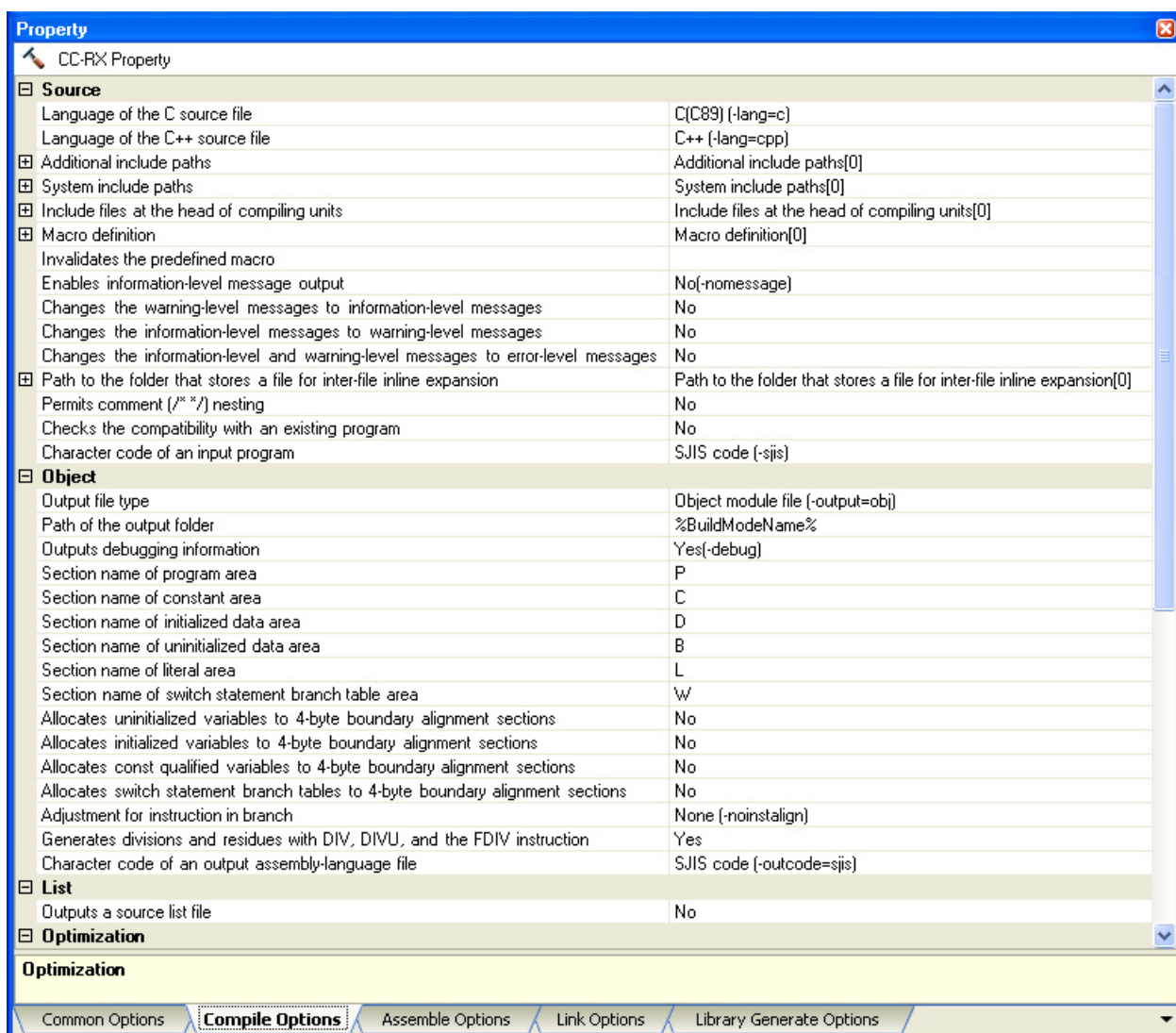
Other additional options	Input the option to be added additionally. The options set here are added at the end of the cx options group.	
	Default	Blank
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters

[Compile Options] tab

This tab shows the detailed information on the compile phase categorized by the following and the configuration can be changed.

- (1) [Source]
- (2) [Object]
- (3) [List]
- (4) [Optimization]
- (5) [Output File]
- (6) [MISRA C rule check]
- (7) [Others]

Figure A.5 Property Panel: [Compile Options] Tab



[Description of each category]

- (1) [Source]
The detailed information on the source is displayed and the configuration can be changed.

Language of the C source file	Selects language of the C source file. This corresponds to the -lang option of the compiler. This property is displayed only when [] in the [] property is specified.		
	Default	C(C89) (-lang=c)	
	How to change	Select from the drop-down list.	
	Restriction	C(C89) (-lang=c)	Compiles as a C (C89) source file.
Language of the C++ source file		C99 (-lang=c99)	Compiles as a C (C99) source file.
		Selects language of the C++ source file. This option corresponds to the -lang option of the compiler. This property is displayed only when [No] in the [Build simultaneously] property is specified.	
		Default	C++ (-lang=cpp)
		How to change	Select from the drop-down list.
Additional include paths		Restriction	C++ (-lang=cpp)
			Compiles as an EC++ source file.
		EC++ (-lang=ecpp)	Compiles as a C++ source file.
		Specifies the name of the path to the folder that stores the include file. The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. The reference point of the path is the project folder. This corresponds to the -include option of the compiler. The specified include path is displayed as the subproperty.	
	Default	Additional include paths[<i>number of defined items</i>]	
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 247 characters Up to 65536 items can be specified.	

System include paths	Changes the specified order of the include paths which the system set during compiling. This corresponds to the -include option of the compiler.	
	Default	System include paths[<i>number of defined items</i>]
	How to change	Edit by the System Include Path Order dialog box which appears when clicking the [...] button.
	Restriction	Changes not allowed (Only the specified order of the include paths can be changed.)
Include files at the head of compiling units	Specifies include files at the head of compiling units. The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. The reference point of the path is the project folder. This corresponds to the -preinclude option of the compiler.	
	Default	Include files at the head of compiling units[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.
	Restriction	Up to 259 characters Up to 65536 items can be specified.
Macro definition	Specifies the macro name to be defined. Specify in the format of " <i>macro name=string</i> ", with one macro name per line. The " <i>=string</i> " part can be omitted, and in this case, the macro name is assumed to be defined. This corresponds to the -define option of the compiler. The specified macro is displayed as the subproperty.	
	Default	Macro definition[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.
Invalidates the pre-defined macro	Specifies invalidates the predefined macro. If multiple macro names are specified, delimit them with a comma (example: <code>__DBL4, __SCHAR</code>). This corresponds to the -undefine option of the compiler.	
	Default	Blank
	How to change	Edit by the Specify The Predefined Macro dialog box which appears when clicking the [...] button.
	Restriction	Up to 32767 characters

Enables information-level message output	Specifies whether information level messages are output. This corresponds to the -message and -nomessage options of the compiler.			
	Default	No(-nomessage)		
	How to change	Select from the drop-down list.		
	Restriction	Yes(-message)	Enables information message output.	
	No(-nomessage)	Disables information message output.		
Suppresses the number of information-level messages	Specifies suppresses the number of information-level message. If multiple message numbers are specified, delimit them with a comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -nomessage option of the compiler. This property is displayed only when [No(-nomessage)] in the [Enables information-level message output] property is specified.			
	Default	Blank		
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.		
	Restriction	Up to 32767 characters		
Changes the warning-level messages to information-level messages	Selects whether to change the warning-level messages to information-level messages. This corresponds to the -change_message option of the compiler.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes(All) (-change_message=information)	Changes all warning-level messages to the information-level messages.	
		Yes(Specifies error number) (-change_message=information=<Error-Number>)	Changes the warning-level messages with the specified error numbers to the information-level messages.	
	No	Does not change the warning-level messages to the information-level messages.		
Error number of warning-level message	Specifies error number of warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=information=<ErrorNumber>)] in the [Changes the warning-level messages to information-level messages] property is specified.			
	Default	Blank		
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.		
	Restriction	Up to 32767 characters		

Changes the information-level messages to warning-level messages	Selects whether to change the information-level messages to warning-level messages. This corresponds to the -change_message option of the compiler.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes(All) (-change_message=warning)	Changes all information-level messages to warning-level messages.	
		Yes(Specifies error number) (-change_message=warning=<Error-Number>)	Changes the information-level messages with the specified error numbers to warning-level messages.	
No		Does not change the information-level messages to warning-level messages.		
Error number of information-level message	Specifies error number of information-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=warning=<ErrorNumber>)] in the [Changes the information-level messages to warning-level messages] property is specified.			
	Default	Blank		
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.		
	Restriction	Up to 32767 characters		
Changes the information-level and warning-level messages to error-level messages	Selects whether to change the information-level and warning-level messages to error-level messages. This corresponds to the -change_message option of the compiler.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes(All) (-change_message=error)	Changes all information-level and warning-level messages to error-level messages.	
		Yes(Specifies error number) (-change_message=error=<ErrorNumber>)	Changes the information-level and warning-level messages with the specified error numbers to error-level messages.	
No		Does not change the warning-level messages to information-level messages.		

Error number of information-level and warning-level message	Specifies error number of information-level and warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=error=<ErrorNumber>)] in the [Changes the information-level and warning-level messages to error-level messages] property is specified.		
	Default	Blank	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Permits comment (/* */) nesting	Selects whether to permit comment (/* */) nesting. This corresponds to the -comment option of the compiler.		
	Default	No (-comment=nonest)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-comment=nest)	Does not permit comment (/* */) nesting.
		No (-comment=nonest)	Permits comment (/* */) nesting.
Checks the compatibility with an existing program	Selects whether to check the compatibility with an existing program. This corresponds to the -check option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(NC compiler) (-check=nc)	Checks the compatibility with the R8C and M16C family C compilers.
		Yes(H8 compiler) (-check=ch38)	Checks the compatibility with the H8, H8S, and H8SX family C/C++ compilers.
		Yes(SH compiler) (-check=sh)	Checks the compatibility with the SuperH family C/C++ compilers.
		No	Does not check the compatibility with an existing program.

Character code of an input program	Selects character code of an input program. This corresponds to the -euc , -sjis , -latin1 , -utf8 , -big5 and -gb2312 option of the assembler.		
	Default	SJIS code (-sjis)	
	How to change	Select from the drop-down list.	
	Restriction	EUC code (-euc)	Handles the characters in strings, character constants, and comments by using EUC.
		SJIS code (-sjis)	Handles the characters in strings, character constants, and comments by using SJIS.
		ISO-Latin1 code (-latin1)	Handles the characters in strings, character constants, and comments by using ISO-Latin1.
		UTF-8 code (-utf8)	Handles the characters in strings, character constants, and comments by using UTF-8.
		Traditional Chinese character (-big5)	Handles the characters in strings, character constants, and comments by using Traditional Chinese character.
		Simplified Chinese character (-gb2312)	Handles the characters in strings, character constants, and comments by using Simplified Chinese character.

(2) [Object]

The detailed information on object is displayed and the configuration can be changed.

Output file type	Selects the type of the output file to be generated during a build. This corresponds to the -output option of the compiler.		
	Default	Object module file (-output=obj)	
	How to change	Select from the drop-down list.	
	Restriction	Object module file (-output=obj)	Outputs a relocatable file.
		Source file after pre-processed (-output=prep)	Outputs a source file after preprocessed.
		Source file after pre-processed(Disables #line output) (-output=prep -noline)	Disables #line output at preprocessor expansion.
Outputs debugging information	Selects whether to output debugging information to object module files. This corresponds to the -debug and -nodebug options of the compiler.		
	Default	Yes (-debug)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-debug)	Outputs debugging information to object module files.
		No (-nodebug)	Does not output debugging information to object module files.

Section name of program area	Specifies the section name of program area. This corresponds to the -section option of the compiler.	
	Default	P
	How to change	Directly enter in the text box.
	Restriction	Up to 32767 characters
Section name of constant area	Specifies the section name of constant area. This corresponds to the -section option of the compiler.	
	Default	C
	How to change	Directly enter in the text box.
	Restriction	Up to 32767 characters
Section name of initialized data area	Specifies the section name of initialized data area. This corresponds to the -section option of the compiler.	
	Default	D
	How to change	Directly enter in the text box.
	Restriction	Up to 32767 characters
Section name of uninitialized data area	Specifies the section name of uninitialized data area. This corresponds to the -section option of the compiler.	
	Default	B
	How to change	Directly enter in the text box.
	Restriction	Up to 32767 characters
Section name of literal area	Specifies the section name of literal area. This corresponds to the -section option of the compiler.	
	Default	L
	How to change	Up to 32767 characters
	Restriction	Up to 32767 characters
Section name of switch statement branch table area	Specifies the section name of switch statement branch table area. This corresponds to the -section option of the compiler.	
	Default	W
	How to change	Directly enter in the text box.
	Restriction	Up to 32767 characters

Allocates uninitialized variables to 4-byte boundary alignment sections	Selects whether to allocate uninitialized variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=B)	Allocates uninitialized variables to 4-byte boundary alignment sections.
		No	Does not allocate uninitialized variables to 4-byte boundary alignment sections.
Allocates initialized variables to 4-byte boundary alignment sections	Selects whether to allocate initialized variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=D)	Allocates initialized variables to 4-byte boundary alignment sections.
		No	Does not allocate initialized variables to 4-byte boundary alignment sections.
Allocates const qualified variables to 4-byte boundary alignment sections	Selects whether to allocate const qualified variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=C)	Allocates const qualified variables to 4-byte boundary alignment sections.
		No	Does not allocate const qualified variables to 4-byte boundary alignment sections.
Allocates switch statement branch tables to 4-byte boundary alignment sections	Selects whether to allocate switch statement branch tables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=W)	Allocates switch statement branch tables to 4-byte boundary alignment sections.
		No	Does not allocate switch statement branch tables to 4-byte boundary alignment sections.

Adjustment for instruction in branch	Selects adjustment for instruction in branch. This corresponds to the <code>-noinstalign</code> , <code>-instalign4</code> , and <code>-instalign8</code> option of the compiler.		
	Default	None (<code>-noinstalign</code>)	
	How to change	Select from the drop-down list.	
	Restriction	None (<code>-noinstalign</code>)	Does not align instructions at branch destinations.
		Execution in 4 bytes (<code>-instalign4</code>)	Aligns instructions at branch destinations to 4-byte boundaries.
		Execution in 4 bytes (Contains each loop head) (<code>-instalign4=loop</code>)	Aligns instructions at branch destinations to 4-byte boundaries (Contains head of each loop).
		Execution in 4 bytes (Contains each inmost loop head) (<code>-instalign4=inmost-loop</code>)	Aligns instructions at branch destinations to 4-byte boundaries (Contains head of each inmost loop).
		Execution in 8 bytes (<code>-instalign8</code>)	Aligns instructions at branch destinations to 8-byte boundaries.
		Execution in 8 bytes (Contains each loop head) (<code>-instalign8=loop</code>)	Aligns instructions at branch destinations to 8-byte boundaries (Contains head of each loop).
		Execution in 8 bytes (Contains each inmost loop head) (<code>-instalign8=inmost-loop</code>)	Aligns instructions at branch destinations to 8-byte boundaries (Contains head of each inmost loop).
Generates divisions and residues with DIV, DIVU, and the FDIV instruction	Selects whether to generate divisions and residues with DIV, DIVU, and the FDIV instruction. This corresponds to the <code>-nouse_div_inst</code> option of the compiler.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Generates code in which DIV, DIVU, or FDIV instructions are used.
		No (<code>-nouse_div_inst</code>)	Generates code in which no DIV, DIVU, or FDIV instructions are used.

Character code of an output assembly-language file	Selects character code of an output assembly-language file. This corresponds to the -outcode option of the compiler.		
	Default	SJIS code (-outcode=sjis)	
	How to change	Select from the drop-down list.	
	Restriction	EUC code (-outcode=euc)	Outputs characters in strings and character constants using EUC.
		SJIS code (-outcode=sjis)	Outputs characters in strings and character constants using SJIS.
		UTF-8 code (-outcode=utf8)	Outputs characters in strings and character constants using UTF-8. This item is not available when [C(C89) (-lang=c)] in the [Language of the C source file] property in the [Source] category is selected
		Traditional Chinese character (-outcode=big5)	Outputs characters in strings and character constants using Traditional Chinese character.
		Simplified Chinese character (-outcode=gb2312)	Outputs characters in strings and character constants using Simplified Chinese character.

(3) [\[List\]](#)

The detailed information on list file is displayed and the configuration can be changed.

This category is displayed only when [Object module file (-output=obj)] in the [Output file type] property in the [\[Object\]](#) category.

Outputs a source list file	Selects whether to output a source list file. This corresponds to the -listfile and -nolistfile option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-lisfile)	Outputs a source list file.
		No (-nolistfile)	Disable output of a source list file.
Outputs the C/C++ source file	Specifies the contents of the source list file. Selects whether to output the C/C++ source file. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=source)	Outputs the C/C++ source file.
		No	Does not output the C/C++ source file.

Outputs the state-ments unsatisfied in conditional assembly	Specifies the contents of the source list file. Selects whether to output the statements unsatisfied in conditional assembly. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=condi-tionals)	Outputs the statements unsatisfied in condi-tional assembly.
	No	Does not output the statements unsatisfied in conditional assembly.	
Outputs the informa-tion before .DEFINE replacement	Specifies the contents of the source list file. Selects whether to output the information before .DEFINE replacement. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=defini-tions)	Outputs the information before .DEFINE replacement.
	No	Does not output the information before .DEFINE replacement.	
Outputs the assem-bler macro expansion statements	Specifies the contents of the source list file. Selects whether to output the assembler macro expansion statements. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=expan-sions)	Outputs the assembler macro expansion statements.
	No	Does not output the assembler macro expan-sion statements.	

(4) [Optimization]

The detailed information on the optimization is displayed and the configuration can be changed.

Optimization level	Selects optimization level. This corresponds to the -optimize option of the compiler.		
	Default	2 (-optimize=2)	
	How to change	Select from the drop-down list.	
	Restriction	0 (-optimize=0)	Does not optimize the program.
		1 (-optimize=1)	Partially optimizes the program by automatically allocating variables to registers, integrating the function exit blocks, integrating multiple instructions which can be integrated, etc.
		2 (-optimize=2)	Performs overall optimization.
		Max (-optimize=max)	Performs optimization as much as possible.
Outputs additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. At linkage, inter-module optimization is applied to files for which this option has been specified. This corresponds to the -goptimize option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.
		No	Does not outputs additional information for inter-module optimization.
Optimization type	Selects optimization type. This corresponds to the -speed and -size option of the compiler.		
	Default	Optimizes with emphasis on code size (-size)	
	How to change	Select from the drop-down list.	
	Restriction	Optimizes with emphasis on execution performance (-speed)	Optimizes with emphasis on execution performance.
		Optimizes with emphasis on code size (-size)	Optimizes with emphasis on code size.
Loop expansion	Selects whether to optimize the loop expansion (for, while, and do-while). This corresponds to the -loop option of the compiler.		
	Default	Depends on the optimization level and optimization type options	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level and optimization type options	Depends on the optimization level and optimization type options.
		Expansion (-loop=<numeric value>)	Expands loop statements (for, while, and do-while).

Expansion maximum number	Specifies expansion maximum number. This corresponds to the suboption of -loop option of the compiler. This property is displayed only when [Expansion (-loop=<numeric value>)] in the [Loop expansion] property is selected.		
	Default	2 (decimal number)	
	How to change	Directly enter in the text box.	
	Restriction	1 to 32 (decimal number)	
Performs inline expansion automatically	Selects whether to perform inline expansion automatically. This option corresponds to the -inline and -noinline option of the compiler.		
	Default	Depends on the optimization level and optimization type options	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level and optimization type options	Depends on the optimization level and optimization type options.
		Yes (-inline=<numeric value>)	Performs inline expansion automatically.
		No (-noinline)	Does not perform inline expansion automatically.
Maximum increasing rate of function size	Specifies maximum increasing rate of function size. For example, when 100 is specified, inline expansion will be performed until the function size has increased by 100% (size is doubled). This option corresponds to the -inline option of the compiler. This property is displayed only when [Yes (-inline=<numeric value>)] in the [Performs inline expansion automatically] property is selected.		
	Default	100 (decimal number)	
	How to change	Select from the drop-down list.	
	Restriction	1 to 65535 (decimal number)	
Expansion method of the switch statement	Selects expansion method of the switch statement. This corresponds to the -case option of the compiler.		
	Default	Compiler automatically selects (-case=auto)	
	How to change	Select from the drop-down list.	
	Restriction	if_then method (-case=ifthen)	Expands the switch statement using the if_then method.
		Jumping to a table method (-case=table)	Expands the switch statement by using the table method.
		Compiler automatically selects (-case=auto)	Automatically selects the if_then method or table method.

Handles external variables as if they are volatile qualified	Selects whether to handle all external variables as if they are volatile qualified. This corresponds to the <code>-volatile</code> and <code>-novolatile</code> option of the compiler.		
	Default	No (<code>-novolatile</code>)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (<code>-volatile</code>)	Handles all external variables as if they were volatile qualified.
		No (<code>-novolatile</code>)	Does not handle external variables as if they were volatile qualified.
Performs the constant propagation of const qualified external variables	Selects whether to perform the constant propagation of const qualified external variables. Const qualified variables in a C++ source file cannot be controlled by this option (constant propagation is always performed). This corresponds to the <code>-const_copy</code> and <code>-noconst_copy</code> option of the compiler.		
	Default	Depends on the optimization level options	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level options	Depends on the optimization level options.
		Yes (<code>-const_copy</code>)	Enables constant propagation of const qualified external variables.
		No (<code>-noconst_copy</code>)	Disables constant propagation of const qualified external variables.
Conversion method of the divisions and residues of integer constants	Selects conversion method of the divisions and residues of integer constants. This corresponds to the <code>-const_div</code> and <code>-noconst_div</code> option of the compiler.		
	Default	Depends on the optimization type option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization type option	Depends on the optimization type option
		Instruction sequence using multiplication (<code>-const_div</code>)	Performs constant division (residue) by an instruction sequence using multiplication.
		Instruction sequence using division (<code>-noconst_div</code>)	Performs constant division (residue) by an instruction sequence using division.

Expansion method of the library function	Selects expansion method of the library function. This corresponds to the -library option of the compiler.		
	Default	Performs instruction expansion of several library functions (-library=intrinsic)	
	How to change	Select from the drop-down list.	
	Restriction	Calls library functions (-library=function)	Calls all library functions.
		Performs instruction expansion of several library functions (-library=intrinsic)	Performs instruction expansion for abs(), fabsf(), and library functions which can use string manipulation instructions.
Divides the optimizing ranges into many sections before compilation	Selects whether to divide the optimizing ranges of the large-size function into many sections before compilation. This corresponds to the -scope and -noscope option of the compiler.		
	Default	Depends on the optimization level option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level option	Depends on the optimization level option.
		Yes (-scope)	Divides the optimizing ranges of the large-size function into many sections before compilation.
		No (-noscope)	Does not divide the optimizing ranges before compilation.
Schedules the instruction taking into consideration pipeline processing	Selects whether to schedule the instruction taking into consideration pipeline processing. This corresponds to the -schedule and -noschedule option of the compiler.		
	Default	Depends on the optimization level option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level option	Depends on the optimization level option.
		Yes (-schedule)	Schedules instructions taking into consideration pipeline processing.
		No (-noschedule)	Does not schedule instructions.

Optimizes accesses to external variables	Selects whether to optimize accesses to external variables. This corresponds to the -nomap , -smap and -map option of the compiler. The item of "Yes(Optimizes the inter-module)(-map)" is hidden when it is library project.		
	Default	When [Max (-optimize=max)] in the [Optimization level] property is selected Yes(Optimizes the inter-module) (-map) Other than above No (-nomap)	
	How to change	Select from the drop-down list.	
	Restriction	Yes(Optimizes the inner-module) (-smap)	Optimizes accesses to external variables which are defined in the file to be compiled.
		Yes(Optimizes the inter-module) (-map)	Optimizes accesses to external variables.
		No (-nomap)	Disables optimization for accesses to external variables.
Perform inter-module optimization	Specify the level of inter-module optimization (such as function merging). Only [Yes(Level 1)(Perform)(-ip_optimize)] and [No] are displayed when [No] in the [Build simultaneously] property in the [Build Method] category from the [Common Options] tab is selected. This corresponds to the -whole_program , -merge_files , and -ip_optimize options of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(Level 3)(Perform with assuming it the whole program)(-whole_program)	Performs inter-module optimization assuming that the source files comprise the entire program. However, operation is not guaranteed if the preconditions are not met. See "-whole_program" for details about the preconditions.
		Yes(Level 2)(Perform with merging files)(-merge_files, -ip_optimize)	Merges two or more C source files and performs inter-module optimization. This item is displayed only when two or more source files are added to the project.
		Yes(Level 1)(Perform)(-ip_optimize)	Performs inter-module optimization for each file.
		No	Does not perform inter-module optimization.

Converts floating-point constant division into multiplication	Selects whether to convert floating-point constant division into multiplication of the corresponding reciprocals as constants. This corresponds to the -approxdiv option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-approxdiv)	Converts floating-point constant division into multiplication.
	No	Does not convert floating-point constant division into multiplication.	
Omits a check of the range for conversion between the floating type and unsigned integer type	Selects whether to omit a check of the range for conversion between the floating type and unsigned integer type. When "Yes" is specified, code performance of the relevant type conversion processing is improved. The conversion result may, however, differ from C/C++ language specifications, so take care on this point. This corresponds to the -simple_float_conv option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-simple_float_conv)	Omits part of the type conversion processing for the floating type.
	No	Does not omit part of the type conversion processing for the floating type.	
Performs optimization considering the type of the data indicated by the pointer	Selects whether to perform optimization considering the type of the data indicated by the pointer. Although the performance of object code is generally better than when -alias=noansi is specified, the results of execution may differ according to whether -alias=ansi or alias=noansi is specified. This corresponds to the -alias option of the compiler.		
	Default	No (-alias=noansi)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-alias=ansi)	Performs optimization considering the type of the data indicated by the pointer.
	No (-alias=noansi)	Does not perform optimization considering the type of the data indicated by the pointer.	

Optimizes modification of the operation order of a floating-point expression	<p>Selects whether to optimize modification of the operation order of a floating-point expression.</p> <p>Specifying the <code>-float_order</code> option generally improves the object performance compared to when not specifying it.</p> <p>However, the accuracy of operations may differ from that when <code>-float_order</code> is not specified.</p> <p>This corresponds to the <code>-float_order</code> option of the compiler.</p> <p>This property is valid only when [2 (<code>-optimize=2</code>)] or [Max (<code>-optimize=max</code>)] in the [Optimization level] property is specified.</p>			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes (<code>-float_order</code>)	Optimizes modification of the operation order in a floating-point expression.	
		No	Does not optimize modification of the operation order in a floating-point expression.	

(5) [Output File]

The detailed information on the output file check is displayed and the configuration can be changed.

Output assembly source file	Select whether to output the assembly source file of the compile result for the C source. This corresponds to the ?output=src option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes(-output=src)	Outputs the assembly source file of the compile result for the C source.	
No		Does not output the assembly source file of the compile result for the C source.		
Output preprocessed source file	Select whether to output the execution result of preprocessing for the source file to a file. This corresponds to the ?output=prep, -noline option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes(-output=prep)	Outputs the execution result of preprocessing for the source file to a file.	
		Yes(Suppress #line)(-output=prep -noline)	Outputs the execution result of preprocessing (suppress #line) for the source file to a file.	
No		Does not output the execution result of pre-processing for the source file to a file.		

(6) [MISRA C rule check]

The detailed information on the MISRA-C: 2004 rules check is displayed and the configuration can be changed.

Apply rule	Selects to apply MISRA C rule. This option corresponds to the -misra2004 option of the compiler.		
	Default	Not apply rule	
	How to change	Select from the drop-down list.	
	Restriction	Apply all rules (-misra2004=all)	Checks the source code against all of the rules that are supported.
		Apply specified rule number (-misra2004=apply)	Checks the source code against the rules with the selected numbers.
		Ignore specified rule number(-misra2004=ignore)	Checks the source code against the rules other than those with the selected numbers.
		Apply rules that are classified as "required" (-misra2004=required)	Checks the source code against the rules of the "required" type.
		Apply rules that are classified as "required" and specified rule number (-misra2004=required_add)	Checks the source code against the rules of the "required" type and the rules with the selected numbers.
		Ignore specified rule number from rules that are classified as "required" (-misra2004=required_remove)	Checks the source code against the rules other than those with the selected numbers among the rules of the "required" type.
		Apply rules that are described in the specified file (-misra2004=<file name>)	Checks the source code against the rules with the numbers written in the specified file.
		Not apply rule	Does not apply MISRA C rule.
Rule number description file	Specifies Rule number description file(misra2004 rule file). The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Apply rules that are described in the specified file (-misra2004=<file name>)] in the [Apply rule] property is selected.		
	Default	Blank	
	How to change	Directly enter to the text box or edit by the Specify Misra2004 Rule File dialog box which appears when clicking the [...] button.	
	Restriction	Up to 259 characters	

Rule number	Specifies the rule number. One or more rule numbers always in decimal must be specified. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Apply specified rule number (-misra2004=apply)] in the [Apply rule] property is selected.	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Specify the rule number dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Exclusion rule number	Specifies the exclusion rule number. One or more rule numbers always in decimal must be specified. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Ignore specified rule number(-misra2004=ignore)] in the [Apply rule] property is selected.	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Specify the rule number dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Check rule number besides required rule	Specifies the check rule number besides required rule. One or more rule numbers always in decimal must be specified. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Apply rules that are classified as "required" and specified rule number (-misra2004=required_add)] in the [Apply rule] property is selected.	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Specify the rule number dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Exclusion rule number from required rule	Specifies the exclusion rule number from required rule. One or more rule numbers always in decimal must be specified. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Ignore specified rule number from rules that are classified as "required" (-misra2004=required_remove)] in the [Apply rule] property is selected.	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Specify the rule number dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters

Rule check exclusion file	Specifies rule check exclusion file. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This option corresponds to the -ignore_files_misra option of the compiler. This option is not display when [Not apply rule] in the [Apply rule] property has been specified.	
	Default	Rule check exclusion file $[number\ of\ defined\ items]$
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.
	Restriction	Up to 259 characters Up to 65536 items can be specified.
Outputs message of the enhanced key word and extended specifications	Selects whether to output message of the enhanced key word and extended specifications. This option corresponds to the -check_language_extension option. This option is not display when [Not apply rule] in the [Apply rule] property has been specified.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-check_language_ext ension)
No		Disables complete checking against the MISRA-C: 2004 rules for parts of the code where it would otherwise be suppressed due to individual extensions from the C/C++ language specification.

(7) [Others]

Other detailed information on compilation is displayed and the configuration can be changed.

Outputs the copyright	Selects whether to output the copyright. This corresponds to the -nologo option of the compiler.		
	Default	No (-nologo)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-logo)	Outputs the copyright.
		No (-nologo)	Disables output of the copyright.

Outputs the cross reference information	Selects whether to output cross reference information. It is necessary to change the setting of the property of "Program Analyzer" to change this option.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes(-Xcref)	Outputs the cross reference information.	
		No	Does not output of the cross reference information.	
Commands executed before compile processing	<p>Specifies the command to be executed before compile processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %CompiledFile%: Replaces with the absolute path of the output file under compiling. %InputFile%: Replaces with the absolute path of the file to be compiled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before compile processing. The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>			
	Default	Commands executed before compile processing[<i>number of defined items</i>]		
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.		
	Restriction	Up to 1023 characters Up to 64 items can be specified.		

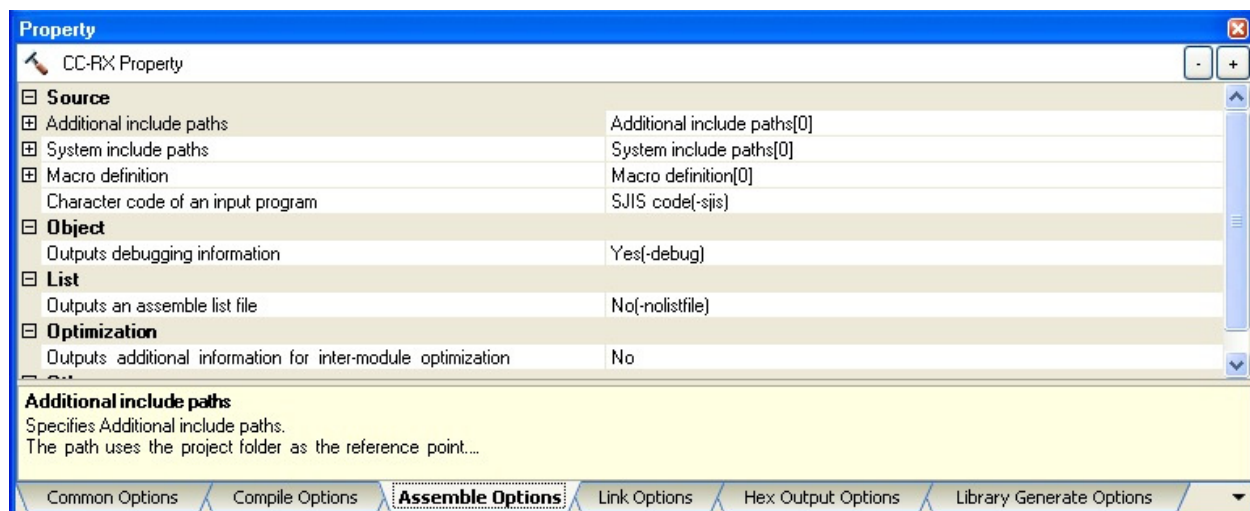
Commands executed after compile processing	<p>Specifies the command to be executed after compile processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %CompiledFile%: Replaces with the absolute path of the output file under compiling. %InputFile%: Replaces with the absolute path of the file to be compiled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after compile processing.</p> <p>The placeholders can be described in the script.</p> <p>The specified command is displayed as the subproperty.</p>	
	Default	Commands executed after compile processing[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.
Other additional options	Inputs the compile options to be added additionally. The options set here are added at the end of the compile options group.	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	The specified option is displayed.	
	Default	Command line[<i>number of defined items</i>]
	How to change	Changes not allowed

[Assemble Options] tab

This tab shows the detailed information on the assemble phase categorized by the following and the configuration can be changed.

- (1) [Source]
- (2) [Object]
- (3) [List]
- (4) [Optimization]
- (5) [Others]

Figure A.6 Property Panel: [Assemble Options] Tab



[Description of each category]

(1) [Source]

The detailed information on the source is displayed and the configuration can be changed.

Additional include paths	<p>Specifies the name of the path to the folder that stores the include file. The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. The reference point of the path is the project folder. This corresponds to the -include option of the assembler. The specified include path is displayed as the subproperty.</p>	
	Default	Additional include paths[<i>number of defined items</i>]
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 247 characters Up to 65536 items can be specified.
System include paths	<p>Changes the specified order of the include paths which the system set during assembling. This corresponds to the -include option of the assembler.</p>	
	Default	System include paths[<i>number of defined items</i>]
	How to change	Edit by the System Include Path Order dialog box which appears when clicking the [...] button.
	Restriction	Changes not allowed (Only the specified order of the include paths can be changed.)
Macro definition	<p>Specifies the macro name to be defined. Specifies in the format of "<i>macro name=string</i>", with one macro name per line. This corresponds to the -define option of the assembler. The specified macro is displayed as the subproperty.</p>	
	Default	Macro definition[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.

Character code of an input program	Selects character code of an input program. This corresponds to the -euc , -sjis , -latin1 , -big5 and -gb2312 option of the assembler.		
	Default	SJIS code (-sjis)	
	How to change	Select from the drop-down list.	
	Restriction	EUC code (-euc)	Handles the characters in strings, character constants, and comments by using EUC.
		SJIS code (-sjis)	Handles the characters in strings, character constants, and comments by using SJIS.
		ISO-Latin1 code (-latin1)	Handles the characters in strings, character constants, and comments by using ISO-Latin1.
		Traditional Chinese character (-big5)	Handles the characters in strings, character constants, and comments by using Traditional Chinese character.
		Simplified Chinese character (-gb2312)	Handles the characters in strings, character constants, and comments by using Simplified Chinese character.

(2) [Object]

The detailed information on the object is displayed and the configuration can be changed.

This category is displayed only when [No] in the [Build simultaneously] property is selected.

Path of the output folder	Specifies the output destination folder for the output file. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -output option of the assembler. This property is displayed only when [No] in the [Build simultaneously] property is selected.		
	Default	%BuildModeName%	
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.	
	Restriction	Up to 247 characters	
Outputs debugging information	Selects whether to output debugging information to object module files. This corresponds to the -debug and -nodebug options of the assembler. This property is displayed only when [No] in the [Build simultaneously] property is selected.		
	Default	Yes (-debug)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-debug)	Outputs debugging information to object module files.
No (-nodebug)		Does not output debugging information to object module files.	

Output additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. At linkage, inter-module optimization is applied to files for which this option has been specified. This corresponds to the <code>-goptimize</code> option of the assembler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.
		No	Does not output additional information for inter-module optimization.

(3) [List]

The detailed information on the list is displayed and the configuration can be changed.
This category is displayed only when [No] in the [Build simultaneously] property is selected.

Outputs a assemble list file	Selects whether to output an assemble list file. This corresponds to the <code>-listfile</code> and <code>-nolistfile</code> option of the assembler. This property is displayed only when [No] in the [Build simultaneously] property is selected.		
	Default	No (<code>-nolistfile</code>)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (<code>-listfile</code>)	Outputs an assemble list file.
	No (<code>-nolistfile</code>)	Does not output an assemble list file.	
Outputs the state-ments unsatisfied in conditional assembly	Specifies the contents of the assemble list file. Selects whether to output the statements unsatisfied in conditional assembly. This corresponds to the <code>-show</code> option of the assembler. This property is displayed only when [Yes (<code>-listfile</code>)] in the [Output a assemble list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (<code>-show=con- ditionals</code>)	Outputs the statements unsatisfied in conditional assembly.
	No	Does not output the statements unsatisfied in conditional assembly.	
Outputs the informa-tion before <code>.DEFINE</code> replacement	Specifies the contents of the assemble list file. Selects whether to output the information before <code>.DEFINE</code> replacement. This corresponds to the <code>-show</code> option of the assembler. This property is displayed only when [Yes (<code>-listfile</code>)] in the [Output a assemble list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (<code>-show=defi- nitions</code>)	Outputs the information before replacement specified with <code>.DEFINE</code> .
	No	Does not output the information before replace-ment specified with <code>.DEFINE</code> .	

Outputs the assembler macro expansion statements	Specifies the contents of the assembler list file. Selects whether to output the assembler macro expansion statements. This corresponds to the -show option of the assembler. This property is displayed only when [Yes (-listfile)] in the [Output a assemble list file] property is selected.				
	Default	No			
	How to change	Select from the drop-down list.			
	Restriction	<table><tr><td>Yes (-show=expansions)</td><td>Outputs the macro expansion statements.</td></tr><tr><td>No</td><td>Does not output the macro expansion statements.</td></tr></table>	Yes (-show=expansions)	Outputs the macro expansion statements.	No
Yes (-show=expansions)	Outputs the macro expansion statements.				
No	Does not output the macro expansion statements.				

(4) [Optimization]

The detailed information on the optimization is displayed and the configuration can be changed.
This category is displayed only when [No] in the [Build simultaneously] property is selected.

Output additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. At linkage, inter-module optimization is applied to files for which this option has been specified. This corresponds to the -goptimize option of the assembler.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.	
		No	Does not output additional information for inter-module optimization.	

(5) [Others]

Other detailed information on assembly is displayed and the configuration can be changed.

Checks for a privileged instruction	Selects whether to check for a privileged instruction. This corresponds to the -chkpm option of the assembler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-chkpm)	Checks for a privileged instruction.
		No	Does not check for a privileged instruction.
Checks for a floating-point operation instruction	Selects whether to check for a floating-point operation instruction. This corresponds to the -chkfpu option of the assembler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-chkfpu)	Checks for a floating-point operation instruction.
		No	Does not check for a floating-point operation instruction.

Checks for a DSP instruction	Selects whether to check for a DSP instruction. This corresponds to the -chkdsp option of the assembler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-chkdsp)	Checks for a DSP instruction.
		No	Does not check for a DSP instruction.
Outputs the copyright	Selects whether to output the copyright. This corresponds to the -logo and -nologo option of the assembler.		
	Default	No (-nologo)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-logo)	Outputs the copyright.
		No (-nologo)	Disables output of the copyright.
Commands executed before assemble processing	<p>Specifies the command to be executed before assemble processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %AssembledFile%: Replaces with the absolute path of the output file under assembling. %BuildModeName%: Replaces with the build mode name. %InputFile%: Replaces with the absolute path of the file to be assembled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before assemble processing. The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>		
	Default	Commands executed before assemble processing[<i>number of defined items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 1023 characters	
		Up to 64 items can be specified.	

Commands executed after assemble processing	<p>Specifies the command to be executed after assemble processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %AssembledFile%: Replaces with the absolute path of the output file under assembling. %BuildModeName%: Replaces with the build mode name. %InputFile%: Replaces with the absolute path of the file to be assembled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after assemble processing.</p> <p>The placeholders can be described in the script.</p> <p>The specified command is displayed as the subproperty.</p>	
	Default	Commands executed after assemble processing[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.
Other additional options	<p>Inputs the assemble options to be added additionally. The options set here are added at the end of the assemble options group.</p>	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	The specified option is displayed.	
	Default	Command line[<i>number of defined items</i>]
	How to change	Changes not allowed

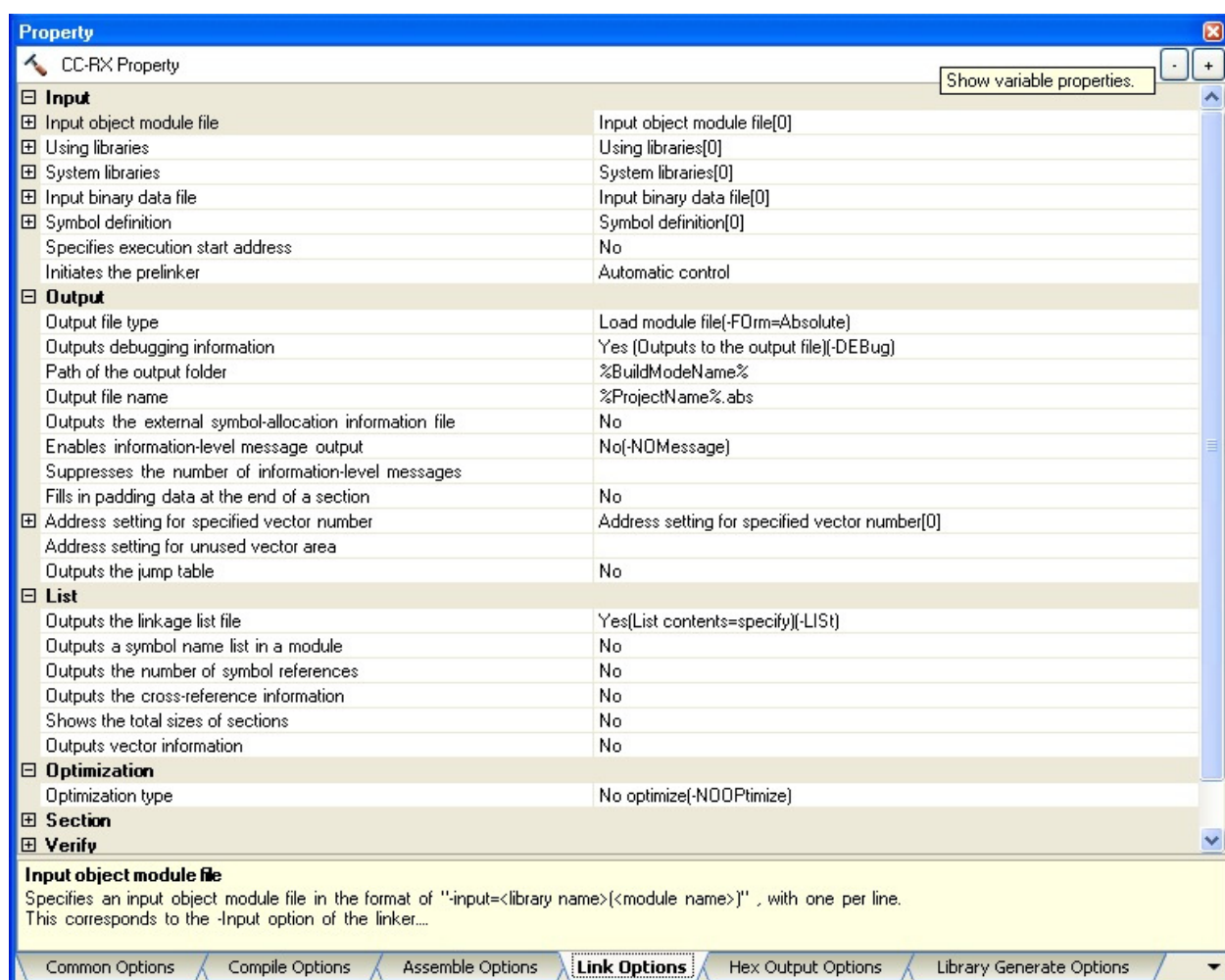
[Link Options] tab

This tab shows the detailed information on the link phase categorized by the following and the configuration can be changed.

- (1) [Input]
- (2) [Output]
- (3) [List]
- (4) [Optimization]
- (5) [Section]
- (6) [Verify]
- (7) [Others]

Caution This tab is not displayed for the library project.

Figure A.7 Property Panel: [Link Options] Tab



[Description of each category]

- (1) [Input]
The detailed information on input files is displayed and the configuration can be changed.

Input object module file	<p>Specifies an input object module file. Add one file in one line. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This corresponds to the -Input option of the linker. The specified file name is displayed as the subproperty.</p>	
	Default	Input object module file[<i>number of defined items</i>]
	How to change	<p>Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.</p>
	Restriction	<p>Up to 32767 characters Up to 65536 items can be specified.</p>
Using libraries	<p>Specifies an input library file. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This corresponds to the -library option of the linker. The library file name is displayed as the subproperty.</p>	
	Default	Input library file [<i>number of defined items</i>]
	How to change	<p>Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.</p>
	Restriction	<p>Up to 259 characters Up to 65536 items can be specified.</p>
System library file	<p>The specified system library files are displayed. This corresponds to the -library option of the linker.</p>	
	Default	System library file[<i>number of defined items</i>]
	Restriction	Changes not allowed

Input binary data file	<p>Specifies an input binary data file.</p> <p>Specifies in the format of "<i>file name (section name:boundary alignment/section attribute,symbol name)</i>", with one file name per line.</p> <p>The "<i>:boundary alignment</i>", "<i>/section attribute</i>", or "<i>,symbol name</i>" part can be omitted.</p> <p>1, 2, 4, 8, 16, or 32 can be specified for the "<i>boundary alignment</i>".</p> <p>When the boundary alignment specification is omitted, 1 is used as the default.</p> <p>CODE or DATA can be specified for the "<i>section attribute</i>".</p> <p>When "<i>section attribute</i>" specification is omitted, the write, read, and execute attributes are all enabled by default.</p> <p>The following placeholders are supported.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the product install folder.</p> <p>This corresponds to the -binary option of the linker.</p> <p>The binary data file name is displayed as the subproperty.</p>		
	Default	Specify binary data file[<i>number of defined items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 32767 characters Up to 65536 items can be specified.	
Symbol definition	<p>Specifies the symbol name to be defined.</p> <p>Specifies in the format of "<i>symbol name=symbol name</i>" or "<i>symbol name=numerical value</i>", with one symbol name per line.</p> <p>Specifies the numerical value in the hexadecimal notation.</p> <p>This corresponds to the -define option of the linker.</p>		
	Default	Symbol definition[<i>number of defined items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 32767 characters Up to 65536 items can be specified.	
Specifies execution start address	<p>Selects whether to specify the execution start address.</p> <p>Specifies in the format of "symbol" or "address".</p> <p>Specifies an address in the hexadecimal notation.</p> <p>This corresponds to the -entry option of the linker.</p>		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-ENTry)	Specifies execution start address.
	No	Does not specify execution start address.	

Execution start address	Specifies the execution start address. Specifies in the format of " <i>symbol name</i> " or " <i>address</i> ". Specifies an address in the hexadecimal notation. This corresponds to the -entry option of the linker. This property is displayed only when [Yes(-ENTry)] in the [Specifies execution start address] property is selected.		
	Default	Blank	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Initiates the prelinker	Selects whether to initiate the prelinker (The automatic generation of C++ template instance). This corresponds to the -noprelink option of the linker.		
	Default	Automatic control	
	How to change	Select from the drop-down list.	
	Restriction	Automatic control	Disables the prelinker initiation if there is no ii file in a file to be input to linker.
		Yes	Enables the prelinker initiation.
		No (-NOPRElink)	Disables the prelinker initiation.

(2) [Output]

The detailed information on output files are displayed and the configuration can be changed.

Output file type	The output file type is displayed. This corresponds to the -form option of the linker.		
	Default	Load module file (-FOrm=Absolute)	
	How to change	Changes not allowed.	
Outputs debugging information	Specifies whether debugging information is output. This corresponds to the -nodebug , -sdebug , and -debug options of the linker.		
	Default	Yes (Outputs to the output file) (-DEBUg)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (Outputs to the output file) (-DEBUg)	Outputs a debugging information to the output file.
		Yes (Outputs to <output file name>.dbg file) (-SDebug)	Outputs a debugging information to <output file name>.dbg file.
		No (-NODEBUg)	Does not output a debugging information.

ROM to RAM mapped section	Reserves ROM and RAM areas in the initialized data area and relocates a defined symbol in the ROM section with the specified address in the RAM section. Specifies in the format of " <i>ROM section name=RAM section name</i> ", with one section name per line. This corresponds to the -rom option of the linker.	
	Default	ROM to RAM mapped section [<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 1024 items can be specified.
Divides load module file	Selects whether to divide load module file. This corresponds to the -output option of the linker.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes Divides load module file. No Does not divide load module file.
Path of the output folder	Specifies path of the output folder. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -output option of the linker.	
	Default	%BuildModeName%
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.
	Restriction	Up to 247 characters
Output file name	Specifies an output file name. If the extension is omitted, ".abs" is automatically added. The following placeholders are supported. %ProjectName%: Replaces with the project name. If this is blank, it is assumed that "%ProjectName%.abs" has been specified. This corresponds to the -output option of the linker.	
	Default	%ProjectName%.abs
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

Division output file	Specifies the divided output file. Specifies in the format of " <i>file name=start address-end address</i> " or " <i>file name=section name</i> ", with one file name per line. To define multiple sections, use a colon to separate each entry written, as in " <i>file name=section name:section name</i> ". (example: file1.abs=sec1:sec2). Specifies an address in the hexadecimal notation (example: file2.abs=400-ffff). If the extension is omitted, ".abs" is automatically added. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. If this is blank, it is assumed that "%ProjectName%.abs" has been specified. This corresponds to the -output option of the linker. This property is displayed only when [Yes] in the [Divides load module file] property is selected.		
	Default	Specifies divide output file[<i>number of defined items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 32767 characters Up to 1024 items can be specified.	
Outputs the external symbol-allocation information file	Selects whether to output the external symbol-allocation information file. This corresponds to the -map option of the linker.		
	Default	When [Yes(Optimizes the inter-module) (-map)] in the [Optimizes accesses to external variables] property in the [[Optimization] category from the [Compile Options] tab is selected Yes (-Map) Other than above No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-Map)	Divides load module file.
		No	Does not divide load module file.
Enables information-level message output	Specifies whether to enable information-level message output. This corresponds to the -message , -msg_unused , and -nomessage options of the linker.		
	Default	No (-NOMessage)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-Message)	Outputs information level messages.
		Yes (Notify unused symbol) (-Message -MSg_unused)	Notifies the user of the externally defined symbol which is not referenced.
		No (-NOMessage)	Disables the output of information level messages.

Suppresses the number of information-level messages	Specifies suppresses the number of information-level messages. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -nomessage option of the linker. This property is displayed only when [No (-NOMessage)] in the [Enables information-level message output] property is specified.			
	Default	Blank		
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.		
	Restriction	Up to 32767 characters		
Fills in padding data at the end of a section	Selects whether to fill in padding data at the end of a section. This corresponds to the -padding option of the linker.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes (-PADDING)	Fills in padding data at the end of a section.	
		No	Does not fill in padding data at the end of a section.	
Address setting for specified vector number	Specifies the address setting for specified vector number. Specifies in the format of " <i>vector number=symbol</i> " or " <i>vector number=address</i> ", with one vector number per line. Specifies a decimal value from 0 to 255 for " <i>vector number</i> ". Specifies the external name of the target function for " <i>symbol</i> ". Specifies an address in the hexadecimal notation. This corresponds to the -vectn option of the linker.			
	Default	Address setting for specified vector number[<i>number of defined items</i>]		
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.		
	Restriction	Up to 32767 characters Up to 65536 items can be specified.		
Address setting for unused vector area	Specifies the address setting for unused vector area. Specifies in the format of " <i>symbol</i> " or " <i>address</i> ". Specifies the external name of the target function for " <i>symbol</i> ". Specifies an address in the hexadecimal notation. This corresponds to the -vect option of the linker.			
	Default	Blank		
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.		
	Restriction	Up to 32767 characters		

Outputs the jump table	Selects whether to output the jump table. This corresponds to the -jump_entries_for_pic option of the linker.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-JUMP_ENTRIES_FOR_PIC)	Outputs a jump table.
No		Does not output a jump table.	
The section that outputs a jump table to branch to external definition symbols	Specifies the section that outputs a jump table to branch to external definition symbols. Specifies in the format of " <i>section name</i> ", with one section per line. This corresponds to the -jump_entry_for_pic option of the linker. This property is displayed only when [Yes (-JUMP_ENTRIES_FOR_PIC)] in the [Outputs the jump table] property is specified.		
	Default	The section that outputs a jump table to branch to external definition symbols[<i>number of defined items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 32767 characters Up to 65536 items can be specified.	

(3) [List]

The detailed information on the list are displayed and the configuration can be changed.

Outputs the linkage list file	Selects whether to output the linkage list file. This corresponds to the -list and -show option of the linker.		
	Default	Yes (List contents=specify) (-LISt)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (List contents=not specify) (-LISt -SHow)	Outputs the default information associated with a output file type to a linkage list file.
		Yes (List contents=ALL) (-LISt -SHow=ALL)	Outputs all information associated with a output file type to a linkage list file.
		Yes (List contents=specify) (-LISt)	Outputs the specified information to a linkage list file.
		No	Does not output a linkage list file.

Outputs a symbol name list in a module	Selects whether to output a symbol name list in a module. This corresponds to the -show option of the linker. This property is not displayed when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-SHow=SYmbo)	Outputs a symbol information to the linkage list file.
	No	Does not output a symbol information to the linkage list file.	
Outputs the number of symbol references	Selects whether to output the number of symbol references. This corresponds to the -show option of the linker. This property is not displayed when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-SHow=Reference)	Outputs the number of symbol references to the linkage list file.
	No	Does not output the number of symbol references to the linkage list file.	
Outputs the cross-reference information	Selects whether to output the cross-reference information. This corresponds to the -show option of the linker. This property is not displayed when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-SHow=Xreference)	Outputs the cross reference information to the linkage list file.
	No	Does not output the cross reference information to the linkage list file.	
Shows the total sizes of sections	Selects whether to show the total sizes of sections. This corresponds to the -show option of the linker. This property is not displayed when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-SHow=Total_size)	Shows the total sizes of sections allocated to the ROM and RAM areas.
	No	Does not show the total sizes of sections.	

Outputs vector information	Selects whether to output vector information. This corresponds to the -show option of the linker. This property is not displayed when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property is selected.				
	Default	No			
	How to change	Select from the drop-down list.			
	Restriction	<table><tr><td>Yes (-SHOW=VECTOR)</td><td>Outputs vector information to the linkage list file.</td></tr><tr><td>No</td><td>Does not output vector information to the linkage list file.</td></tr></table>	Yes (-SHOW=VECTOR)	Outputs vector information to the linkage list file.	No
Yes (-SHOW=VECTOR)	Outputs vector information to the linkage list file.				
No	Does not output vector information to the linkage list file.				

(4) [Optimization]

The detailed information on the optimization is displayed and the configuration can be changed.

Optimization type	Specifies optimization type. This corresponds to the -nooptimize and -optimize options of the linker.		
	Default	No optimize (-NOOPTimize)	
	How to change	Select from the drop-down list.	
	Restriction	No optimize (-NOOPTimize)	Does not execute optimization for a module.
		All(-OPTimize)	Provides all optimizations.
		Speed-oriented optimization (-OPTimize=SPeed)	Provides optimization for speed.
		Safe optimization (-OPTimize=SAFE)	Provides safe optimization.
	Custom	Performs optimization for the specified options.	
Deletes variables/ functions that are not referenced	Selects whether to delete variables/functions that are not referenced. This corresponds to the -optimize option of the linker. This property is displayed only when [Custom] in the [Optimization type] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-OPTimize=SYmbol_delete)	Deletes variables/functions that are not referenced.
		No	Does not delete variables/functions that are not referenced.

Creates a subroutine for the same instruction sequence	Selects whether to create a subroutine for the same instruction sequence. This corresponds to the -optimize option of the linker. This property is displayed only when [Custom] in the [Optimization type] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-OPTimize=SAME_code)	Creates a subroutine for the same instruction sequence.
		No	Does not create a subroutine for the same instruction sequence.
Minimum code size	Specifies the minimum code size for the optimization. This corresponds to the -samesize option of the linker. This property is displayed only when [Yes (-OPTimize=SAME_code)] in the [Creates a subroutine for the same instruction sequence] property is specified.		
	Default	1E (hexadecimal number)	
	How to change	Directly enter in the text box.	
	Restriction	8 to 7FFF (hexadecimal number)	
Replaces an instruction with a smaller-size instruction	Selects whether to replace an instruction with a smaller-size instruction. This corresponds to the -optimize option of the linker. This property is displayed only when [Custom] in the [Optimization type] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-OPTimize=SHort_format)	Replaces an instruction with a smaller-size instruction.
		No	Does not replace an instruction with a smaller-size instruction.
Optimizes branch instruction size	Selects whether to optimize branch instruction size. This corresponds to the -optimize option of the linker. This property is displayed only when [Custom] in the [Optimization type] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-OPTimize=Branch)	Optimizes branch instruction size according to program allocation information.
		No	Does not optimize branch instruction size.

Unreferenced symbol that disables deletion by optimization	Specifies the unreferenced symbol that disables deletion by optimization. Specifies in the format of " <i>symbol name</i> ", with one symbol name per line. This option corresponds to the -symbol_forbid option of the linker. This property is not displayed when [No optimize (-NOOPTimize)] in the [Optimization type] property is selected.	
	Default	Unreferenced symbol that disables deletion by optimization[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.
Same-code that disables unification regarding optimization	Specifies the same-code that disables unification regarding optimization. Specifies in the format of " <i>function name</i> ", with one function name per line. This option corresponds to the -samecode_forbid option of the linker. This property is not displayed when [No optimize (-NOOPTimize)] in the [Optimization type] property is selected.	
	Default	Same-code that disables unification regarding optimization[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.
Section to disable optimization	Specifies the Section to disable optimization. Specifies in the format of " <i>file name</i> (<i>Section name</i> [...])" or " <i>module name</i> (<i>section name</i> [...])", with one per line. The " <i>file name</i> " or " <i>module name</i> " part can be omitted. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This option corresponds to the -section_forbid option of the linker. This property is not displayed when [No optimize (-NOOPTimize)] in the [Optimization type] property is selected.	
	Default	Section to disable optimization[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.

Address range to disable optimization	Specifies an address range where optimization is disabled. Specifies in the format of " <i>address+size</i> ", with one per line. The " <i>+size</i> " part can be omitted. Specifies an address or the size in the hexadecimal notation. This option corresponds to the -absolute_forbid option of the linker. This property is not displayed when [No optimize (-NOOptimize)] in the [Optimization type] property is selected.	
	Default	Address range to disable optimization[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.

(5) [Section]

The detailed information on the section is displayed and the configuration can be changed.

Section start address	Specifies the section start address. This corresponds to the -start option of the linker.	
	Default	The peculiar value for the target device
	How to change	Directly enter in the text box or edit by the Section Settings dialog box which appears when clicking the [...] button.
	Restriction	Up to 1022 characters
The specified section that outputs externally defined symbols to the file	Specifies the section start address. This corresponds to the -fsymbol option of the linker.	
	Default	The specified section that outputs externally defined symbols to the file[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.
Section alignment	Specifies the section name to be changed to 0x10 bytes alignment. Specifies in the format of " <i>section name</i> ", with one section name per line. This corresponds to the -aligned_section option of the linker.	
	Default	Section alignment[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.

ROM to RAM mapped section	Reserves ROM and RAM areas in the initialized data area and relocates a defined symbol in the ROM section with the specified address in the RAM section. Specifies in the format of "ROM section name=RAM section name", with one section name per line. This corresponds to the -rom option of the linker.	
	Default	ROM to RAM mapped section [number of defined items]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 1024 items can be specified.

(6) [Verify]

The detailed information on the verify is displayed and the configuration can be changed.

Checks the section larger than the specified range of addresses	Selects whether to check the section larger than the specified range of addresses. This corresponds to the -cpu option of the linker.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-CPu) Check CPU information. No Does not check CPU information.
Address range of the memory type	Specifies the address range of the memory type. Specifies in the format of " <i>memory type</i> =start address-end address", with one memory type per line. Specifies ROM, RAM, or FIX for " <i>memory type</i> ". Specifies an address in hexadecimal notation. This corresponds to the -cpu option of the linker. This property is displayed only when [Yes (-CPu)] in the [Checks the section larger than the specified range of addresses] property is selected.	
	Default	Address range to disable optimization[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.
Allocates to the next area of the same memory type or the section is divided	Selects whether to allocate to the next area of the same memory type or the section is divided. This corresponds to the -cpu option of the linker. This property is displayed only when the address range of the memory type in the [Address range of the memory type] property is specified.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-CPu=stride) Allocates same sort of memory space. No Does not allocate same sort of memory space.

Not divide the specified section	Allocates the specified section to another available area of the same memory type without dividing the section. Specifies in the format of " <i>section name</i> ", with one section name per line. This corresponds to the -contiguous_section option of the linker. This property is displayed only when [Yes (-CPu=stride)] in the [Allocates to the next area of the same memory type or the section is divided] property is selected.	
	Default	Not divide the specified section[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.

(7) [Others]

Other detailed information on linking are displayed and the configuration can be changed.

Outputs a stack use information file	Selects whether to output a stack information file. This corresponds to the -stack option of the linker.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-STACK) Outputs a stack information file. No Does not output a stack information file.
Compresses the debugging information	Selects whether to compress the debugging information. This corresponds to the -compress and -nocompress option of the linker.	
	Default	No (-NOCOMpress)
	How to change	Select from the drop-down list.
	Restriction	Yes (-Compress) Compresses the debugging information. No (-NOCOMpress) Does not compress the debugging information.
Reduces the memory occupancy	Selects whether to reduce the memory occupancy. This corresponds to the -memory option of the linker. This property is not displayed when any one of the following items is selected. [Yes (-Map)] in the [Outputs the external symbol-allocation information file] property in the [Output] category [Yes (-SHow=Reference)] in the [Outputs the number of symbol references] property, or [Yes(-SHow=Xreference)] in the [Outputs the cross-reference information] property in the [List] category [Yes (-CPu=stride)] in the [Allocates to the next area of the same memory type or the section is divided] property, [Yes (-STACK)] in the [Outputs a stack use information file] property, or [Yes (-Compress)] in the [Compresses the debugging information] property in the [Verify] category.	
	Default	No (-MEMory=High)
	How to change	Select from the drop-down list.
	Restriction	Yes (-MEMory=Low) Reduces the memory occupancy. No (-MEMory=High) Does not reduce the memory occupancy.

Changes the warning-level messages to information-level messages	Selects whether to change the warning-level messages to information-level messages. This corresponds to the -change_message option of the linker.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=information)	Changes all warning-level messages to the information-level messages.
		Yes(Specifies error number) (-change_message=information=<Error-Number>)	Changes the warning-level messages with the specified error numbers to the information-level messages.
No		Does not change the warning-level messages to the information-level messages.	
Error number of warning-level message	Specifies error number of warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the linker. This property is displayed only when [Yes(Specifies error number) (-change_message=information=<ErrorNumber>)] in the [Changes the warning-level messages to information-level messages] property is specified.		
	Default	Blank	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Changes the information-level messages to warning-level messages	Selects whether to change the information-level messages to warning-level messages. This corresponds to the -change_message option of the linker.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=warning)	Changes all information-level messages to warning-level messages.
		Yes(Specifies error number) (-change_message=warning=<Error-Number>)	Changes the information-level messages with the specified error numbers to warning-level messages.
No		Does not change the information-level messages to warning-level messages.	

Error number of information-level message	Specifies error number of information-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the linker. This property is displayed only when [Yes(Specifies error number) (-change_message=warning=<ErrorNumber>)] in the [Changes the information-level messages to warning-level messages] property is specified.	
	Default	Blank
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 32767 characters
Changes the information-level and warning-level messages to error-level messages	Selects whether to change the information-level and warning-level messages to error-level messages. This corresponds to the -change_message option of the linker.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes(All) (-change_message=error) Changes all information-level and warning-level messages to error-level messages.
		Yes(Specifies error number) (-change_message=error=<ErrorNumber>) Changes the information-level and warning-level messages with the specified error numbers to error-level messages.
	No	Does not change the warning-level messages to information-level messages.
Error number of information-level and warning-level message	Specifies error number of information-level and warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the linker. This property is displayed only when [Yes(Specifies error number) (-change_message=error=<ErrorNumber>)] in the [Changes the information-level and warning-level messages to error-level messages] property is specified.	
	Default	Blank
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 32767 characters
Deletes local symbol name information	Selects whether to delete local symbol name information. This corresponds to the -hide option of the linker.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-Hide) Deletes local symbol name information.
		No Does not delete local symbol name information.

Displays the total sizes of sections	Selects whether to display the total sizes of sections. This corresponds to the -total_size option of the linker.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-Total_size)	Displays the total sizes of sections.
		No	Does not display the total sizes of sections.
Displays the copyright information	Selects whether to display the copyright information. This corresponds to the -logo and -nologo option of the linker.		
	Default	No (-NOLOgo)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-LOgo)	Displays the total sizes of sections.
		No (-NOLOgo)	Does not display the total sizes of sections.
Commands executed before link processing	<p>Specifies the command to be executed before link processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %LinkedFile%: Replaces with the absolute path of the output file under link processing. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before link processing. The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>		
	Default	Commands executed before link processing[<i>number of defined items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 1023 characters	
		Up to 64 items can be specified.	

Commands executed after link processing	<p>Specifies the command to be executed after link processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %LinkedFile%: Replaces with the absolute path of the output file under link processing. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after link processing. The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>	
	Default	Commands executed after link processing[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.
Other additional options	<p>Inputs the link options to be added additionally. The options set here are added at the end of the link options group.</p>	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	The specified option is displayed.	
	Default	Command line[<i>number of defined items</i>]
	How to change	Changes not allowed

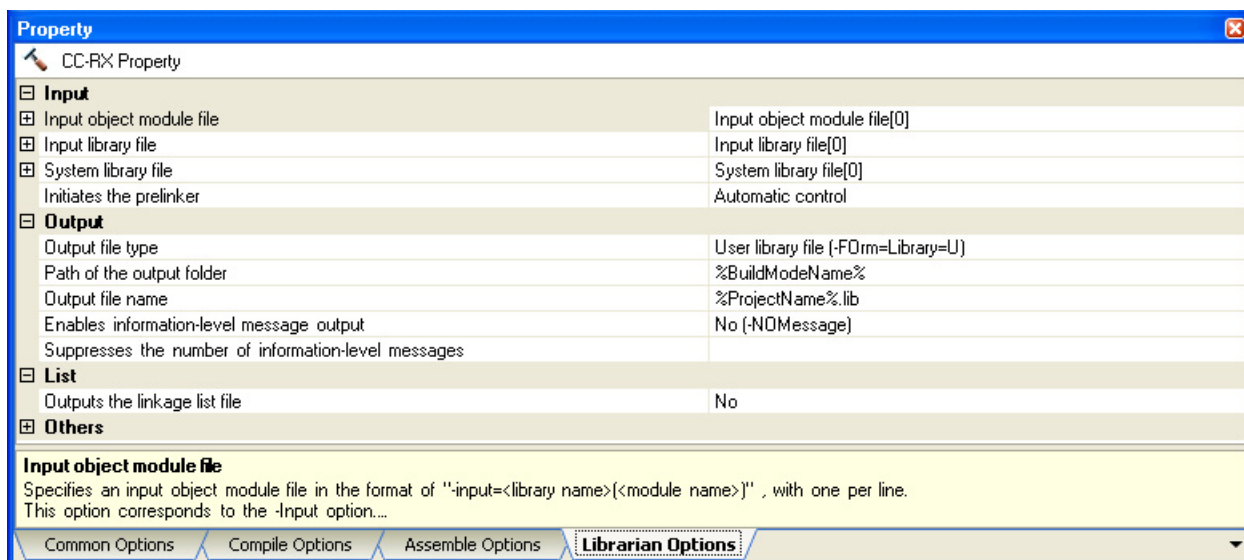
[Librarian Options] tab

This tab shows the detailed information on the link phase categorized by the following and the configuration can be changed.

- (1) [Input]
- (2) [Output]
- (3) [List]
- (4) [Others]

Caution This tab is not displayed for the application project.

Figure A.8 Property Panel: [Librarian Options] Tab



[Description of each category]

- (1) [Input]

The detailed information on input files is displayed and the configuration can be changed.

Input object module file	Specifies an input object module file. Add one file in one line. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This corresponds to the -Input option of the linker. The specified file name is displayed as the subproperty.	
	Default	Input object module file[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.

Using libraries	<p>Specifies an input library file.</p> <p>The following placeholders are supported.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the product install folder.</p> <p>This corresponds to the -library option of the linker.</p> <p>The library file name is displayed as the subproperty.</p>	
	Default	Input library file [<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.
	Restriction	Up to 259 characters Up to 65536 items can be specified.
System library file	<p>Changes the specified order of the library files which the system set during linking.</p> <p>The specified system library files are displayed.</p> <p>This corresponds to the -library option of the linker.</p>	
	Default	System library file [<i>number of defined items</i>]
	How to change	Edit by the System Include Path Order dialog box which appears when clicking the [...] button.
	Restriction	Changes not allowed (Only the specified order of the system library files can be changed.)
Input binary data file	<p>Specifies an input binary data file.</p> <p>Specifies in the format of "<i>file name (section name:boundary alignment/section attribute,symbol name)</i>", with one file name per line.</p> <p>The "<i>:boundary alignment</i>", "<i>/section attribute</i>", or "<i>,symbol name</i>" part can be omitted.</p> <p>1, 2, 4, 8, 16, or 32 can be specified for the "<i>boundary alignment</i>".</p> <p>When the "<i>boundary alignment</i>" specification is omitted, 1 is used as the default.</p> <p>CODE or DATA can be specified for the "<i>section attribute</i>".</p> <p>When "<i>section attribute</i>" specification is omitted, the write, read, and execute attributes are all enabled by default.</p> <p>The following placeholders are supported.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the product install folder.</p> <p>This corresponds to the -binary option of the linker.</p> <p>This property is displayed only when [Relocatble module file (-FOrM=Relocate)] in the [Output file type] property in the [Output] category is selected.</p>	
	Default	Specify binary data file [<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 32767 characters Up to 65536 items can be specified.

Initiates the prelinker	Selects whether to initiate the prelinker (The automatic generation of C++ template instance). This corresponds to the -noprelink option of the linker.		
	Default	Automatic control	
	How to change	Select from the drop-down list.	
	Restriction	Automatic control	Disables the prelinker initiation if there is no ii file in a file to be input to linker.
		Yes	Enables the prelinker initiation.
		No (-NOPRElink)	Disables the prelinker initiation.

(2) [Output]

The detailed information on output files are displayed and the configuration can be changed.

Output file type	Displays the output file type. This corresponds to the -form option of the linker.		
	Default	User library file (-FOrm=Library=U)	
	How to change	Changes not allowed.	
	Restriction	User library file (-FOrm=Library=U)	Outputs a user library file. This item is not available when [Yes (Notify unused symbol) (-Message -MSg_unused)] in the [Enables information-level message output] property is selected.
		System library file (-FOrm=Library=S)	Outputs a system library file. This item is not available when [Yes (Notify unused symbol) (-Message -MSg_unused)] in the [Enables information-level message output] property is selected.
		Relocatable module file (-FOrm=Relocate)	Outputs a relocatable module file.
Outputs debugging information	Specifies whether debugging information is output. This corresponds to the -nodebug and -debug options of the linker. This property is displayed only when [Relocatable module file (-FOrm=Relocate)] in the [Output file type] property is selected.		
	Default	Yes (Outputs to the output file) (-DEBbug)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (Outputs to the output file) (-DEBbug)	Outputs a debugging information to the output file.
		No (-NODEBbug)	Does not output a debugging information.

Path of the output folder	Specifies path of the output folder. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -output option of the linker.			
	Default	%BuildModeName%		
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.		
	Restriction	Up to 247 characters		
Output file name	Specifies an output file name. The default extensions depends on [Output file type] property when extension omitted. The default extensions are as follows: "User library file (-FOrm=Library=U)" : .lib "System library file (-FOrm=Library=S)" : .lib "Relocatble module file (-FOrm=Relocate)" : .rel. The following placeholders are supported. %ProjectName%: Replaces with the project name. This corresponds to the -output option of the linker.			
	Default	When [User library file (-FOrm=Library=U)] in the [Output file type] property is selected %ProjectName%.lib When [System library file (-FOrm=Library=S)] in the [Output file type] property is selected %ProjectName%.lib When [Relocatble module file (-FOrm=Relocate)] in the [Output file type] property is selected %ProjectName%.rel		
	How to change	Directly enter in the text box.		
	Restriction	Up to 259 characters		
Enables information-level message output	Specifies whether to enable information-level message output. This corresponds to the -message , -msg_unused , and -nomessage options of the linker.			
	Default	No (-NOMessage)		
	How to change	Select from the drop-down list.		
	Restriction	Yes (-Message)	Outputs information level messages.	
		Yes (Notify unused symbol) (-Message -MSg_unused)	Notifies the user of the externally defined symbol which is not referenced. This item is available only when [Relocatble module file (-FOrm=Relocate)] in the [Output file type] property is selected.	
	No (-NOMessage)	Disables the output of information level messages.		

Suppresses the number of information-level messages	Specifies suppresses the number of information-level messages. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -nomessage option of the linker. This property is displayed only when [No (-NOMessage)] in the [Enables information-level message output] property is specified.	
	Default	Blank
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 32767 characters

(3) [List]

The detailed information on the list are displayed and the configuration can be changed.

Outputs the linkage list file	Selects whether to output the linkage list file. This corresponds to the -list and -show option of the linker.		
	Default	Yes (List contents=specify) (-LISt)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (List contents=not specify) (-LISt -SHow)	Outputs the default information associated with a output file type to a linkage list file.
		Yes (List contents=ALL) (-LISt -SHow=ALL)	Outputs all information associated with a output file type to a linkage list file.
		Yes (List contents=specify) (-LISt)	Outputs the specified information to a linkage list file.
No		Does not output a linkage list file.	
Outputs a symbol name list in a module	Selects whether to output a symbol name list in a module. This corresponds to the -show option of the linker. This property is displayed only when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-SHow=SYmbo)	Outputs a symbol name list in a module.
		No	Does not output a symbol name list in a module.

Outputs a section list in a module	Selects whether to output a section list in a module. This corresponds to the -show option of the linker. This property is displayed only when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property and [User library file (-FOrm=Library=U)] or [System library file (-FOrm=Library=S)] in the [Output file type] property in the [Output] category is selected.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-SHow=SEction) Outputs a section list in a module. No Does not output a section list in a module.
Outputs the cross-reference information	Selects whether to output the cross-reference information. This corresponds to the -show option of the linker. This property is displayed only when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property and [Relocatble module file (-FOrm=Relocate)] in the [Output file type] property in the [Output] category is selected.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-SHow=Xreference) Outputs the cross reference information to the linkage list file. No Does not output the cross reference information to the linkage list file.
Shows the total sizes of sections	Selects whether to show the total sizes of sections. This corresponds to the -show option of the linker. This property is displayed only when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property and [Relocatble module file (-FOrm=Relocate)] in the [Output file type] property in the [Output] category is selected.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-SHow=Total_size) Shows the total sizes of sections allocated to the ROM and RAM areas. No Does not show the total sizes of sections.
Outputs vector information	Selects whether to output vector information. This corresponds to the -show option of the linker. This property is displayed only when [Yes (List contents=specify) (-LISt)] in the [Outputs the linkage list file] property and [Relocatble module file (-FOrm=Relocate)] in the [Output file type] property in the [Output] category is selected.	
	Default	No
	How to change	Select from the drop-down list.
	Restriction	Yes (-SHow=VECTOR) Outputs vector information to the linkage list file. No Does not output vector information to the linkage list file.

- (4) [Others]
Other detailed information on library generators are displayed and the configuration can be changed.

Reduces the memory occupancy	Selects whether to reduce the memory occupancy. This corresponds to the -memory option of the linker. This property is not displayed when any one of the following items is selected. [User library file (-FOrm=Library=U)] or [System library file (-FOrm=Library=S)] in the [Output file type] property in the [Output] category and [Yes (-Hide)] in the [Deletes local symbol name information] property [System library file (-FOrm=Relocate)] in the [Output file type] property in the [Output] category			
	Default	No (-MEMory=High)		
	How to change	Select from the drop-down list.		
	Restriction	Yes (-MEMory=Low)	Reduces the memory occupancy.	
		No (-MEMory=High)	Does not reduce the memory occupancy.	
Changes the warning-level messages to information-level messages	Selects whether to change the warning-level messages to information-level messages. This corresponds to the -change_message option of the linker.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes(All) (-change_message=information)	Changes all warning-level messages to the information-level messages.	
		Yes(Specifies error number) (-change_message=information=<Error-Number>)	Changes the warning-level messages with the specified error numbers to the information-level messages.	
No		Does not change the warning-level messages to the information-level messages.		
Error number of warning-level message	Specifies error number of warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the linker. This property is displayed only when [Yes(Specifies error number) (-change_message=information=<ErrorNumber>)] in the [Changes the warning-level messages to information-level messages] property is specified.			
	Default	Blank		
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.		
	Restriction	Up to 32767 characters		

Changes the information-level messages to warning-level messages	Selects whether to change the information-level messages to warning-level messages. This corresponds to the -change_message option of the linker.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=warning)	Changes all information-level messages to warning-level messages.
		Yes(Specifies error number) (-change_message=warning=<Error-Number>)	Changes the information-level messages with the specified error numbers to warning-level messages.
No		Does not change the information-level messages to warning-level messages.	
Error number of information-level message	Specifies error number of information-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the linker. This property is displayed only when [Yes(Specifies error number) (-change_message=warning=<ErrorNumber>)] in the [Changes the information-level messages to warning-level messages] property is specified.		
	Default	Blank	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Changes the information-level and warning-level messages to error-level messages	Selects whether to change the information-level and warning-level messages to error-level messages. This corresponds to the -change_message option of the linker.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=error)	Changes all information-level and warning-level messages to error-level messages.
		Yes(Specifies error number) (-change_message=error=<ErrorNumber>)	Changes the information-level and warning-level messages with the specified error numbers to error-level messages.
No		Does not change the warning-level messages to information-level messages.	

Error number of information-level and warning-level message	Specifies error number of information-level and warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the linker. This property is displayed only when [Yes(Specifies error number) (-change_message=error=<ErrorNumber>)] in the [Changes the information-level and warning-level messages to error-level messages] property is specified.		
	Default	Blank	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Deletes local symbol name information	Selects whether to delete local symbol name information. This corresponds to the -hide option of the linker.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-Hide)	Deletes local symbol name information.
	No	Does not delete local symbol name information.	
Displays the total sizes of sections	Selects whether to display the total sizes of sections. This corresponds to the -total_size option of the linker. This property is displayed only when [Relocatable module file (-FOrm=Relocate)] in the [Output file type] property in the [Output] category is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-Total_size)	Displays the total sizes of sections.
	No	Does not display the total sizes of sections.	
Displays the copyright information	Selects whether to display the copyright information. This corresponds to the -logo and -nologo option of the linker.		
	Default	No (-NOLOgo)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-LOgo)	Displays the total sizes of sections.
	No (-NOLOgo)	Does not display the total sizes of sections.	

Commands executed before librarian processing	<p>Specify the command to be executed before librarian processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %LibrarianFile%: Replaces with the absolute path of the output file under librarian processing. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before librarian processing.</p> <p>The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>	
	Default	Commands executed before librarian processing[<i>number of defined items</i>]
	How to change	<p>Edit by the Text Edit dialog box which appears when clicking the [...] button.</p> <p>For the subproperty, you can use the text box directly enter the text.</p>
	Restriction	<p>Up to 1023 characters Up to 64 items can be specified.</p>

Commands executed after librarian processing	<p>Specify the command to be executed after librarian processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %LibrarianFile%: Replaces with the absolute path of the output file under librarian processing. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after librarian processing. The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>	
	Default	Commands executed after librarian processing[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.
Other additional options	<p>Inputs the link options to be added additionally. The options set here are added at the end of the link options group.</p>	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	The specified option is displayed.	
	Default	Command line[<i>number of defined items</i>]
	How to change	Changes not allowed

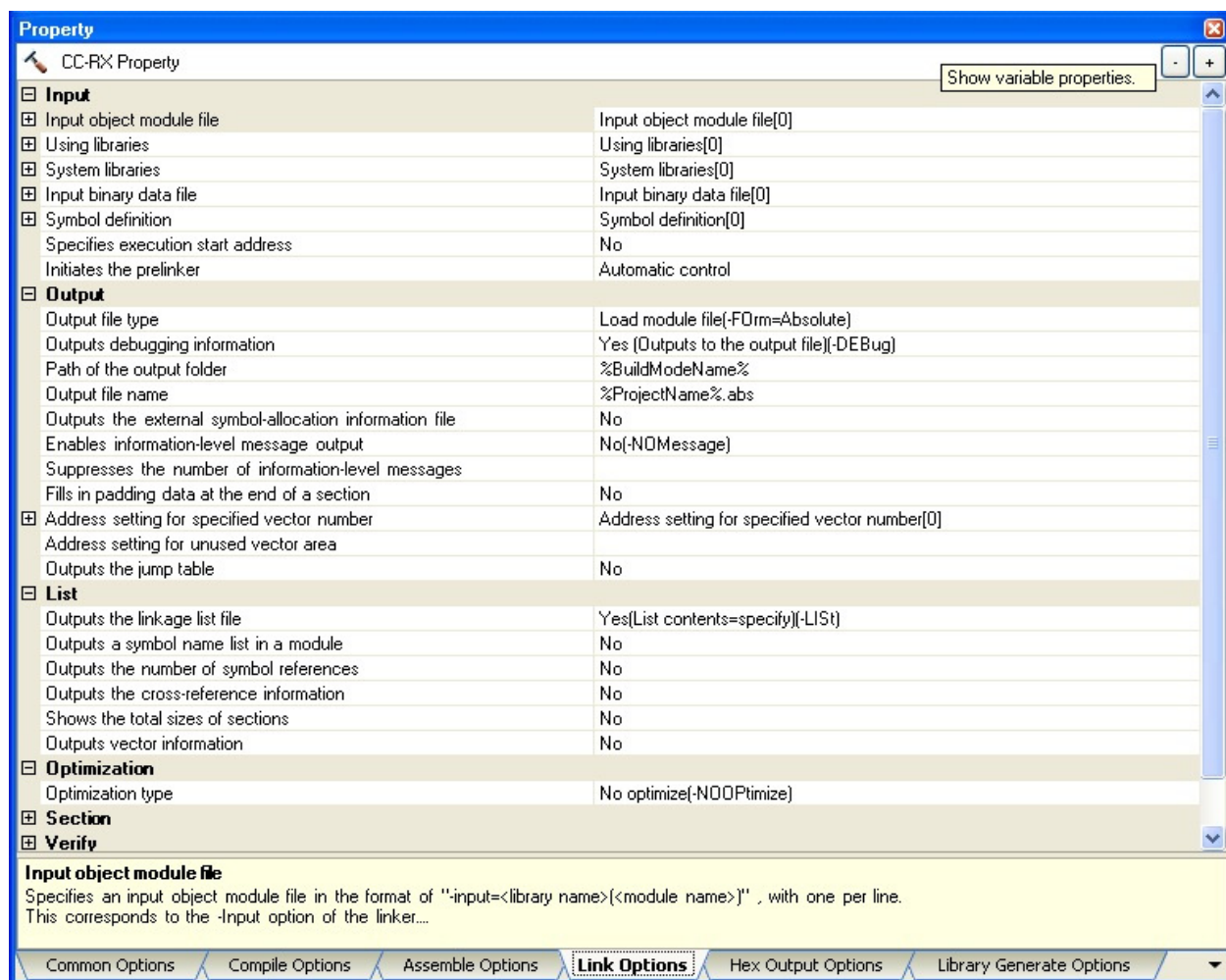
[Hex Output Options] tab

This tab shows the detailed information on the link phase categorized by the following and the configuration can be changed.

- (1) [Output File]
- (2) [Hex Format]
- (3) [Others]

Caution This tab is not displayed for the library project.

Figure A.9 Property Panel: [Link Options] Tab



[Description of each category]

- (1) [Output File]

The detailed information on output files is displayed and the configuration can be changed.

Output hex file	Specifies whether hex file is output.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Outputs a hex file.
		No	Does not output a hex file.

Output folder	<p>Specifies path of the output folder.</p> <p>The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder.</p> <p>%ActiveProjectName%: Replaces with the active project name.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%MainProjectDir%: Replaces with the absolute path of the main project folder.</p> <p>%MainProjectName%: Replaces with the main project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.</p> <p>%OutputDir%: Replaces with the absolute path of the output folder.</p> <p>%OutputFile%: Replaces with the absolute path of the output file.</p> <p>%ProjectDir%: Replaces with the absolute path of the project folder.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%TempDir%: Replaces with the absolute path of the temporary folder.</p> <p>%WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>If this is blank, it is assumed that the project folder has been specified.</p> <p>This corresponds to the -output option of the linker.</p> <p>This property is displayed only when [Yes] in the [Output hex file] property is selected.</p>	
	Default	%BuildModeName%
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.
	Restriction	Up to 247 characters
Output file name	<p>Specifies the output file name.</p> <p>The default extensions depends on [Load module file convert format] property when extension omitted.</p> <p>The default extensions are as follows:</p> <p>"Hex file (-Form=Hexadecimal)" : .hex</p> <p>"S record file (-Form=Stype)" : .mot</p> <p>"Binary data file (-Form=Binary)" : .bin.</p> <p>The following placeholders are supported.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.</p> <p>This corresponds to the -output option of the linker.</p> <p>This property is displayed only when [Yes] in the [Output hex file] property is selected.</p>	
	Default	<p>- When [Hex file (-Form=Hexadecimal)] in the [Load module file convert format] property is selected</p> <p>%ProjectName%.hex</p> <p>- When [S record file (-Form=Stype)] in the [Load module file convert format] property is selected</p> <p>%ProjectName%.mot</p> <p>- When [Binary data file (-Form=Binary)] in the [Load module file convert format] property is selected</p> <p>%ProjectName%.bin</p>
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

Division output file	<p>Specifies the division conversion file.</p> <p>Specifies in the format of "file name=start address-end address" or "file name=section name", with one file name per line.</p> <p>Specifies an address in the hexadecimal notation (example: file2.mot=400-ffff).</p> <p>To define multiple sections, use a colon to separate each entry written, as in "file name=section name:section name" (example: file1.mot=stack:istack).</p> <p>The default extensions depends on [Load module file convert format] property when extension omitted.</p> <p>The default extensions are as follows:</p> <p>"Hex file (-Form=Hexadecimal)" : .hex</p> <p>"S record file (-Form=Stype)" : .mot</p> <p>"Binary data file (-Form=Binary)" : .bin.</p> <p>The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder.</p> <p>%ActiveProjectName%: Replaces with the active project name.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%MainProjectDir%: Replaces with the absolute path of the main project folder.</p> <p>%MainProjectName%: Replaces with the main project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.</p> <p>%OutputDir%: Replaces with the absolute path of the output folder.</p> <p>%OutputFile%: Replaces with the absolute path of the output file.</p> <p>%ProjectDir%: Replaces with the absolute path of the project folder.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%TempDir%: Replaces with the absolute path of the temporary folder.</p> <p>%WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>This corresponds to the -output option of the linker.</p> <p>This property is displayed only when [Yes] in the [Output hex file] property is selected.</p>	
	Default	Division conversion file[number of defined items]
	How to change	<p>Edit by the Text Edit dialog box which appears when clicking the [...] button.</p> <p>For the subproperty, you can use the text box directly enter the text.</p>
	Restriction	<p>Up to 255 characters</p> <p>Up to 65536 items can be specified.</p>

(2) [Hex Format]

The detailed information on output files are displayed and the configuration can be changed.

Hex file format	Selects the load module file convert format. This corresponds to the -form option of the linker.		
	Default	S record file (-FOrm=Stype)	
	How to change	Select from the drop-down list.	
	Restriction	Intel expanded hex file (-FOrm=Hexadecimal)	Outputs a Intel expanded hex file.
		Motorola S type file (-FOrm=Stype)	Outputs a Motorola S-type file.
		Binary file (-FOrm=Binary)	Outputs a binary file.

Unifies the record size	Selects whether to output data with the specified data record regardless of the address range. This corresponds to the -record option of the linker. This property is displayed only when [Hex file (-FOrm=Hexadecimal)] in the [Hex file format] property is selected.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes (HEX record) (-REcord=H16)	Outputs the HEX record for the data records.	
		Yes (Expansion HEX record) (-REcord=H20)	Outputs the Expansion HEX record for the data records.	
Yes (32-bit HEX record) (-REcord=H32)		Outputs the 32-bit HEX record for the data records.		
No		Outputs various data records according to each address.		
Unifies the record size	Selects whether to output data with the specified data record regardless of the address range. This corresponds to the -record option of the linker. This property is displayed only when [S record file (-FOrm=Stype)] in the [Hex file format] property is selected.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes (S1 record) (-REcord=S1)	Outputs the S1 record for the data records.	
		Yes (S2 record) (-REcord=S2)	Outputs the S2 record for the data records.	
Yes (S3 record) (-REcord=S3)		Outputs the S3 record for the data records.		
No		Outputs various data records according to each address.		
Fills the unused areas in the output ranges with the value	Selects whether to fill the unused areas in the output ranges with the value. This corresponds to the -space option of the linker. When [Not convert] in the [Load module file convert format] property is selected or a convert file name in the [Division conversion file] property is not specified, this property is not displayed.			
	Default	Division conversion file[number of defined items]		
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.		
	Restriction	Up to 32767 characters Up to 1024 items can be specified.		

Fills the unused areas in the output ranges with the value	Selects whether to fill the unused areas in the output ranges with the value. This corresponds to the -space option of the linker. When [Not convert] in the [Load module file convert format] property is selected or a convert file name in the [Division conversion file] property is not specified, this property is not displayed.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (Random) (-SPace=Random)	Fills the unused areas in the output ranges with random values.
		Yes (Specification value) (-SPace=<numerical value>)	Fills the unused areas in the output ranges with user-specified hexadecimal value.
	No	Does not fill the unused areas in the output ranges with data.	
Output padding data	Specifies the output padding data. This corresponds to the -space option of the linker. This property is displayed only when [Yes (Specification value) (-SPace=<numerical value>)] in the [Fills the unused areas in the output ranges with the value] property is specified.		
	Default	FF (hexadecimal number)	
	How to change	Directly enter in the text box.	
	Restriction	0 to FFFFFFFF (hexadecimal number)	
Specifies byte count for data record	Selects whether to specify the maximum byte count for a data record. This corresponds to the -byte_count option of the linker. This property is displayed only when [Hex file (-FOrm=Hexadecimal)] in the [Load module file convert format] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-BYte_count)	Specifies the maximum byte count for a data.
		No	Does not specify the byte count for a data.
Maximum byte count for a data record	Specifies the maximum byte count for a data record. This corresponds to the -byte_count option of the linker. This property is displayed only when [Yes(-BYte_count)] in the [Specifies byte count for data record] property is specified.		
	Default	FF (hexadecimal number)	
	How to change	Directly enter in the text box.	
	Restriction	1 to FF (hexadecimal number)	

Outputs the calculation result of CRC	Selects whether to generate CRC code. This corresponds to the -crc option of the linker. This property is displayed only when [Hex file (-Form=Hexadecimal)] or [S record file (-Form=Stype)] in the [Load module file convert format] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (Polynomial expression: CRC-CCITT, Endian: Automatic) (-CRc)	Selects CRC-CCITT as a polynomial expression.
		Yes (Polynomial expression: CRC-CCITT, Endian: Big-endian data) (-CRc)	Selects the CRC-CCITT as a polynomial expression and selects the BIG as an endian.
		Yes (Polynomial expression: CRC-CCITT, Endian: Little-endian data) (-CRc)	Selects the CRC-CCITT as a polynomial expression and selects the LITTLE as an endian.
		Yes (Polynomial expression: CRC-16, Endian: Automatic) (-CRc)	Selects CRC-16 as a polynomial expression.
		Yes (Polynomial expression: CRC-16, Endian: Big-endian data)(-CRc)	Selects the CRC-16 as a polynomial expression and selects the BIG as an endian.
		Yes (Polynomial expression: CRC-16, Endian: Little-endian) data (-CRc)	Selects the CRC-16 as a polynomial expression and selects the LITTLE as an endian.
No	Does not generate the CRC code.		
Output address	Specifies the address where the result is output. Specifies an address in hexadecimal. This corresponds to the -crc option of the linker. This property is not displayed when [No] in the [Outputs the calculation result of CRC] property is specified.		
	Default	0	
	How to change	Directly enter in the text box.	
	Restriction	0 to FFFFFFFF (hexadecimal number)	
Target range	Specifies the target range. Specifies in the format of " <i>start address-end address</i> ". Specifies an address in the hexadecimal notation (example: 400-ffff/) This corresponds to the -crc option of the linker. This property is not displayed when [No] in the [Outputs the calculation result of CRC] property is specified.		
	Default	Target range[<i>number of defined items</i>]	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	

Outputs the S9 record at the end	Selects whether to output the S9 record at the end. This corresponds to the -s9 option of the linker. This property is displayed only when [S record file (-FOrm=Stype)] in the [Load module file convert format property is specified.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-S9)	Outputs the S9 record at the end.
No		Does not output the S9 record at the end.	

(3) [Others]

Other detailed information on linking are displayed and the configuration can be changed.

Other additional options	Inputs the link options to be added additionally. The options set here are added at the end of the link options group.	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	The specified option is displayed.	
	Default	Command line[<i>number of defined items</i>]
	How to change	Changes not allowed

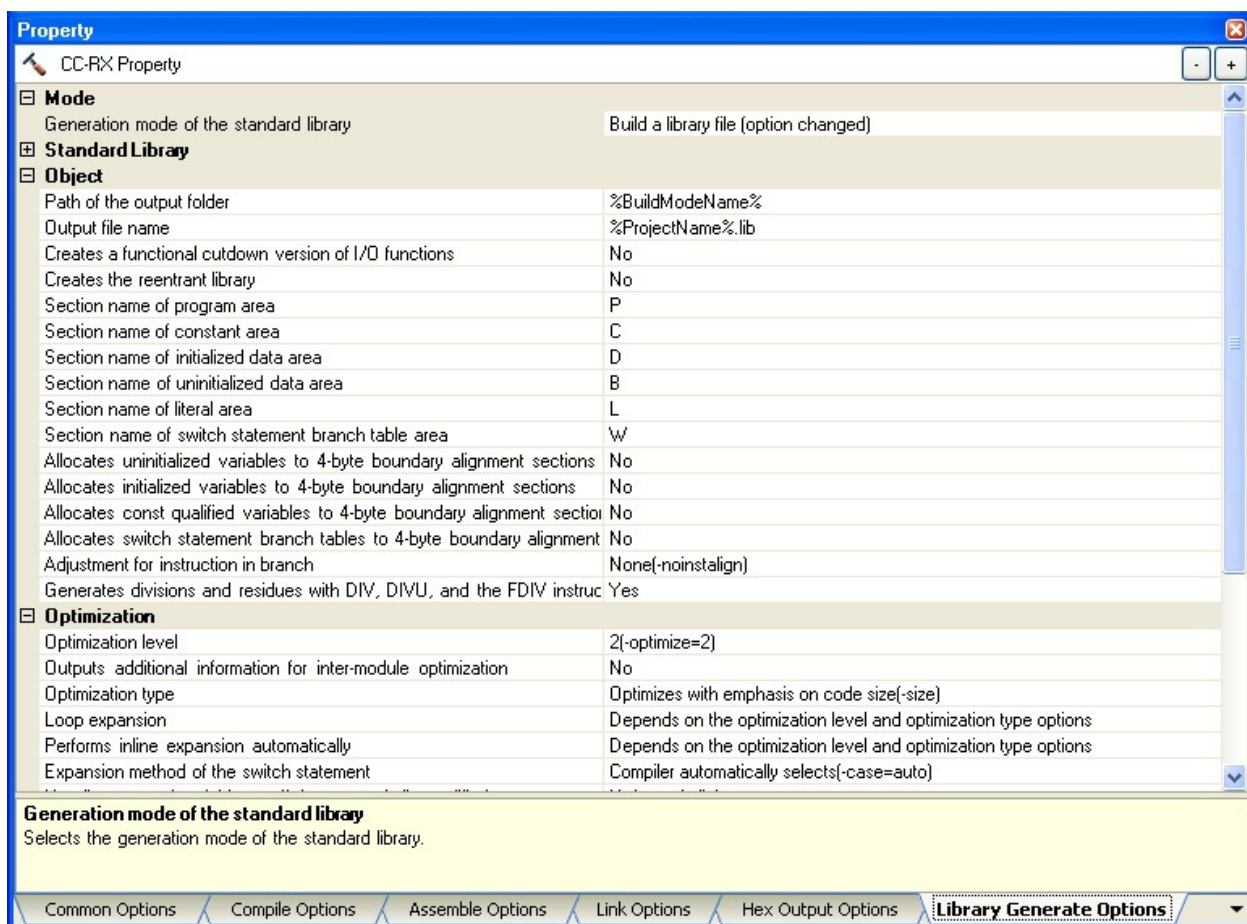
[Library Generate Options] tab

This tab shows the detailed information on the library generate phase categorized by the following and the configuration can be changed.

- (1) [Mode]
- (2) [Standard Library]
- (3) [Object]
- (4) [Optimization]
- (5) [Others]

Caution This tab is not displayed for the library project.

Figure A.10 Property Panel: [Library Generate Options] Tab



[Description of each category]

- (1) [Mode]

The detailed information on modes are displayed and the configuration can be changed.

Generation mode of the standard library	Select the generation mode of the standard library.		
	Default	Build a library file (option changed)	
	How to change	Select from the drop-down list.	
	Restriction	Build a library file (any-time)	Creates a latest standard library file.
		Build a library file (option changed)	Creates a latest standard library file only when an option is changed regardless of whether Build or Rebuild.
		Do not add a library file	Does not add a library file.

(2) [Standard Library]

The detailed information on standard library are displayed and the configuration can be changed.

This category is not displayed when [Do not add a library file] in the [Generation mode of the standard library] property in the [Mode] category.

Library configuration	Selects which functions are to be usable in the C standard library. This corresponds to the -lang option of the library generator.		
	Default	C(C89) (-lang=c)	
	How to change	Select from the drop-down list.	
	Restriction	C(C89) (-lang=c)	Includes only the functions conforming to the C89 standard in the C standard library.
		C99 (-lang=c99)	Includes the functions conforming to the C89 standard and the functions conforming to the C99 standard in the C standard library.
Configuration library	Selects the configuration library. This corresponds to the -head option of the library generator.		
	Default	Custom (-head=<SubOption>)	
	How to change	Select from the drop-down list.	
	Restriction	Custom (-head=<SubOption>)	Specifies a configuration library.
		All enable (-head=all)	Specifies all library functions and runtime library.
		All disable (-head=runtime)	Does not specify a configuration library.
Enables runtime library	Selects whether to enable runtime library. This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	Yes (-head=runtime)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=runtime)	Enables the runtime library.
		No	Disables the runtime library.

Enables ctype.h(C89/C99)	Selects whether to enable ctype.h(C89/C99). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=ctype)	Enables the ctype.h (C89/C99) and run-time library.
		No	Disables the ctype.h (C89/C99).
Enables math.h(C89/C99)	Selects whether to enable math.h(C89/C99). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=math)	Enables the math.h (C89/C99) and run-time library.
		No	Disables the math.h (C89/C99).
Enables mathf.h(C89/C99)	Selects whether to enable mathf.h(C89/C99). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=mathf)	Enables the mathf.h (C89/C99) and run-time library.
		No	Disables the mathf.h (C89/C99).
Enables stdarg.h(C89/C99)	Selects whether to enable stdarg.h(C89/C99). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=stdarg)	Enables the stdarg.h (C89/C99) and run-time library.
		No	Disables the stdarg.h (C89/C99).

Enables stdio.h(C89/C99)	Selects whether to enable stdio.h(C89/C99). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	Yes (-head=stdio)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=stdio)	Enables the stdio.h (C89/C99) and run-time library.
	No	Disables the stdio.h (C89/C99).	
Enables stdlib.h(C89/C99)	Selects whether to enable stdlib.h(C89/C99). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	Yes (-head=stdlib)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=stdlib)	Enables the stdlib.h (C89/C99) and run-time library.
	No	Disables the stdlib.h (C89/C99).	
Enables string.h(C89/C99)	Selects whether to enable string.h(C89/C99). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	Yes (-head=string)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=string)	Enables the string.h (C89/C99) and run-time library.
	No	Disables the string.h (C89/C99).	
Enables ios(EC++)	Selects whether to enable ios(EC++). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=ios)	Enables the ios(EC++) and runtime library.
	No	Disables the ios(EC++).	

Enables new(EC++)	Selects whether to enable new(EC++). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	Yes (-head=new)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=new)	Enables the new(EC++) and runtime library.
	No	Disables the new(EC++).	
Enables complex(EC++)	Selects whether to enable complex(EC++). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=complex)	Enables the complex(EC++) and runtime library.
	No	Disables the complex(EC++).	
Enables string(EC++)	Selects whether to enable string(EC++). This corresponds to the -head option of the library generator. This property is displayed only when [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=cppstring)	Enables the string(EC++) and runtime library.
	No	Disables the string(EC++).	
Enables complex.h(C99)	Selects whether to enable complex.h(C99). This corresponds to the -head option of the library generator. This property is displayed only when [C99 (-lang=c99)] in the [Library configuration] property and [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=C99_complex)	Enables the complex.h(C99) and runtime library.
	No	Disables the complex.h(C99).	

Enables fenv.h(C99)	Selects whether to enable fenv.h(C99). This corresponds to the -head option of the library generator. This property is displayed only when [C99 (-lang=c99)] in the [Library configuration] property and [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=fenv)	Enables the fenv.h(C99) and runtime library.
	No	Disables the fenv.h(C99).	
Enables inttypes.h(C99)	Selects whether to enable inttypes.h(C99). This corresponds to the -head option of the library generator. This property is displayed only when [C99 (-lang=c99)] in the [Library configuration] property and [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=inttypes)	Enables the inttypes.h(C99) and runtime library.
	No	Disables the inttypes.h(C99).	
Enables wchar.h(C99)	Selects whether to enable wchar.h(C99). This corresponds to the -head option of the library generator. This property is displayed only when [C99 (-lang=c99)] in the [Library configuration] property and [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=wchar)	Enables the wchar.h(C99) and runtime library.
	No	Disables the wchar.h(C99).	
Enables wctype.h(C99)	Selects whether to enable wctype.h(C99). This corresponds to the -head option of the library generator. This property is displayed only when [C99 (-lang=c99)] in the [Library configuration] property and [Custom (-head=<SubOption>)] in the [Configuration library] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-head=wctype)	Enables the wctype.h(C99) and runtime library.
	No	Disables the wctype.h(C99).	

(3) [Object]

The detailed information on output files are displayed and the configuration can be changed.

This category is not displayed when [Do not add a library file] in the [Generation mode of the standard library] property in the [\[Mode\]](#) category.

Path of the output folder	Specifies path of the output folder. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -output option of the library generator.		
	Default	%BuildModeName%	
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.	
	Restriction	Up to 247 characters	
Output file name	Specify the output file name. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. The reference point of the path is the project folder. This corresponds to the -output option of the library generator.		
	Default	%ProjectName%.lib	
	How to change	Directly enter to the text box.	
	Restriction	Up to 32767 characters	
Generation mode of the standard library	Selects whether to create a functional cutdown version of I/O functions. This corresponds to the -nofloat and -simple_stdio option of the library generator.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (Functional cutdown version 1) (-nofloat)	Creates simple I/O functions that do not support the conversion of floating-point numbers (%f, %e, %E, %g, %G).
		Yes (Functional cutdown version 2) (-simple_stdio)	Does not include the conversion of the floating point numbers, the conversion of long long type, and the conversion of 2-byte code.
		No	Does not create a functional cutdown version of I/O functions.
Section name of program area	Specifies the section name of program area. This corresponds to the -section option of the compiler.		
	Default	P	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	

Section name of constant area	Specifies the section name of constant area. This corresponds to the -section option of the compiler.		
	Default	C	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of initialized data area	Specifies the section name of initialized data area. This corresponds to the -section option of the compiler.		
	Default	D	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of uninitialized data area	Specifies the section name of uninitialized data area. This corresponds to the -section option of the compiler.		
	Default	B	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of literal area	Specifies the section name of literal area. This corresponds to the -section option of the compiler.		
	Default	L	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of switch statement branch table area	Specifies the section name of switch statement branch table area. This corresponds to the -section option of the compiler.		
	Default	W	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Allocates uninitialized variables to 4-byte boundary alignment sections	Selects whether to allocate uninitialized variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=B)	Allocates uninitialized variables to 4-byte boundary alignment sections.
		No	Does not allocate uninitialized variables to 4-byte boundary alignment sections.

Allocates initialized variables to 4-byte boundary alignment sections	Selects whether to allocate initialized variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=D)	Allocates initialized variables to 4-byte boundary alignment sections.
Allocates const qualified variables to 4-byte boundary alignment sections	Restriction	No	Does not allocate initialized variables to 4-byte boundary alignment sections.
	Default	No	
	How to change	Select from the drop-down list.	
Allocates switch statement branch tables to 4-byte boundary alignment sections	Restriction	Yes (-nostuff=C)	Allocates const qualified variables to 4-byte boundary alignment sections.
		No	Does not allocate const qualified variables to 4-byte boundary alignment sections.
	Default	No	
	How to change	Select from the drop-down list.	
Allocates switch statement branch tables to 4-byte boundary alignment sections	Restriction	Yes (-nostuff=W)	Allocates switch statement branch tables to 4-byte boundary alignment sections.
		No	Does not allocate switch statement branch tables to 4-byte boundary alignment sections.
	Default	No	
	How to change	Select from the drop-down list.	

Adjustment for instruction in branch	Selects adjustment for instruction in branch. This corresponds to the <code>-noinstalign</code> , <code>-instalign4</code> , and <code>-instalign8</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Execution in 4 bytes (-instalign4)	Aligns instructions at branch destinations to 4-byte boundaries.
		Execution in 4 bytes(Contains each loop head) (-instalign4=loop)	Aligns instructions at branch destinations to 4-byte boundaries (Contains head of each loop).
		Execution in 4 bytes(Contains each inmost loop head) (-instalign4=inmost loop)	Aligns instructions at branch destinations to 4-byte boundaries (Contains head of each inmost loop).
		Execution in 8 bytes (-instalign8)	Aligns instructions at branch destinations to 8-byte boundaries.
		Execution in 8 bytes (Contains each loop head) (-instalign8=loop)	Aligns instructions at branch destinations to 8-byte boundaries (Contains head of each loop).
		Execution in 8 bytes (Contains each inmost loop head) (-instalign8=inmost loop)	Aligns instructions at branch destinations to 8-byte boundaries (Contains head of each inmost loop).
		None (-noinstalign)	Does not align instructions at branch destinations.
Generates divisions and residues with DIV, DIVU, and the FDIV instruction	Selects whether to generate divisions and residues with DIV, DIVU, and the FDIV instruction. This corresponds to the <code>-nouse_div_inst</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Generates code in which DIV, DIVU, or FDIV instructions are used.
		No (-nouse_div_inst)	Generates code in which no DIV, DIVU, or FDIV instructions are used.

(4) [Optimization]

The detailed information on optimizations are displayed and the configuration can be changed.

This category is not displayed when [Do not add a library file] in the [Generation mode of the standard library] property in the [Mode] category.

Optimization level	Selects optimization level. This corresponds to the -optimize option of the library generator.		
	Default	2 (-optimize=2)	
	How to change	Select from the drop-down list.	
	Restriction	0 (-optimize=0)	Does not optimize the program.
		1 (-optimize=1)	Partially optimizes the program by automatically allocating variables to registers, integrating the function exit blocks, integrating multiple instructions which can be integrated, etc.
		2 (-optimize=2)	Performs overall optimization.
		Max (-optimize=max)	Performs optimization as much as possible.
Outputs additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. This corresponds to the -goptimize option of the library generator.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.
		No	Does not outputs additional information for inter-module optimization.
Optimization type	Selects optimization type. This corresponds to the -speed and -size option of the library generator.		
	Default	Optimizes with emphasis on code size (-size)	
	How to change	Select from the drop-down list.	
	Restriction	Optimizes with emphasis on execution performance (-speed)	Optimizes with emphasis on execution performance.
		Optimizes with emphasis on code size (-size)	Optimizes with emphasis on code size.
Loop expansion	Selects whether to optimize the loop expansion (for, while, and do-while). This corresponds to the -loop option of the library generator.		
	Default	Depends on the optimization level and optimization type options	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level and optimization type options	Depends on the optimization level and optimization type options.
		Expansion (-loop=<numeric value>)	Expands loop statements (for, while, and do-while).

Expansion maximum number	Specifies expansion maximum number. This corresponds to the suboption of -loop option of the library generator. This property is displayed only when [Expansion (-loop=<numeric value>)] in the [Loop expansion] property is selected.		
	Default	2 (decimal number)	
	How to change	Directly enter in the text box.	
	Restriction	1 to 32 (decimal number)	
Performs inline expansion automatically	Selects whether to perform inline expansion automatically. This option corresponds to the -inline and -noinline option of the library generator.		
	Default	Depends on the optimization level and optimization type options	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level and optimization type options	Depends on the optimization level and optimization type options.
		Yes (-inline=<numeric value>)	Performs inline expansion automatically.
		No (-noinline)	Does not perform inline expansion automatically.
Maximum increasing rate of function size	Specify maximum increasing rate of function size. This option corresponds to the -inline option of the library generator. This property is displayed only when [Yes (-inline=<numeric value>)] in the [Performs inline expansion automatically] property is selected.		
	Default	100 (decimal number)	
	How to change	Select from the drop-down list.	
	Restriction	1 to 65535 (decimal number)	
Expansion method of the switch statement	Selects expansion method of the switch statement. This corresponds to the -case option of the library generator.		
	Default	Compiler automatically selects (-case=auto)	
	How to change	Select from the drop-down list.	
	Restriction	if_then method (-case=ifthen)	Expands the switch statement using the if_then method.
		Jumping to a table method (-case=table)	Expands the switch statement by using the table method.
		Compiler automatically selects (-case=auto)	Automatically selects the if_then method or table method.

Handles external variables as if they are volatile qualified	Selects whether to handle all external variables as if they are volatile qualified. This corresponds to the <code>-volatile</code> and <code>-novolatile</code> option of the library generator.		
	Default	No (<code>-novolatile</code>)	
	How to change	Select from the drop-down list.	
	Restriction	Yes (<code>-volatile</code>)	Handles all external variables as if they were volatile qualified.
		No (<code>-novolatile</code>)	Does not handle external variables as if they were volatile qualified.
Performs the constant propagation of const qualified external variables	Selects whether to perform the constant propagation of const qualified external variables. This corresponds to the <code>-const_copy</code> and <code>-noconst_copy</code> option of the library generator.		
	Default	Depends on the optimization level options	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level options	Depends on the optimization level options.
		Yes (<code>-const_copy</code>)	Enables constant propagation of const qualified external variables.
		No (<code>-noconst_copy</code>)	Disables constant propagation of const qualified external variables.
Conversion method of the divisions and residues of integer constants	Selects conversion method of the divisions and residues of integer constants. This corresponds to the <code>-const_div</code> and <code>-noconst_div</code> option of the library generator.		
	Default	Depends on the optimization type option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization type option	Depends on the optimization type option
		Instruction sequence using multiplication (<code>-const_div</code>)	Performs constant division (residue) by an instruction sequence using multiplication.
		Instruction sequence using division (<code>-noconst_div</code>)	Performs constant division (residue) by an instruction sequence using division.
Expansion method of the library function	Selects expansion method of the library function. This corresponds to the <code>-library</code> option of the library generator.		
	Default	Performs instruction expansion of several library functions (<code>-library=intrinsic</code>)	
	How to change	Select from the drop-down list.	
	Restriction	Calls library functions (<code>-library=function</code>)	Calls all library functions.
		Performs instruction expansion of several library functions (<code>-library=intrinsic</code>)	Performs instruction expansion for <code>abs()</code> , <code>fabsf()</code> , and library functions which can use string manipulation instructions.

Divides the optimizing ranges into many sections before compilation	Selects whether to divide the optimizing ranges of the large-size function into many sections before compilation. This corresponds to the -scope and -noscope option of the library generator.		
	Default	Depends on the optimization level option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level option	Depends on the optimization level option.
		Yes (-scope)	Divides the optimizing ranges of the large-size function into many sections before compilation.
		No (-noscope)	Does not divide the optimizing ranges before compilation.
Schedules the instruction taking into consideration pipeline processing	Selects whether to schedule the instruction taking into consideration pipeline processing. This corresponds to the -schedule and -noschedule option of the library generator.		
	Default	Depends on the optimization level option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level option	Depends on the optimization level option.
		Yes (-schedule)	Schedules instructions taking into consideration pipeline processing.
		No (-noschedule)	Does not schedule instructions.
Optimizes accesses to external variables	Selects whether to optimize accesses to external variables. This corresponds to the -nomap , -smap and -map option of the library generator.		
	Default	No (-nomap)	
	How to change	Select from the drop-down list.	
	Restriction	No (-nomap)	Disables optimization for accesses to external variables.
		Yes(Optimizes the inner-module) (-smap)	Optimizes accesses to external variables which are defined in the file to be compiled.
		Yes(Optimizes the inter-module) (-map)	Optimizes accesses to external variables.

Converts floating-point constant division into multiplication	Selects whether to convert floating-point constant division into multiplication of the corresponding reciprocals as constants. This corresponds to the -approxdiv option of the library generator.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-approxdiv)	Converts floating-point constant division into multiplication.
		No	Does not convert floating-point constant division into multiplication.
Allocates preferentially the variables with register storage class specification to registers	Selects whether to allocate preferentially the variables with register storage class specification to registers. This corresponds to the -enable_register option of the library generator.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-enable_register)	Allocates preferentially the variables with register storage class specification to registers.
		No	Does not allocate preferentially the variables with register storage class specification to registers.
Omits a check of the range for conversion between the floating type and unsigned integer type	Selects whether to omit a check of the range for conversion between the floating type and unsigned integer type. This corresponds to the -simple_float_conv option of the library generator.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-simple_float_conv)	Omits part of the type conversion processing for the floating type.
		No	Does not omit part of the type conversion processing for the floating type.
Performs optimization considering the type of the data indicated by the pointer	Selects whether to perform optimization considering the type of the data indicated by the pointer. This corresponds to the -alias option of the library generator.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-alias=ansi)	Performs optimization considering the type of the data indicated by the pointer.
		No	Does not perform optimization considering the type of the data indicated by the pointer.

Optimizes modification of the operation order of a floating-point expression	Selects whether to optimize modification of the operation order of a floating-point expression. This corresponds to the <code>-float_order</code> option of the library generator. This option is valid only when [2 (<code>-optimize=2</code>)] or [Max (<code>-optimize=max</code>)] in the [Optimization level] property is specified.				
	Default	No			
	How to change	Select from the drop-down list.			
	Restriction	<table><tr><td>Yes (<code>-float_order</code>)</td><td>Optimizes modification of the operation order in a floating-point expression.</td></tr><tr><td>No</td><td>Does not optimize modification of the operation order in a floating-point expression.</td></tr></table>	Yes (<code>-float_order</code>)	Optimizes modification of the operation order in a floating-point expression.	No
Yes (<code>-float_order</code>)	Optimizes modification of the operation order in a floating-point expression.				
No	Does not optimize modification of the operation order in a floating-point expression.				

(5) [Others]

Other detailed information on library generators are displayed and the configuration can be changed.

Outputs the copyright	Selects whether to output the copyright. This corresponds to the -nologo option of the library generator. This property is not displayed when [Do not add a library file] in the [Generation mode of the standard library] property in the [Mode] category.			
	Default	No (-nologo)		
	How to change	Select from the drop-down list.		
	Restriction	Yes(-logo)	Outputs the copyright.	
		No (-nologo)	Disables output of the copyright.	

Commands executed before library generate processing	<p>Specify the command to be executed before library generator processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %LibraryFile%: Replaces with the absolute path of the output file under the library generator processing. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before library generator processing. The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>	
	Default	Commands executed before library generate processing[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.

Commands executed after library generate processing	<p>Specify the command to be executed after library generator processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %LibraryFile%: Replaces with the absolute path of the output file under the library generator processing. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after library generator processing.</p> <p>The placeholders can be described in the script.</p> <p>The specified command is displayed as the subproperty.</p>	
	Default	Commands executed after library generate processing[<i>number of defined items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.
Other additional options	<p>Input the library generator options to be added additionally. The options set here are added at the end of the librarian options group. This property is not displayed when [Do not add a library file] in the [Generation mode of the standard library] property in the [Mode] category.</p>	
	Default	Blank
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	<p>The specified option is displayed. This property is not displayed when [Do not add a library file] in the [Generation mode of the standard library] property in the [Mode] category.</p>	
	Default	Command line[<i>number of defined items</i>]
	How to change	Changes not allowed

[Build Settings] tab

This tab shows the detailed information on each C source file, C++ source file, assembler source file, object module file, and library file categorized by the following and the configuration can be changed.

(1) [Build]

Figure A.11 Property Panel: [Build Settings] Tab (When Selecting C Source File)

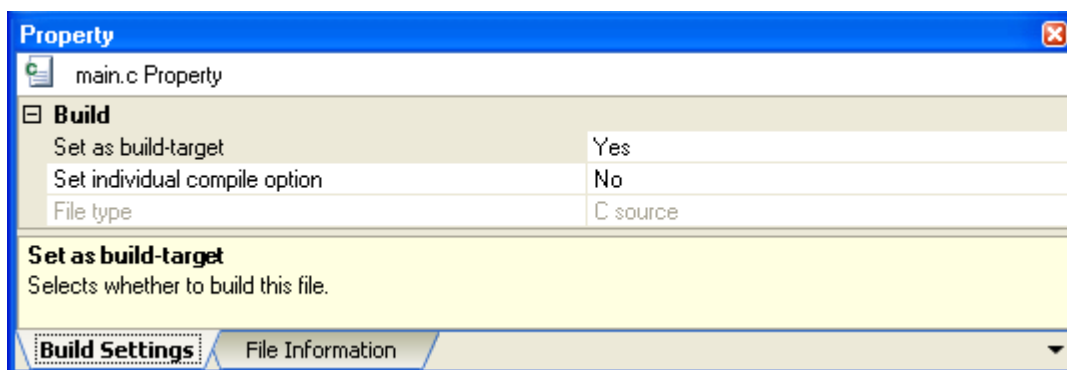


Figure A.12 Property Panel: [Build Settings] Tab (When Selecting C++ Source File)

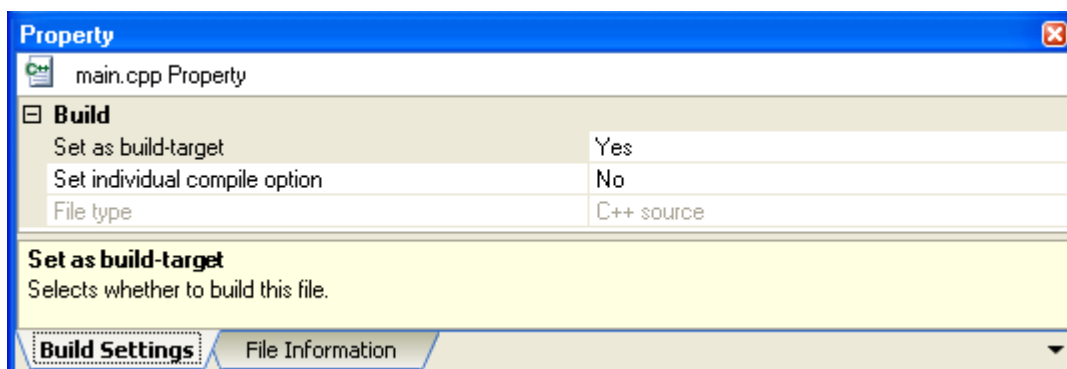


Figure A.13 Property Panel: [Build Settings] Tab (When Selecting Assembler Source File)

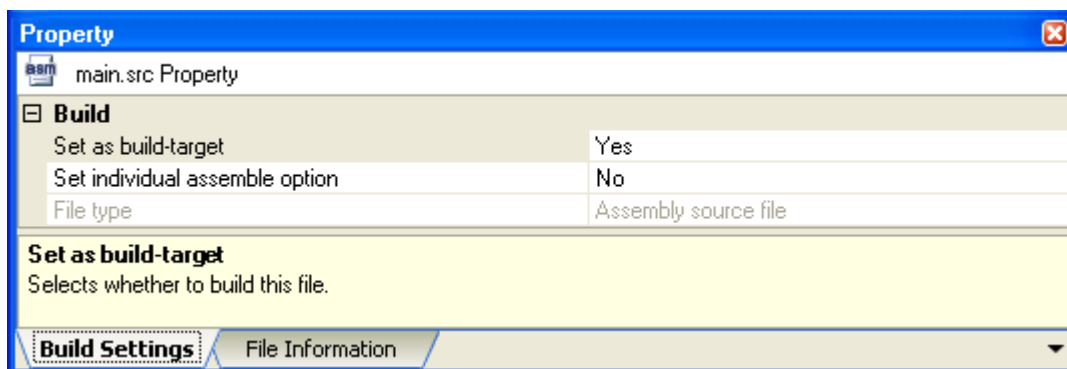


Figure A.14 Property Panel: [Build Settings] Tab (When Selecting Object Module File)

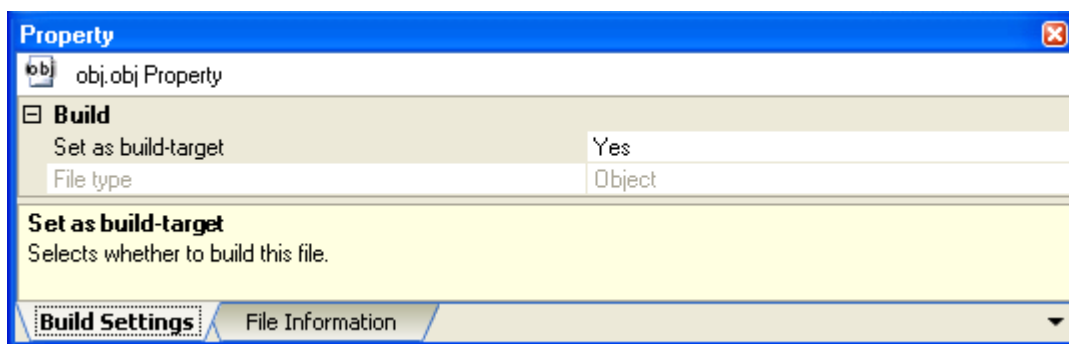
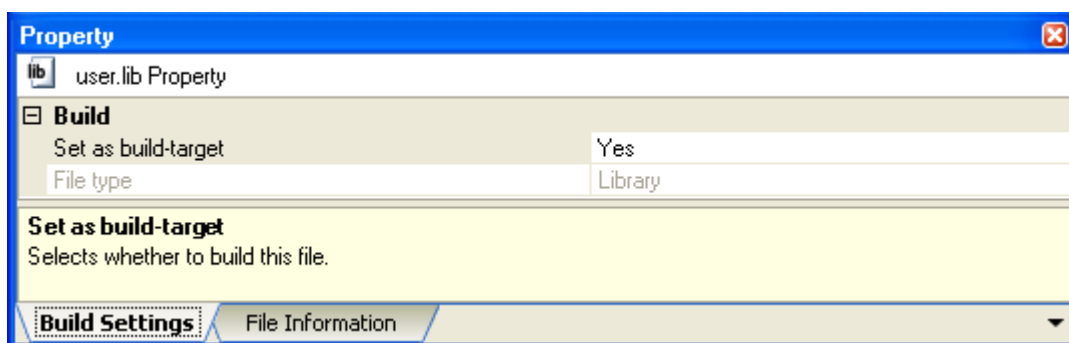


Figure A.15 Property Panel: [Build Settings] Tab (When Selecting Library File)



[Description of each category]

(1) [Build]

The detailed information on the build are displayed and the configuration can be changed.

Set as build-target	Select whether to build the selected file.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Builds the selected file.
Set individual compile option		No	Does not build the selected file.
	Select whether to set a compile option that differs from the project settings to the selected C or C++ source file. This property is displayed only when a C or C++ source file is selected on the Project Tree panel and [Yes] is selected in the [Set as build-target] property in the [Build] category.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Sets a compile option that differs from the project settings to the selected C or C++ source file.
		No	Does not set a compile option that differs from the project settings to the selected C or C++ source file.

Set individual assemble option	Select whether to set an assemble option that differs from the project settings to the selected assembler source file. This property is displayed only when an assembler source file is selected on the Project Tree panel and [Yes] is selected in the [Set as build-target] property in the [Build] category.			
	Default	No		
	How to change	Select from the drop-down list.		
	Restriction	Yes	Sets a compile option that differs from the project settings to the selected assembler source file.	
		No	Does not set a compile option that differs from the project settings to the selected assembler source file.	
File type	Display the type of the selected file.			
	Default	C source (when C source file is selected) C++ source (when C++ source file is selected) Assembly source file (when assembler source file is selected) Object (when object module file is selected) Library (when library file is selected)		
	How to change	Changes not allowed.		

[Individual Compile Options(C)] tab

This tab shows the detailed information on a C source file categorized by the following and the configuration can be changed.

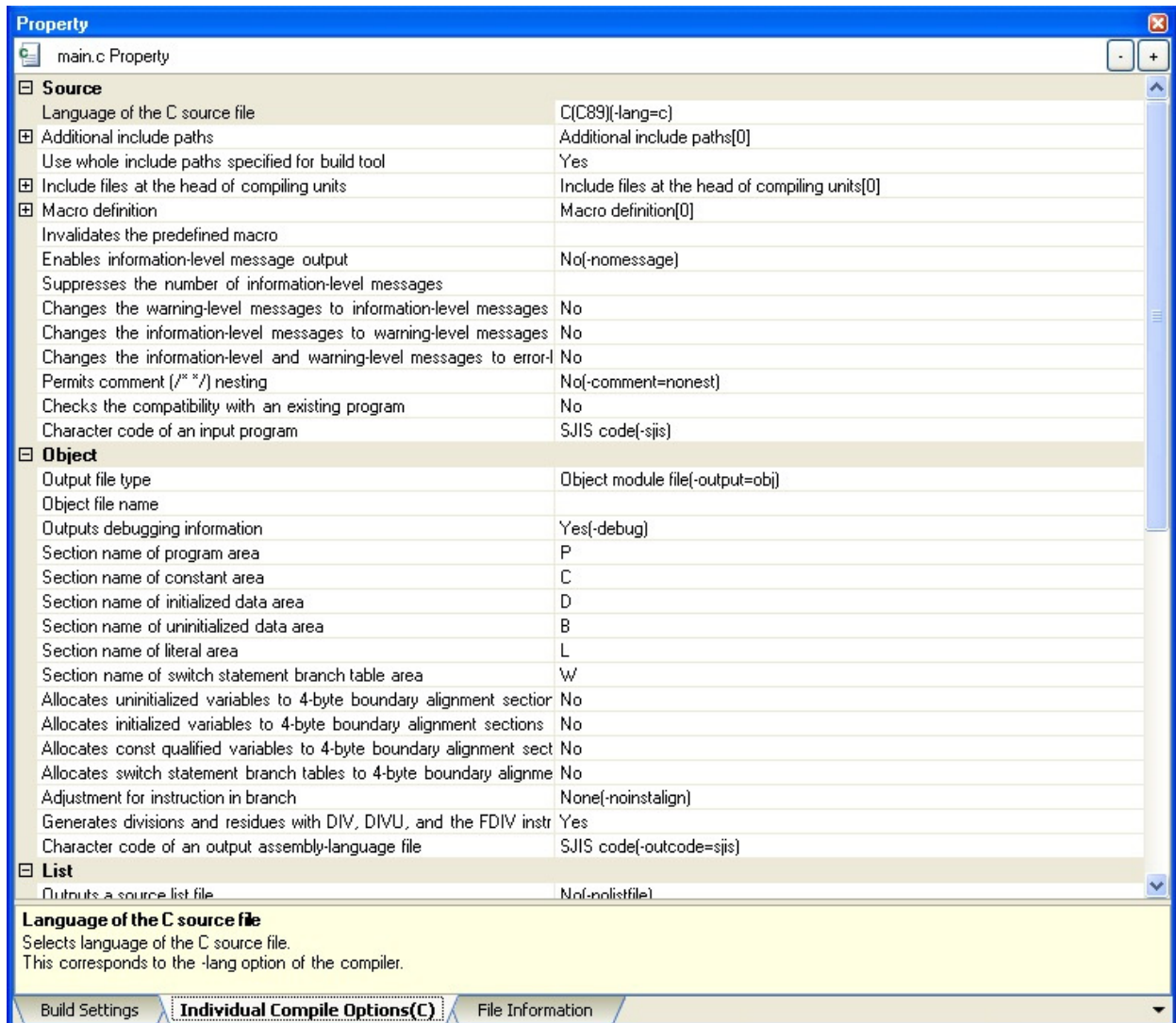
Note that this tab takes over the settings of the [\[Compile Options\] tab](#).

If the settings are changed from the [\[Compile Options\] tab](#), the properties are displayed in boldface.

Remark This tab is displayed only when [Yes] in the [Set individual compile option] property in the [\[Build\]](#) category from the [\[Build Settings\] tab](#) is selected.

- (1) [\[Source\]](#)
- (2) [\[Object\]](#)
- (3) [\[List\]](#)
- (4) [\[Optimization\]](#)
- (5) [\[Output File\]](#)
- (6) [\[MISRA C rule check\]](#)
- (7) [\[Others\]](#)

Figure A.16 Property Panel: [Individual Compile Options(C)] Tab



[Description of each category]

- (1) [\[Source\]](#)
The detailed information on the source is displayed and the configuration can be changed.

Language of the C source file	Selects language of the C source file. This corresponds to the -lang option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	C(C89) (-lang=c)	Compiles as a C (C89) source file.	
		C99 (-lang=c99)	Compiles as a C (C99) source file.	
Additional include paths	Specifies the name of the path to the folder that stores the include file. The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. The reference point of the path is the project folder. This corresponds to the -include option of the compiler. The specified include path is displayed as the subproperty.			
	Default	Additional include paths ^[number of defined items]		
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.		
	Restriction	Up to 247 characters Up to 65536 items can be specified.		
Use whole include paths specified for build tool	Select whether to compile using the include path specified in the [Additional include paths] property in the [Source] category from the [Compile Options] tab of the build tool to be used. The include paths are added by the following procedure. - Paths specified in the [Additional include paths] property from this tab - Paths specified in the [Additional include paths] in the [Source] category from the [Compile Options] tab This corresponds to the -include option of the compiler.			
	Default	Yes		
	How to change	Select from the drop-down list.		
	Restriction	Yes	Compiles using the include path specified in the property of the build tool to be used.	
		No	Does not use the include path specified in the property of the build tool to be used.	

Include files at the head of compiling units	Specifies include files at the head of compiling units. The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. The reference point of the path is the project folder. This corresponds to the -preinclude option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.	
	Restriction	Up to 259 characters Up to 65536 items can be specified.	
Macro definition	Specifies the macro name to be defined. Specify in the format of " <i>macro name=string</i> ", with one macro name per line. The " <i>=string</i> " part can be omitted, and in this case, the macro name is assumed to be defined. This corresponds to the -define option of the compiler. The specified macro is displayed as the subproperty.		
	Default	Configuration of the compile option	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 32767 characters Up to 65536 items can be specified.	
Invalidates the pre-defined macro	Specifies invalidates the predefined macro. If multiple macro names are specified, delimit them with a comma (example: <code>__DBL4, __SCHAR</code>). This corresponds to the -undefine option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Edit by the Specify The Predefined Macro dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Enables information-level message output	Specifies whether information level messages are output. This corresponds to the -message and -nomessage options of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-message)	Enables information message output.
		No(-nomessage)	Disables information message output.

Suppresses the number of information-level messages	Specifies suppresses the number of information-level message. If multiple message numbers are specified, delimit them with a comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -nomessage option of the compiler. This property is displayed only when [No(-nomessage)] in the [Enables information-level message output] property is specified.	
	Default	Configuration of the compile option
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 32767 characters
Changes the warning-level messages to information-level messages	Selects whether to change the warning-level messages to information-level messages. This corresponds to the -change_message option of the compiler.	
	Default	Configuration of the compile option
	How to change	Select from the drop-down list.
	Restriction	Yes(All) (-change_message=information) Changes all warning-level messages to the information-level messages.
		Yes(Specifies error number) (-change_message=information=<Error-Number>) Changes the warning-level messages with the specified error numbers to the information-level messages.
Error number of warning-level message	No	Does not change the warning-level messages to the information-level messages.
	Specifies error number of warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=information=<ErrorNumber>)] in the [Changes the warning-level messages to information-level messages] property is specified.	
	Default	Configuration of the compile option
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 32767 characters

Changes the information-level messages to warning-level messages	Selects whether to change the information-level messages to warning-level messages. This corresponds to the -change_message option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=warning)	Changes all information-level messages to warning-level messages.
		Yes(Specifies error number) (-change_message=warning=<Error-Number>)	Changes the information-level messages with the specified error numbers to warning-level messages.
	No	Does not change the information-level messages to warning-level messages.	
Error number of information-level message	Specifies error number of information-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=warning=<ErrorNumber>)] in the [Changes the information-level messages to warning-level messages] property is specified.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Changes the information-level and warning-level messages to error-level messages	Selects whether to change the information-level and warning-level messages to error-level messages. This corresponds to the -change_message option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=error)	Changes all information-level and warning-level messages to error-level messages.
		Yes(Specifies error number) (-change_message=error=<ErrorNumber>)	Changes the information-level and warning-level messages with the specified error numbers to error-level messages.
	No	Does not change the warning-level messages to information-level messages.	

Error number of information-level and warning-level message	Specifies error number of information-level and warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=error=<ErrorNumber>)] in the [Changes the information-level and warning-level messages to error-level messages] property is specified.			
	Default	Configuration of the compile option		
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.		
	Restriction	Up to 32767 characters		
Permits comment (/* */) nesting	Selects whether to permit comment (/* */) nesting. This corresponds to the -comment option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes (-comment=nest)	Does not permit comment (/* */) nesting.	
No (-comment=nonest)		Permits comment (/* */) nesting.		
Checks the compatibility with an existing program	Selects whether to check the compatibility with an existing program. This corresponds to the -check option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes(NC compiler) (-check=nc)	Checks the compatibility with the R8C and M16C family C compilers.	
		Yes(H8 compiler) (-check=ch38)	Checks the compatibility with the H8, H8S, and H8SX family C/C++ compilers.	
		Yes(SH compiler) (-check=sh)	Checks the compatibility with the SuperH family C/C++ compilers.	
No		Does not check the compatibility with an existing program.		

Character code of an input program	Selects character code of an input program. This corresponds to the -euc , -sjis , -latin1 , -utf8 , -big5 and ?gb2312 option of the compiler. option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	EUC code (-euc)	Handles the characters in strings, character constants, and comments by using EUC.
		SJIS code (-sjis)	Handles the characters in strings, character constants, and comments by using SJIS.
		ISO-Latin1 code (-latin1)	Handles the characters in strings, character constants, and comments by using ISO-Latin1.
		UTF-8 code (-utf8)	Handles the characters in strings, character constants, and comments by using UTF-8. This item is not available when [C(C89) (-lang=c)] in the [Language of the C source file] property is selected
		Traditional Chinese character (-big5)	Handles the characters in strings, character constants, and comments by using Traditional Chinese character.
		Simplified Chinese character (-gb2312)	Handles the characters in strings, character constants, and comments by using Simplified Chinese character.

(2) [Object]

The detailed information on object is displayed and the configuration can be changed.

Output file type	Selects the type of the output file to be generated during a build. This corresponds to the -output option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Object module file (-output=obj)	Outputs a relocatable file.
		Source file after pre-processed (-output=prep)	Outputs a source file after preprocessed.
		Source file after pre-processed(Disables #line output) (-output=prep -noline)	Disables #line output at preprocessor expansion.

Object module file name	Specify the name of the object module file generated after compilation. The extension other than ".obj" cannot be specified. If the extension is omitted, ".obj" is automatically added. If this is blank, the file name will be the source file name with the extension replaced by ".obj". This corresponds to the -output option of the compiler.		
	Default	Blank	
	How to change	Directly enter in the text box.	
	Restriction	Up to 259 characters	
Path of the output folder	Specifies the output destination folder for the output file. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -output option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.	
	Restriction	Up to 247 characters	
Outputs debugging information	Selects whether to output debugging information to object module files. This corresponds to the -debug and -nodebug options of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-debug)	Outputs debugging information to object module files.
		No (-nodebug)	Does not output debugging information to object module files.
Section name of program area	Specifies the section name of program area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of constant area	Specifies the section name of constant area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	

Section name of initialized data area	Specifies the section name of initialized data area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of uninitialized data area	Specifies the section name of uninitialized data area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of literal area	Specifies the section name of literal area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of switch statement branch table area	Specifies the section name of switch statement branch table area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Allocates uninitialized variables to 4-byte boundary alignment sections	Selects whether to allocate uninitialized variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=B)	Allocates uninitialized variables to 4-byte boundary alignment sections.
		No	Does not allocate uninitialized variables to 4-byte boundary alignment sections.
Allocates initialized variables to 4-byte boundary alignment sections	Selects whether to allocate initialized variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=D)	Allocates initialized variables to 4-byte boundary alignment sections.
		No	Does not allocates initialized variables to 4-byte boundary alignment sections.

Allocates const qualified variables to 4-byte boundary alignment sections	Selects whether to allocate const qualified variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=C)	Allocates const qualified variables to 4-byte boundary alignment sections.
		No	Does not allocate const qualified variables to 4-byte boundary alignment sections.
Allocates switch statement branch tables to 4-byte boundary alignment sections	Selects whether to allocate switch statement branch tables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=W)	Allocates switch statement branch tables to 4-byte boundary alignment sections.
		No	Does not allocate switch statement branch tables to 4-byte boundary alignment sections.

Adjustment for instruction in branch	Selects adjustment for instruction in branch. This corresponds to the <code>-noinstalign</code> , <code>-installign4</code> , and <code>-installign8</code> option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	None (-noinstalign)	Does not align instructions at branch destinations.	
		Execution in 4 bytes (-installign4)	Aligns instructions at branch destinations to 4-byte boundaries.	
		Execution in 4 bytes(Contains each loop head) (-installign4=loop)	Aligns instructions at branch destinations to 4-byte boundaries (Contains head of each loop).	
		Execution in 4 bytes(Contains each inmost loop head) (-installign4=inmost-loop)	Aligns instructions at branch destinations to 4-byte boundaries (Contains head of each inmost loop).	
		Execution in 8 bytes (-installign8)	Aligns instructions at branch destinations to 8-byte boundaries.	
Execution in 8 bytes (Contains each loop head) (-installign8=loop)		Aligns instructions at branch destinations to 8-byte boundaries (Contains head of each loop).		
Execution in 8 bytes (Contains each inmost loop head) (-installign8=inmost-loop)	Aligns instructions at branch destinations to 8-byte boundaries (Contains head of each inmost loop).			
Generates divisions and residues with DIV, DIVU, and the FDIV instruction	Selects whether to generate divisions and residues with DIV, DIVU, and the FDIV instruction. This corresponds to the <code>-nouse_div_inst</code> option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes	Generates code in which DIV, DIVU, or FDIV instructions are used.	
		No (-nouse_div_inst)	Generates code in which no DIV, DIVU, or FDIV instructions are used.	

Character code of an output assembly-language file	Selects character code of an output assembly-language file. This corresponds to the -outcode option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	EUC code (-out-code=euc)	Outputs characters in strings and character constants using EUC.
		SJIS code (-out-code=sjis)	Outputs characters in strings and character constants using SJIS.
		UTF-8 code (-out-code=utf8)	Outputs characters in strings and character constants using UTF-8. This item is not available when [C(C89) (-lang=c)] in the [Language of the C source file] property in the [Source] category is selected
		Traditional Chinese character (-out-code=big5)	Outputs characters in strings and character constants using Traditional Chinese character.
		Simplified Chinese character (-out-code=gb2312)	Outputs characters in strings and character constants using Simplified Chinese character.

(3) [\[List\]](#)

The detailed information on list file is displayed and the configuration can be changed.

Outputs a source list file	Selects whether to output a source list file. This corresponds to the -listfile and -nolistfile option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-lisfile)	Outputs a source list file.
		No (-nolistfile)	Disable output of a source list file.
Outputs the C/C++ source file	Specifies the contents of the source list file. Selects whether to output the C/C++ source file. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=source)	Outputs the C/C++ source file.
		No	Does not output the C/C++ source file.

Outputs the state-ments unsatisfied in conditional assembly	Specifies the contents of the source list file. Selects whether to output the statements unsatisfied in conditional assembly. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=condi-tionals)	Outputs the statements unsatisfied in condi-tional assembly.
	No	Does not output the statements unsatisfied in conditional assembly.	
Outputs the informa-tion before .DEFINE replacement	Specifies the contents of the source list file. Selects whether to output the information before .DEFINE replacement. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=defini-tions)	Outputs the information before .DEFINE replacement.
	No	Does not output the information before .DEFINE replacement.	
Outputs the assem-bler macro expansion statements	Specifies the contents of the source list file. Selects whether to output the assembler macro expansion statements. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=expan-sions)	Outputs the assembler macro expansion statements.
	No	Does not output the assembler macro expan-sion statements.	

(4) [Optimization]

The detailed information on the optimization is displayed and the configuration can be changed.

Optimization level	Selects optimization level. This corresponds to the -optimize option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	0 (-optimize=0)	Does not optimize the program.
		1 (-optimize=1)	Partially optimizes the program by automatically allocating variables to registers, integrating the function exit blocks, integrating multiple instructions which can be integrated, etc.
		2 (-optimize=2)	Performs overall optimization.
		Max (-optimize=max)	Performs optimization as much as possible.
Outputs additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. At linkage, inter-module optimization is applied to files for which this option has been specified. This corresponds to the -goptimize option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.
		No	Does not outputs additional information for inter-module optimization.
Optimization type	Selects optimization type. This corresponds to the -speed and -size option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Optimizes with emphasis on execution performance (-speed)	Optimizes with emphasis on execution performance.
		Optimizes with emphasis on code size (-size)	Optimizes with emphasis on code size.
Loop expansion	Selects whether to optimize the loop expansion (for, while, and do-while). This corresponds to the -loop option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level and optimization type options	Depends on the optimization level and optimization type options.
		Expansion (-loop=<numeric value>)	Expands loop statements (for, while, and do-while).

Expansion maximum number	Specifies expansion maximum number. This corresponds to the suboption of -loop option of the compiler. This property is displayed only when [Expansion (-loop=<numeric value>)] in the [Loop expansion] property is selected.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	1 to 32 (decimal number)	
Performs inline expansion automatically	Selects whether to perform inline expansion automatically. This option corresponds to the -inline and -noinline option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level and optimization type options	Depends on the optimization level and optimization type options.
		Yes (-inline=<numeric value>)	Performs inline expansion automatically.
		No (-noinline)	Does not perform inline expansion automatically.
Maximum increasing rate of function size	Specifies maximum increasing rate of function size. For example, when 100 is specified, inline expansion will be performed until the function size has increased by 100% (size is doubled). This option corresponds to the -inline option of the compiler. This property is displayed only when [Yes (-inline=<numeric value>)] in the [Performs inline expansion automatically] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	1 to 65535 (decimal number)	
Expansion method of the switch statement	Selects expansion method of the switch statement. This corresponds to the -case option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	if_then method (-case=ifthen)	Expands the switch statement using the if_then method.
		Jumping to a table method (-case=table)	Expands the switch statement by using the table method.
		Compiler automatically selects (-case=auto)	Automatically selects the if_then method or table method.

Handles external variables as if they are volatile qualified	Selects whether to handle all external variables as if they are volatile qualified. This corresponds to the <code>-volatile</code> and <code>-novolatile</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-volatile)	Handles all external variables as if they were volatile qualified.
		No (-novolatile)	Does not handle external variables as if they were volatile qualified.
Performs the constant propagation of const qualified external variables	Selects whether to perform the constant propagation of const qualified external variables. Const qualified variables in a C++ source file cannot be controlled by this option (constant propagation is always performed). This corresponds to the <code>-const_copy</code> and <code>-noconst_copy</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level options	Depends on the optimization level options.
		Yes (-const_copy)	Enables constant propagation of const qualified external variables.
		No (-noconst_copy)	Disables constant propagation of const qualified external variables.
Conversion method of the divisions and residues of integer constants	Selects conversion method of the divisions and residues of integer constants. This corresponds to the <code>-const_div</code> and <code>-noconst_div</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization type option	Depends on the optimization type option
		Instruction sequence using multiplication (-const_div)	Performs constant division (residue) by an instruction sequence using multiplication.
		Instruction sequence using division (-noconst_div)	Performs constant division (residue) by an instruction sequence using division.
Expansion method of the library function	Selects expansion method of the library function. This corresponds to the <code>-library</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Calls library functions (-library=function)	Calls all library functions.
		Performs instruction expansion of several library functions (-library=intrinsic)	Performs instruction expansion for <code>abs()</code> , <code>fabsf()</code> , and library functions which can use string manipulation instructions.

Divides the optimizing ranges into many sections before compilation	Selects whether to divide the optimizing ranges of the large-size function into many sections before compilation. This corresponds to the -scope and -noscope option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level option	Depends on the optimization level option.
		Yes (-scope)	Divides the optimizing ranges of the large-size function into many sections before compilation.
		No (-noscope)	Does not divide the optimizing ranges before compilation.
Schedules the instruction taking into consideration pipeline processing	Selects whether to schedule the instruction taking into consideration pipeline processing. This corresponds to the -schedule and -noschedule option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level option	Depends on the optimization level option.
		Yes (-schedule)	Schedules instructions taking into consideration pipeline processing.
		No (-noschedule)	Does not schedule instructions.
Optimizes accesses to external variables	Selects whether to optimize accesses to external variables. This corresponds to the -nomap , -smap and -map option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(Optimizes the inner-module) (-smap)	Optimizes accesses to external variables which are defined in the file to be compiled.
		Yes(Optimizes the inter-module) (-map)	Optimizes accesses to external variables.
		No (-nomap)	Disables optimization for accesses to external variables.
Perform inter-module optimization	Specify the level of inter-module optimization (such as function merging). This corresponds to the -ip_optimize option of the ccrh command.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(Level 1)(Perform)(-ip_optimize)	Performs inter-module optimization for each file.
		No	Does not perform inter-module optimization.

Converts floating-point constant division into multiplication	Selects whether to convert floating-point constant division into multiplication of the corresponding reciprocals as constants. This corresponds to the -approxdiv option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-approxdiv)	Converts floating-point constant division into multiplication.
	No	Does not convert floating-point constant division into multiplication.	
Omits a check of the range for conversion between the floating type and unsigned integer type	Selects whether to omit a check of the range for conversion between the floating type and unsigned integer type. When "Yes" is specified, code performance of the relevant type conversion processing is improved. The conversion result may, however, differ from C/C++ language specifications, so take care on this point. This corresponds to the -simple_float_conv option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-simple_float_conv)	Omits part of the type conversion processing for the floating type.
	No	Does not omit part of the type conversion processing for the floating type.	
Performs optimization considering the type of the data indicated by the pointer	Selects whether to perform optimization considering the type of the data indicated by the pointer. Although the performance of object code is generally better than when -alias=noansi is specified, the results of execution may differ according to whether -alias=ansi or alias=noansi is specified. This corresponds to the -alias option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-alias=ansi)	Performs optimization considering the type of the data indicated by the pointer.
	No (-alias=noansi)	Does not perform optimization considering the type of the data indicated by the pointer.	

Optimizes modification of the operation order of a floating-point expression	<p>Selects whether to optimize modification of the operation order of a floating-point expression.</p> <p>Specifying the <code>-float_order</code> option generally improves the object performance compared to when not specifying it.</p> <p>However, the accuracy of operations may differ from that when <code>-float_order</code> is not specified.</p> <p>This corresponds to the <code>-float_order</code> option of the compiler.</p> <p>This property is valid only when [2 (-optimize=2)] or [Max (-optimize=max)] in the [Optimization level] property is specified.</p>			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes (-float_order)	Optimizes modification of the operation order in a floating-point expression.	
		No	Does not optimize modification of the operation order in a floating-point expression.	

(5) [Output File]

The detailed information on the output file check is displayed and the configuration can be changed.

Output assembly source file

Select whether to output the assembly source file of the compile result for the C source.

This corresponds to the `-output=src` option of the compiler.

Output assembly source file	Select whether to output the assembly source file of the compile result for the C source. This corresponds to the ?output=src option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-output=src)	Outputs the assembly source file of the compile result for the C source.
No		Does not output the assembly source file of the compile result for the C source.	
Output preprocessed source file	Select whether to output the execution result of preprocessing for the source file to a file. This corresponds to the ?output=prep, -noline option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-output=prep)	Outputs the execution result of preprocessing for the source file to a file.
Yes(Suppress #line)(-output=prep -noline)		Outputs the execution result of preprocessing (suppress #line) for the source file to a file.	
No		Does not output the execution result of preprocessing for the source file to a file.	

(6) [MISRA C rule check]

The detailed information on the MISRA-C: 2004 rules check is displayed and the configuration can be changed.

Apply rule	Selects to apply MISRA C rule. This option corresponds to the -misra2004 option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Apply all rules (-misra2004=all)	Checks the source code against all of the rules that are supported.
		Apply specified rule number (-misra2004=apply)	Checks the source code against the rules with the selected numbers.
		Ignore specified rule number(-misra2004=ignore)	Checks the source code against the rules other than those with the selected numbers.
		Apply rules that are classified as "required" (-misra2004=required)	Checks the source code against the rules of the "required" type.
		Apply rules that are classified as "required" and specified rule number (-misra2004=required_add)	Checks the source code against the rules of the "required" type and the rules with the selected numbers.
		Ignore specified rule number from rules that are classified as "required" (-misra2004=required_remove)	Checks the source code against the rules other than those with the selected numbers among the rules of the "required" type.
		Apply rules that are described in the specified file (-misra2004=<file name>)	Checks the source code against the rules with the numbers written in the specified file.
	Not apply rule	Does not apply MISRA C rule.	
Rule number description file	Specifies Rule number description file(misra2004 rule file). The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Apply rules that are described in the specified file (-misra2004=<file name>)] in the [Apply rule] property is selected.		
	Default	Configuration of the compile option	
	How to change	Directly enter to the text box or edit by the Specify Misra2004 Rule File dialog box which appears when clicking the [...] button.	
	Restriction	Up to 259 characters	

Rule number	Specifies the rule number. One or more rule numbers always in decimal must be specified. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Apply specified rule number (-misra2004=apply)] in the [Apply rule] property is selected.	
	Default	Configuration of the compile option
	How to change	Directly enter to the text box or edit by the Specify the rule number dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Exclusion rule number	Specifies the exclusion rule number. One or more rule numbers always in decimal must be specified. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Ignore specified rule number(-misra2004=ignore)] in the [Apply rule] property is selected.	
	Default	Configuration of the compile option
	How to change	Directly enter to the text box or edit by the Specify the rule number dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Check rule number besides required rule	Specifies the check rule number besides required rule. One or more rule numbers always in decimal must be specified. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Apply rules that are classified as "required" and specified rule number (-misra2004=required_add)] in the [Apply rule] property is selected.	
	Default	Configuration of the compile option
	How to change	Directly enter to the text box or edit by the Specify the rule number dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Exclusion rule number from required rule	Specifies the exclusion rule number from required rule. One or more rule numbers always in decimal must be specified. This option corresponds to the -misra2004 option of the compiler. This property is displayed only when [Ignore specified rule number from rules that are classified as "required" (-misra2004=required_remove)] in the [Apply rule] property is selected.	
	Default	Configuration of the compile option
	How to change	Directly enter to the text box or edit by the Specify the rule number dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters

Rule check exclusion file	Specifies rule check exclusion file. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. This option corresponds to the -ignore_files_misra option of the compiler. This option is not display when [Not apply rule] in the [Apply rule] property has been specified.		
	Default	Configuration of the compile option	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.	
	Restriction	Up to 247 characters Up to 65536 items can be specified.	
Outputs message of The the enhanced key word and extended specifications	Selects whether to output message of The the enhanced key word and extended specifications. This option corresponds to the -check_language_extension option. This option is not display when [Not apply rule] in the [Apply rule] property has been specified.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-check_language_ext ension)	Enables complete checking against the MISRA-C: 2004 rules for parts of the code where it would otherwise be suppressed due to individual extensions from the C/C++ language specification.
		No	Disables complete checking against the MISRA-C: 2004 rules for parts of the code where it would otherwise be suppressed due to individual extensions from the C/C++ language specification.

(7) [Others]

Other detailed information on compilation is displayed and the configuration can be changed.

Outputs the copyright	<p>Selects whether to output the copyright.</p> <p>This corresponds to the -nologo option of the compiler.</p>		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-logo)	Outputs the copyright.
		No (-nologo)	Disables output of the copyright.

Outputs the cross reference information	Selects whether to output cross reference information. It is necessary to change the setting of the property of "Program Analyzer" to change this option.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes(-Xcref)	Outputs the cross reference information.	
		No	Does not output of the cross reference information.	
Commands executed before compile processing	Specifies the command to be executed before compile processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %CompiledFile%: Replaces with the absolute path of the output file under compiling. %InputFile%: Replaces with the absolute path of the file to be compiled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before compile processing. The placeholders can be described in the script. The specified command is displayed as the subproperty.			
	Default	Configuration of the compile option		
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.		
	Restriction	Up to 1023 characters Up to 64 items can be specified.		

Commands executed after compile processing	<p>Specifies the command to be executed after compile processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %CompiledFile%: Replaces with the absolute path of the output file under compiling. %InputFile%: Replaces with the absolute path of the file to be compiled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after compile processing.</p> <p>The placeholders can be described in the script.</p> <p>The specified command is displayed as the subproperty.</p>	
	Default	Configuration of the compile option
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.
Other additional options	Inputs the compile options to be added additionally. The options set here are added at the end of the compile options group.	
	Default	Configuration of the compile option
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	The specified option is displayed.	
	Default	Configuration of the compile option
	How to change	Changes not allowed

[Individual Compile Options(C++)] tab

This tab shows the detailed information on a C++ source file categorized by the following and the configuration can be changed.

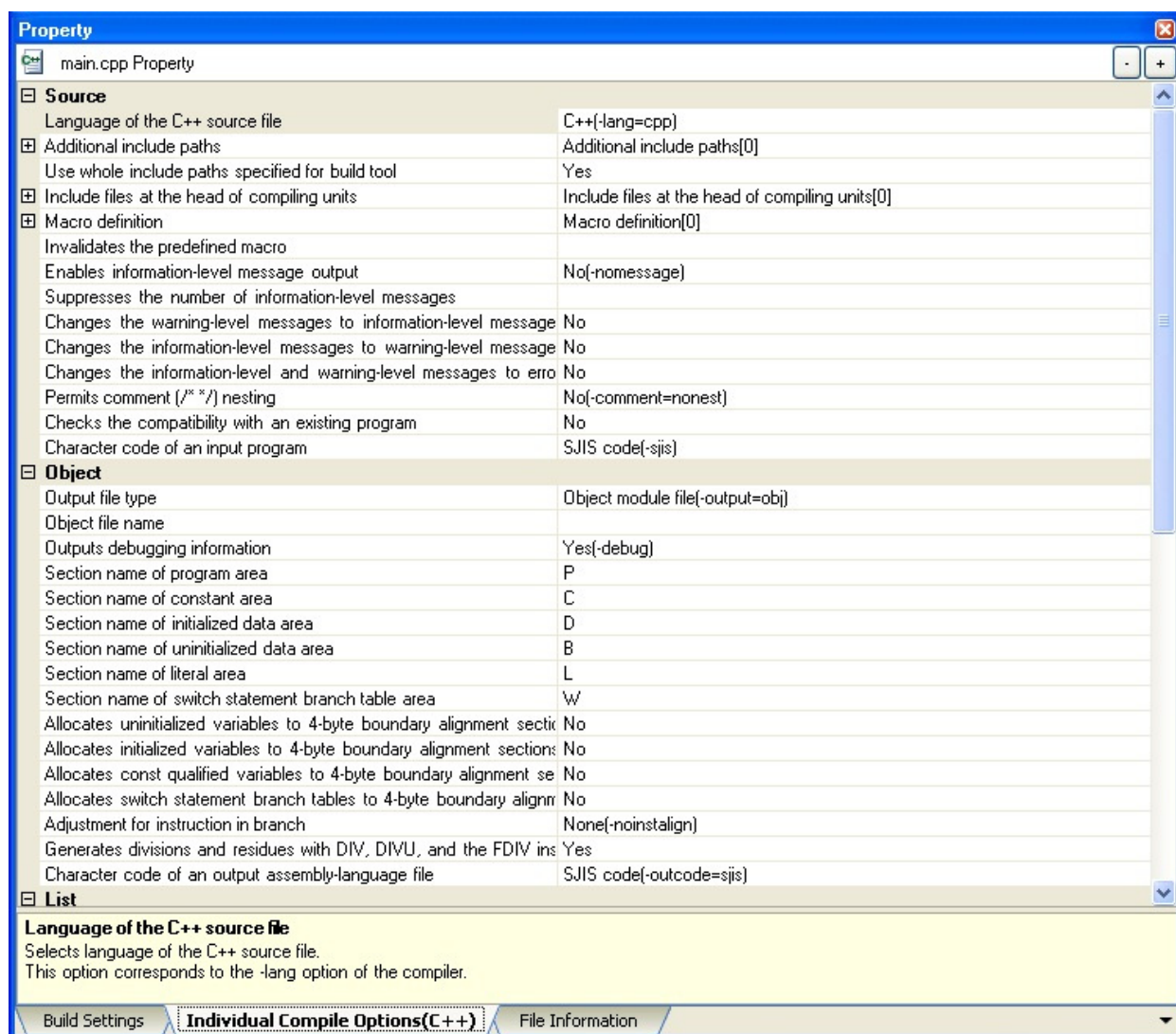
Note that this tab takes over the settings of the [\[Compile Options\] tab](#).

If the settings are changed from the [\[Compile Options\] tab](#), the properties are displayed in boldface.

Remark This tab is displayed only when [Yes] in the [Set individual compile option] property in the [\[Build\]](#) category from the [\[Build Settings\] tab](#) is selected.

- (1) [\[Source\]](#)
- (2) [\[Object\]](#)
- (3) [\[List\]](#)
- (4) [\[Optimization\]](#)
- (5) [\[Output File\]](#)
- (6) [\[Others\]](#)

Figure A.17 Property Panel: [Individual Compile Options(C++)] Tab



[Description of each category]

- (1) [\[Source\]](#)
The detailed information on the source is displayed and the configuration can be changed.

Language of the C++ source file	Selects language of the C++ source file. This option corresponds to the -lang option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	C++ (-lang=cpp)	Compiles as an EC++ source file.
EC++ (-lang=ecpp)		Compiles as a C++ source file.	
Additional include paths	Specifies the name of the path to the folder that stores the include file. The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. The reference point of the path is the project folder. This corresponds to the -include option of the compiler. The specified include path is displayed as the subproperty.		
	Default	Additional include paths[<i>number of defined items</i>]	
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 247 characters Up to 65536 items can be specified.	
Use whole include paths specified for build tool	Select whether to compile using the include path specified in the [Additional include paths] property in the [Source] category from the [Compile Options] tab of the build tool to be used. The include paths are added by the following procedure. - Paths specified in the [Additional include paths] property from this tab - Paths specified in the [Additional include paths] in the [Source] category from the [Compile Options] tab This corresponds to the -include option of the compiler.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Compiles using the include path specified in the property of the build tool to be used.
No		Does not use the include path specified in the property of the build tool to be used.	

Include files at the head of compiling units	Specifies include files at the head of compiling units. The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. The reference point of the path is the project folder. This corresponds to the -preinclude option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.	
	Restriction	Up to 259 characters Up to 65536 items can be specified.	
Macro definition	Specifies the macro name to be defined. Specify in the format of " <i>macro name=string</i> ", with one macro name per line. The " <i>=string</i> " part can be omitted, and in this case, the macro name is assumed to be defined. This corresponds to the -define option of the compiler. The specified macro is displayed as the subproperty.		
	Default	Configuration of the compile option	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 32767 characters Up to 65536 items can be specified.	
Invalidates the pre-defined macro	Specifies invalidates the predefined macro. If multiple macro names are specified, delimit them with a comma (example: <code>__DBL4, __SCHAR</code>). This corresponds to the -undefine option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Edit by the Specify The Predefined Macro dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Enables information-level message output	Specifies whether information level messages are output. This corresponds to the -message and -nomessage options of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-message)	Enables information message output.
		No(-nomessage)	Disables information message output.

Suppresses the number of information-level messages	Specifies suppresses the number of information-level message. If multiple message numbers are specified, delimit them with a comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -nomessage option of the compiler. This property is displayed only when [No(-nomessage)] in the [Enables information-level message output] property is specified.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Changes the warning-level messages to information-level messages	Selects whether to change the warning-level messages to information-level messages. This corresponds to the -change_message option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=information)	Changes all warning-level messages to the information-level messages.
		Yes(Specifies error number) (-change_message=information=<Error-Number>)	Changes the warning-level messages with the specified error numbers to the information-level messages.
		No	Does not change the warning-level messages to the information-level messages.
Error number of warning-level message	Specifies error number of warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=information=<ErrorNumber>)] in the [Changes the warning-level messages to information-level messages] property is specified.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	

Changes the information-level messages to warning-level messages	Selects whether to change the information-level messages to warning-level messages. This corresponds to the -change_message option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=warning)	Changes all information-level messages to warning-level messages.
		Yes(Specifies error number) (-change_message=warning=<Error-Number>)	Changes the information-level messages with the specified error numbers to warning-level messages.
	No	Does not change the information-level messages to warning-level messages.	
Error number of information-level message	Specifies error number of information-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=warning=<ErrorNumber>)] in the [Changes the information-level messages to warning-level messages] property is specified.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Changes the information-level and warning-level messages to error-level messages	Selects whether to change the information-level and warning-level messages to error-level messages. This corresponds to the -change_message option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(All) (-change_message=error)	Changes all information-level and warning-level messages to error-level messages.
		Yes(Specifies error number) (-change_message=error=<ErrorNumber>)	Changes the information-level and warning-level messages with the specified error numbers to error-level messages.
	No	Does not change the warning-level messages to information-level messages.	

Error number of information-level and warning-level message	Specifies error number of information-level and warning-level message. If multiple message numbers are specified, delimit them with comma (example: 4,200). Also, the range can be set using hyphen (example: 4,200-203,1300). This corresponds to the -change_message option of the compiler. This property is displayed only when [Yes(Specifies error number) (-change_message=error=<ErrorNumber>)] in the [Changes the information-level and warning-level messages to error-level messages] property is specified.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.	
	Restriction	Up to 32767 characters	
Permits comment (/* */) nesting	Selects whether to permit comment (/* */) nesting. This corresponds to the -comment option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-comment=nest)	Does not permit comment (/* */) nesting.
		No (-comment=nonest)	Permits comment (/* */) nesting.
Checks the compatibility with an existing program	Selects whether to check the compatibility with an existing program. This corresponds to the -check option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(NC compiler) (-check=nc)	Checks the compatibility with the R8C and M16C family C compilers.
		Yes(H8 compiler) (-check=ch38)	Checks the compatibility with the H8, H8S, and H8SX family C/C++ compilers.
		Yes(SH compiler) (-check=sh)	Checks the compatibility with the SuperH family C/C++ compilers.
		No	Does not check the compatibility with an existing program.

Character code of an input program	Selects character code of an input program. This corresponds to the -euc , -sjis , -latin1 , -utf8 , -big5 and -gb2312 option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	EUC code (-euc)	Handles the characters in strings, character constants, and comments by using EUC.
		SJIS code (-sjis)	Handles the characters in strings, character constants, and comments by using SJIS.
		ISO-Latin1 code (-latin1)	Handles the characters in strings, character constants, and comments by using ISO-Latin1.
		UTF-8 code (-utf8)	Handles the characters in strings, character constants, and comments by using UTF-8. This item is not available when [C(C89) (-lang=c)] in the [Language of the C source file] property is selected
		Traditional Chinese character (-big5)	Handles the characters in strings, character constants, and comments by using Traditional Chinese character.
		Simplified Chinese character (-gb2312)	Handles the characters in strings, character constants, and comments by using Simplified Chinese character.

(2) [Object]

The detailed information on object is displayed and the configuration can be changed.

Output file type	Selects the type of the output file to be generated during a build. This corresponds to the -output option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Object module file (-output=obj)	Outputs a relocatable file.
		Source file after pre-processed (-output=prep)	Outputs a source file after preprocessed.
		Source file after pre-processed(Disables #line output) (-output=prep -noline)	Disables #line output at preprocessor expansion.

Object module file name	Specify the name of the object module file generated after compilation. The extension other than ".obj" cannot be specified. If the extension is omitted, ".obj" is automatically added. If this is blank, the file name will be the source file name with the extension replaced by ".obj". This corresponds to the -output option of the compiler.		
	Default	Blank	
	How to change	Directly enter in the text box.	
	Restriction	Up to 259 characters	
Path of the output folder	Specifies the output destination folder for the output file. The following placeholders are supported. %BuildModeName%: Replaces with the build mode name. %ProjectName%: Replaces with the project name. %MicomToolPath%: Replaces with the absolute path of the product install folder. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -output option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.	
	Restriction	Up to 247 characters	
Outputs debugging information	Selects whether to output debugging information to object module files. This corresponds to the -debug and -nodebug options of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-debug)	Outputs debugging information to object module files.
		No (-nodebug)	Does not output debugging information to object module files.
Section name of program area	Specifies the section name of program area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of constant area	Specifies the section name of constant area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	

Section name of initialized data area	Specifies the section name of initialized data area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of uninitialized data area	Specifies the section name of uninitialized data area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of literal area	Specifies the section name of literal area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Section name of switch statement branch table area	Specifies the section name of switch statement branch table area. This corresponds to the -section option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	Up to 32767 characters	
Allocates uninitialized variables to 4-byte boundary alignment sections	Selects whether to allocate uninitialized variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=B)	Allocates uninitialized variables to 4-byte boundary alignment sections.
		No	Does not allocate uninitialized variables to 4-byte boundary alignment sections.
Allocates initialized variables to 4-byte boundary alignment sections	Selects whether to allocate initialized variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=D)	Allocates initialized variables to 4-byte boundary alignment sections.
		No	Does not allocates initialized variables to 4-byte boundary alignment sections.

Allocates const qualified variables to 4-byte boundary alignment sections	Selects whether to allocate const qualified variables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=C)	Allocates const qualified variables to 4-byte boundary alignment sections.
		No	Does not allocate const qualified variables to 4-byte boundary alignment sections.
Allocates switch statement branch tables to 4-byte boundary alignment sections	Selects whether to allocate switch statement branch tables to 4-byte boundary alignment sections. This corresponds to the -nostuff option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-nostuff=W)	Allocates switch statement branch tables to 4-byte boundary alignment sections.
		No	Does not allocate switch statement branch tables to 4-byte boundary alignment sections.

Adjustment for instruction in branch	Selects adjustment for instruction in branch. This corresponds to the <code>-noinstalign</code> , <code>-instalign4</code> , and <code>-instalign8</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	None (<code>-noinstalign</code>)	Does not align instructions at branch destinations.
		Execution in 4 bytes (<code>-instalign4</code>)	Aligns instructions at branch destinations to 4-byte boundaries.
		Execution in 4 bytes(Contains each loop head) (<code>-instalign4=loop</code>)	Aligns instructions at branch destinations to 4-byte boundaries (Contains head of each loop).
		Execution in 4 bytes(Contains each inmost loop head) (<code>-instalign4=inmost-loop</code>)	Aligns instructions at branch destinations to 4-byte boundaries (Contains head of each inmost loop).
		Execution in 8 bytes (<code>-instalign8</code>)	Aligns instructions at branch destinations to 8-byte boundaries.
		Execution in 8 bytes (Contains each loop head) (<code>-instalign8=loop</code>)	Aligns instructions at branch destinations to 8-byte boundaries (Contains head of each loop).
Execution in 8 bytes (Contains each inmost loop head) (<code>-instalign8=inmost-loop</code>)	Aligns instructions at branch destinations to 8-byte boundaries (Contains head of each inmost loop).		
Generates divisions and residues with DIV, DIVU, and the FDIV instruction	Selects whether to generate divisions and residues with DIV, DIVU, and the FDIV instruction. This corresponds to the <code>-nouse_div_inst</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Generates code in which DIV, DIVU, or FDIV instructions are used.
		No (<code>-nouse_div_inst</code>)	Generates code in which no DIV, DIVU, or FDIV instructions are used.

Character code of an output assembly-language file	Selects character code of an output assembly-language file. This corresponds to the -outcode option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	EUC code (-out-code=euc)	Outputs characters in strings and character constants using EUC.
		SJIS code (-out-code=sjis)	Outputs characters in strings and character constants using SJIS.
		UTF-8 code (-out-code=utf8)	Outputs characters in strings and character constants using UTF-8. This item is not available when [C(C89) (-lang=c)] in the [Language of the C source file] property in the [Source] category is selected
		Traditional Chinese character (-out-code=big5)	Outputs characters in strings and character constants using Traditional Chinese character.
		Simplified Chinese character (-out-code=gb2312)	Outputs characters in strings and character constants using Simplified Chinese character.

(3) [\[List\]](#)

The detailed information on list file is displayed and the configuration can be changed.

Outputs a source list file	Selects whether to output a source list file. This corresponds to the -listfile and -nolistfile option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-lisfile)	Outputs a source list file.
		No (-nolistfile)	Disable output of a source list file.
Outputs the C/C++ source file	Specifies the contents of the source list file. Selects whether to output the C/C++ source file. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=source)	Outputs the C/C++ source file.
		No	Does not output the C/C++ source file.

Outputs the statements unsatisfied in conditional assembly	Specifies the contents of the source list file. Selects whether to output the statements unsatisfied in conditional assembly. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=conditionals)	Outputs the statements unsatisfied in conditional assembly.
	No	Does not output the statements unsatisfied in conditional assembly.	
Outputs the information before .DEFINE replacement	Specifies the contents of the source list file. Selects whether to output the information before .DEFINE replacement. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=definitions)	Outputs the information before .DEFINE replacement.
	No	Does not output the information before .DEFINE replacement.	
Outputs the assembler macro expansion statements	Specifies the contents of the source list file. Selects whether to output the assembler macro expansion statements. This corresponds to the -show option of the compiler. This property is not displayed when [Yes (-lisfile)] in the [Outputs a source list file] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=expansions)	Outputs the assembler macro expansion statements.
	No	Does not output the assembler macro expansion statements.	

(4) [Optimization]

The detailed information on the optimization is displayed and the configuration can be changed.

Optimization level	Selects optimization level. This corresponds to the -optimize option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	0 (-optimize=0)	Does not optimize the program.
		1 (-optimize=1)	Partially optimizes the program by automatically allocating variables to registers, integrating the function exit blocks, integrating multiple instructions which can be integrated, etc.
		2 (-optimize=2)	Performs overall optimization.
		Max (-optimize=max)	Performs optimization as much as possible.
Outputs additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. At linkage, inter-module optimization is applied to files for which this option has been specified. This corresponds to the -goptimize option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.
		No	Does not outputs additional information for inter-module optimization.
Optimization type	Selects optimization type. This corresponds to the -speed and -size option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Optimizes with emphasis on execution performance (-speed)	Optimizes with emphasis on execution performance.
		Optimizes with emphasis on code size (-size)	Optimizes with emphasis on code size.
Loop expansion	Selects whether to optimize the loop expansion (for, while, and do-while). This corresponds to the -loop option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level and optimization type options	Depends on the optimization level and optimization type options.
		Expansion (-loop=<numeric value>)	Expands loop statements (for, while, and do-while).

Expansion maximum number	Specifies expansion maximum number. This corresponds to the suboption of -loop option of the compiler. This property is displayed only when [Expansion (-loop=<numeric value>)] in the [Loop expansion] property is selected.		
	Default	Configuration of the compile option	
	How to change	Directly enter in the text box.	
	Restriction	1 to 32 (decimal number)	
Performs inline expansion automatically	Selects whether to perform inline expansion automatically. This option corresponds to the -inline and -noinline option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level and optimization type options	Depends on the optimization level and optimization type options.
		Yes (-inline=<numeric value>)	Performs inline expansion automatically.
		No (-noinline)	Does not perform inline expansion automatically.
Maximum increasing rate of function size	Specifies maximum increasing rate of function size. For example, when 100 is specified, inline expansion will be performed until the function size has increased by 100% (size is doubled). This option corresponds to the -inline option of the compiler. This property is displayed only when [Yes (-inline=<numeric value>)] in the [Performs inline expansion automatically] property is selected.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	1 to 65535 (decimal number)	
Expansion method of the switch statement	Selects expansion method of the switch statement. This corresponds to the -case option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	if_then method (-case=ifthen)	Expands the switch statement using the if_then method.
		Jumping to a table method (-case=table)	Expands the switch statement by using the table method.
		Compiler automatically selects (-case=auto)	Automatically selects the if_then method or table method.

Handles external variables as if they are volatile qualified	Selects whether to handle all external variables as if they are volatile qualified. This corresponds to the <code>-volatile</code> and <code>-novolatile</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-volatile)	Handles all external variables as if they were volatile qualified.
		No (-novolatile)	Does not handle external variables as if they were volatile qualified.
Performs the constant propagation of const qualified external variables	Selects whether to perform the constant propagation of const qualified external variables. Const qualified variables in a C++ source file cannot be controlled by this option (constant propagation is always performed). This corresponds to the <code>-const_copy</code> and <code>-noconst_copy</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level options	Depends on the optimization level options.
		Yes (-const_copy)	Enables constant propagation of const qualified external variables.
		No (-noconst_copy)	Disables constant propagation of const qualified external variables.
Conversion method of the divisions and residues of integer constants	Selects conversion method of the divisions and residues of integer constants. This corresponds to the <code>-const_div</code> and <code>-noconst_div</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization type option	Depends on the optimization type option
		Instruction sequence using multiplication (-const_div)	Performs constant division (residue) by an instruction sequence using multiplication.
		Instruction sequence using division (-noconst_div)	Performs constant division (residue) by an instruction sequence using division.
Expansion method of the library function	Selects expansion method of the library function. This corresponds to the <code>-library</code> option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Calls library functions (-library=function)	Calls all library functions.
		Performs instruction expansion of several library functions (-library=intrinsic)	Performs instruction expansion for <code>abs()</code> , <code>fabsf()</code> , and library functions which can use string manipulation instructions.

Divides the optimizing ranges into many sections before compilation	Selects whether to divide the optimizing ranges of the large-size function into many sections before compilation. This corresponds to the -scope and -noscope option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level option	Depends on the optimization level option.
		Yes (-scope)	Divides the optimizing ranges of the large-size function into many sections before compilation.
		No (-noscope)	Does not divide the optimizing ranges before compilation.
Schedules the instruction taking into consideration pipeline processing	Selects whether to schedule the instruction taking into consideration pipeline processing. This corresponds to the -schedule and -noschedule option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Depends on the optimization level option	Depends on the optimization level option.
		Yes (-schedule)	Schedules instructions taking into consideration pipeline processing.
		No (-noschedule)	Does not schedule instructions.
Optimizes accesses to external variables	Selects whether to optimize accesses to external variables. This corresponds to the -nomap , -smap and -map option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(Optimizes the inner-module) (-smap)	Optimizes accesses to external variables which are defined in the file to be compiled.
		Yes(Optimizes the inter-module) (-map)	Optimizes accesses to external variables.
		No (-nomap)	Disables optimization for accesses to external variables.
Perform inter-module optimization	Specify the level of inter-module optimization (such as function merging). This corresponds to the -ip_optimize option of the ccrh command.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(Level 1)(Perform)(-ip_optimize)	Performs inter-module optimization for each file.
		No	Does not perform inter-module optimization.

Converts floating-point constant division into multiplication	Selects whether to convert floating-point constant division into multiplication of the corresponding reciprocals as constants. This corresponds to the -approxdiv option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-approxdiv)	Converts floating-point constant division into multiplication.
	No	Does not convert floating-point constant division into multiplication.	
Omits a check of the range for conversion between the floating type and unsigned integer type	Selects whether to omit a check of the range for conversion between the floating type and unsigned integer type. When "Yes" is specified, code performance of the relevant type conversion processing is improved. The conversion result may, however, differ from C/C++ language specifications, so take care on this point. This corresponds to the -simple_float_conv option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-simple_float_conv)	Omits part of the type conversion processing for the floating type.
	No	Does not omit part of the type conversion processing for the floating type.	
Performs optimization considering the type of the data indicated by the pointer	Selects whether to perform optimization considering the type of the data indicated by the pointer. Although the performance of object code is generally better than when -alias=noansi is specified, the results of execution may differ according to whether -alias=ansi or alias=noansi is specified. This corresponds to the -alias option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-alias=ansi)	Performs optimization considering the type of the data indicated by the pointer.
	No (-alias=noansi)	Does not perform optimization considering the type of the data indicated by the pointer.	

Optimizes modification of the operation order of a floating-point expression	<p>Selects whether to optimize modification of the operation order of a floating-point expression.</p> <p>Specifying the <code>-float_order</code> option generally improves the object performance compared to when not specifying it.</p> <p>However, the accuracy of operations may differ from that when <code>-float_order</code> is not specified.</p> <p>This corresponds to the <code>-float_order</code> option of the compiler.</p> <p>This property is valid only when [2 (-optimize=2)] or [Max (-optimize=max)] in the [Optimization level] property is specified.</p>			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes (-float_order)	Optimizes modification of the operation order in a floating-point expression.	
		No	Does not optimize modification of the operation order in a floating-point expression.	

(5) [Output File]

The detailed information on the output file check is displayed and the configuration can be changed.

Output assembly source file

Select whether to output the assembly source file of the compile result for the C source.

This corresponds to the `-output=src` option of the compiler.

Output assembly source file	Select whether to output the assembly source file of the compile result for the C source. This corresponds to the ?output=src option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes(-output=src)	Outputs the assembly source file of the compile result for the C source.	
No		Does not output the assembly source file of the compile result for the C source.		
Output preprocessed source file	Select whether to output the execution result of preprocessing for the source file to a file. This corresponds to the ?output=prep, -noline option of the compiler.			
	Default	Configuration of the compile option		
	How to change	Select from the drop-down list.		
	Restriction	Yes(-output=prep)	Outputs the execution result of preprocessing for the source file to a file.	
		Yes(Suppress #line)(-output=prep -noline)	Outputs the execution result of preprocessing (suppress #line) for the source file to a file.	
No		Does not output the execution result of preprocessing for the source file to a file.		

(6) [Others]

Other detailed information on compilation is displayed and the configuration can be changed.

Outputs the copyright	Selects whether to output the copyright. This corresponds to the -nologo option of the compiler.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-logo)	Outputs the copyright.
		No (-nologo)	Disables output of the copyright.
Outputs the cross reference information	Selects whether to output cross reference information. It is necessary to change the setting of the property of "Program Analyzer" to change this option.		
	Default	Configuration of the compile option	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-Xcref)	Outputs the cross reference information.
		No	Does not output of the cross reference information.
Commands executed before compile processing	<p>Specifies the command to be executed before compile processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %CompiledFile%: Replaces with the absolute path of the output file under compiling. %InputFile%: Replaces with the absolute path of the file to be compiled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before compile processing.</p> <p>The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>		
	Default	Configuration of the compile option	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 1023 characters	
		Up to 64 items can be specified.	

Commands executed after compile processing	<p>Specifies the command to be executed after compile processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %CompiledFile%: Replaces with the absolute path of the output file under compiling. %InputFile%: Replaces with the absolute path of the file to be compiled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after compile processing.</p> <p>The placeholders can be described in the script.</p> <p>The specified command is displayed as the subproperty.</p>	
	Default	Configuration of the compile option
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.
Other additional options	Inputs the compile options to be added additionally. The options set here are added at the end of the compile options group.	
	Default	Configuration of the compile option
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	The specified option is displayed.	
	Default	Configuration of the compile option
	How to change	Changes not allowed

[Individual Assemble Options] tab

This tab shows the detailed information on an assemble source file categorized by the following and the configuration can be changed.

Note that this tab takes over the settings of the [\[Common Options\] tab](#) and [\[Assemble Options\] tab](#).

When the settings are changed from these tabs, the properties are displayed in boldface.

Remark This tab is displayed when [Yes] in the [Set individual assemble option] property in the [\[Build\]](#) category from the [\[Build Settings\] tab](#) is selected.

(1) [\[Source\]](#)

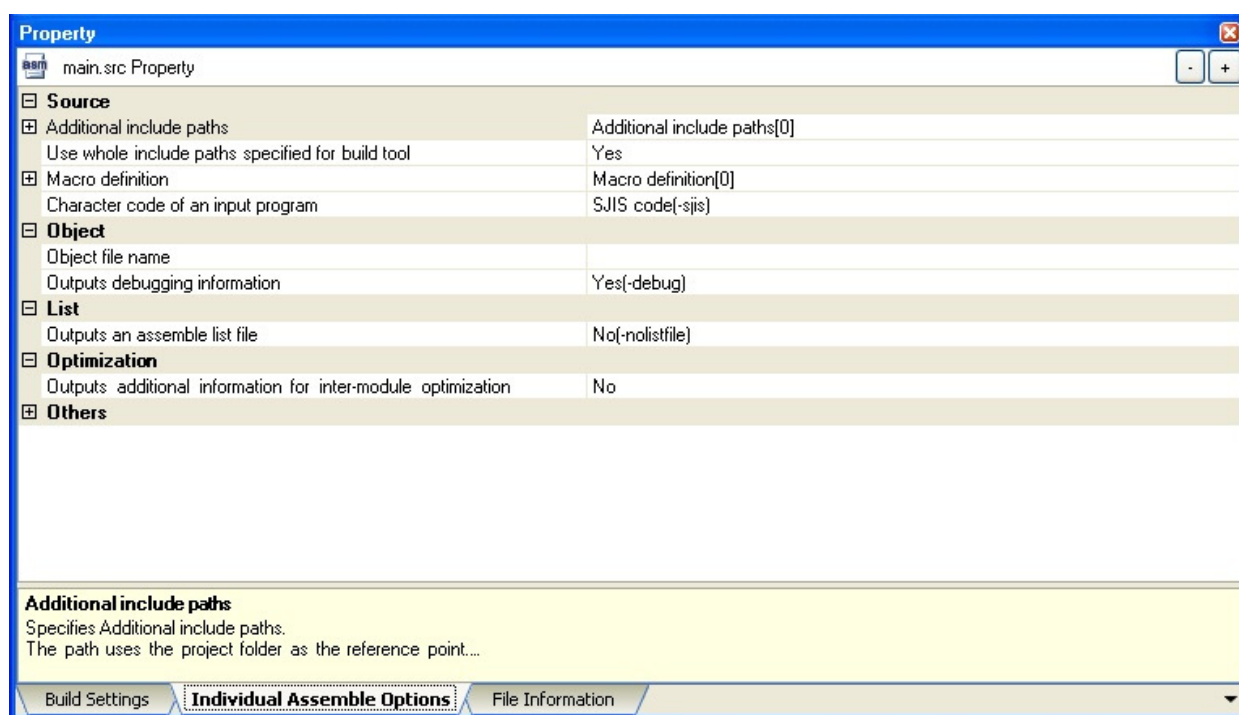
(2) [\[Object\]](#)

(3) [\[List\]](#)

(4) [\[Object\]](#)

(5) [\[Others\]](#)

Figure A.18 Property Panel: [Individual Assemble Options] Tab



[Description of each category]

(1) [\[Source\]](#)

The detailed information on the source is displayed and the configuration can be changed.

Additional include paths	<p>Specifies the name of the path to the folder that stores the include file. The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder.</p> <p>%ActiveProjectName%: Replaces with the active project name.</p> <p>%BuildModeName%: Replaces with the build mode name.</p> <p>%MainProjectDir%: Replaces with the absolute path of the main project folder.</p> <p>%MainProjectName%: Replaces with the main project name.</p> <p>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.</p> <p>%ProjectDir%: Replaces with the absolute path of the project folder.</p> <p>%ProjectName%: Replaces with the project name.</p> <p>%TempDir%: Replaces with the absolute path of the temporary folder.</p> <p>%WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>The reference point of the path is the project folder.</p> <p>This corresponds to the -include option of the assembler.</p> <p>The specified include path is displayed as the subproperty.</p>			
	Default	Additional include paths[<i>number of defined items</i>]		
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.		
	Restriction	Up to 247 characters Up to 65536 items can be specified.		
Use whole include paths specified for build tool	<p>Select whether to compile using the include path specified in the [Additional include paths] property in the [Source] category from the [Assemble Options] tab of the build tool to be used.</p> <p>The include paths are added by the following procedure.</p> <p>Paths specified in the [Additional include paths] property from this tab</p> <p>Paths specified in the [Additional include paths] in the [Source] category from the [Assemble Options] tab</p> <p>This corresponds to the -include option of the assembler.</p>			
	Default	Yes		
	How to change	Select from the drop-down list.		
	Restriction	Yes	Compiles using the include path specified in the property of the build tool to be used.	
		No	Does not use the include path specified in the property of the build tool to be used.	
Macro definition	<p>Specifies the macro name to be defined.</p> <p>Specifies in the format of "<i>macro name=string</i>", with one macro name per line.</p> <p>This corresponds to the -define option of the assembler.</p> <p>The specified macro is displayed as the subproperty.</p>			
	Default	Configuration of the assemble option		
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.		
	Restriction	Up to 32767 characters Up to 65536 items can be specified.		

Character code of an input program	Selects character code of an input program. This corresponds to the -euc , -sjis , -latin1 , -big5 , and -gb2312 option of the assembler.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	EUC code (-euc)	Handles the characters in strings, character constants, and comments by using EUC.
		SJIS code (-sjis)	Handles the characters in strings, character constants, and comments by using SJIS.
		ISO-Latin1 code (-latin1)	Handles the characters in strings, character constants, and comments by using ISO-Latin1.
		Traditional Chinese character (-big5)	Handles the characters in strings, character constants, and comments by using Traditional Chinese character.
		Simplified Chinese character (-gb2312)	Handles the characters in strings, character constants, and comments by using Simplified Chinese character.

(2) [Object]

The detailed information on the object is displayed and the configuration can be changed.

Object module file name	Specify the name of the object module file generated after assembling. The extension other than ".obj" cannot be specified. If the extension is omitted, ".obj" is automatically added. If this is blank, the file name will be the source file name with the extension replaced by ".obj". This corresponds to the -output option of the assembler.			
	Default	Configuration of the assemble option		
	How to change	Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button.		
	Restriction	Up to 259 characters		
Outputs debugging information	Selects whether to output debugging information to object module files. This corresponds to the -debug and -nodebug options of the assembler.			
	Default	Configuration of the assemble option		
	How to change	Select from the drop-down list.		
	Restriction	Yes (-debug)	Outputs debugging information to object module files.	
		No (-nodebug)	Does not output debugging information to object module files.	

(3) [List]

The detailed information on the list is displayed and the configuration can be changed.

Outputs a assemble list file	Selects whether to output an assemble list file. This corresponds to the -listfile and -nolistfile option of the assembler.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-listfile)	Outputs an assemble list file.
		No (-nolistfile)	Does not output an assemble list file.
Outputs the statements unsatisfied in conditional assembly	Specifies the contents of the assemble list file. Selects whether to output the statements unsatisfied in conditional assembly. This corresponds to the -show option of the assembler. This property is displayed only when [Yes (-listfile)] in the [Output a assemble list file] property is selected.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=conditionals)	Outputs the statements unsatisfied in conditional assembly.
		No	Does not output the statements unsatisfied in conditional assembly.

Outputs the information before .DEFINE replacement	Specifies the contents of the assemble list file. Selects whether to output the information before .DEFINE replacement. This corresponds to the -show option of the assembler. This property is displayed only when [Yes (-listfile)] in the [Output a assemble list file] property is selected.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=definitions)	Outputs the information before replacement specified with .DEFINE.
		No	Does not output the information before replacement specified with .DEFINE.
Outputs the assembler macro expansion statements	Specifies the contents of the assemble list file. Selects whether to output the assembler macro expansion statements. This corresponds to the -show option of the assembler. This property is displayed only when [Yes (-listfile)] in the [Output a assemble list file] property is selected.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-show=expansions)	Outputs the macro expansion statements.
		No	Does not output the macro expansion statements.

(4) [Object]

The detailed information on the optimization is displayed and the configuration can be changed.

Output additional information for inter-module optimization	Selects whether to output additional information for inter-module optimization. At linkage, inter-module optimization is applied to files for which this option has been specified. This corresponds to the <code>-goptimize</code> option of the assembler.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-goptimize)	Outputs additional information for inter-module optimization.
		No	Does not output additional information for inter-module optimization.

(5) [Others]

Other detailed information on assembly is displayed and the configuration can be changed.

Checks for a privileged instruction	Selects whether to check for a privileged instruction. This corresponds to the -chkpm option of the assembler.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-chkpm)	Checks for a privileged instruction.
		No	Does not check for a privileged instruction.
Checks for a floating-point operation instruction	Selects whether to check for a floating-point operation instruction. This corresponds to the -chkfpu option of the assembler.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-chkfpu)	Checks for a floating-point operation instruction.
		No	Does not check for a floating-point operation instruction.
Checks for a DSP instruction	Selects whether to check for a DSP instruction. This corresponds to the -chkdsp option of the assembler.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-chkdsp)	Checks for a DSP instruction.
		No	Does not check for a DSP instruction.
Outputs the copyright	Selects whether to output the copyright. This corresponds to the -logo and -nologo option of the assembler.		
	Default	Configuration of the assemble option	
	How to change	Select from the drop-down list.	
	Restriction	Yes (-logo)	Outputs the copyright.
		No (-nologo)	Disables output of the copyright.

Commands executed before assemble processing	<p>Specifies the command to be executed before assemble processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %AssembledFile%: Replaces with the absolute path of the output file under assembling. %BuildModeName%: Replaces with the build mode name. %InputFile%: Replaces with the absolute path of the file to be assembled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicromToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder. When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed before assemble processing. The placeholders can be described in the script. The specified command is displayed as the subproperty.</p>	
	Default	Configuration of the assemble option
	How to change	<p>Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.</p>
	Restriction	<p>Up to 1023 characters Up to 64 items can be specified.</p>

Commands executed after assemble processing	<p>Specifies the command to be executed after assemble processing. Use the call instruction to specify a batch file (example: call a.bat). The following placeholders are supported.</p> <p>%ActiveProjectDir%: Replaces with the absolute path of the active project folder. %ActiveProjectName%: Replaces with the active project name. %AssembledFile%: Replaces with the absolute path of the output file under assembling. %BuildModeName%: Replaces with the build mode name. %InputFile%: Replaces with the absolute path of the file to be assembled. %MainProjectDir%: Replaces with the absolute path of the main project folder. %MainProjectName%: Replaces with the main project name. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the file name of the running program. %ProjectDir%: Replaces with the absolute path of the project folder. %ProjectName%: Replaces with the project name. %TempDir%: Replaces with the absolute path of the temporary folder. %WinDir%: Replaces with the absolute path of the Windows system folder.</p> <p>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed after assemble processing.</p> <p>The placeholders can be described in the script.</p> <p>The specified command is displayed as the subproperty.</p>	
	Default	Configuration of the assemble option
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 1023 characters Up to 64 items can be specified.
Other additional options	Inputs the assemble options to be added additionally. The options set here are added at the end of the assemble options group.	
	Default	Configuration of the assemble option
	How to change	Directly enter to the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Command line	The specified option is displayed.	
	Default	Configuration of the assemble option
	How to change	Changes not allowed

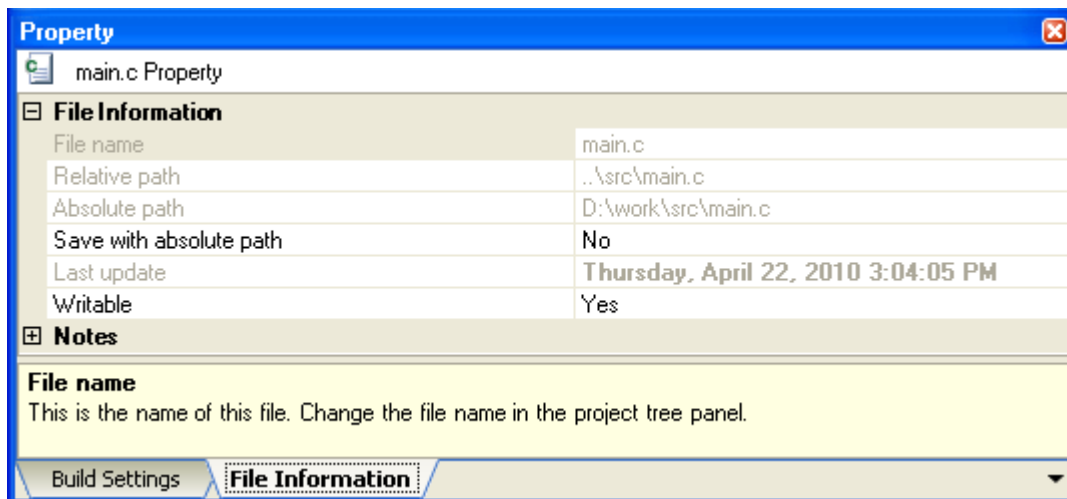
[File Information] tab

This tab shows the detailed information on each file categorized by the following and the configuration can be changed.

(1) [File Information]

(2) [Notes]

Figure A.19 Property Panel: [File Information] Tab



[Description of each category]

(1) [File Information]

The detailed information on the file are displayed and the configuration can be changed.

File name	Display the file name. Change the file name on the Project Tree panel .		
	Default	File name	
	How to change	Changes not allowed	
Relative path	Display the relative path of the file from the project folder.		
	Default	The relative path of the file from the project folder	
	How to change	Changes not allowed	
Absolute path	Display the absolute path of the file.		
	Default	The absolute path of the file	
	How to change	Changes not allowed	
Save with absolute path	Select whether to save the file location with the absolute path.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Saves the file location with the absolute path.
		No	Saves the file location with the relative path.

Last update	Display the time and date on which this file was changed last.		
	Default	File updated time and date	
	How to change	Changes not allowed	
Writable	Select whether to enable writing to the file.		
	Default	Yes (when the file is write enabled) No (when the file is not write enabled)	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Enables the file to write.
		No	Does not enable the file to write.

(2) [Notes]

The detailed information on notes is displayed and the configuration can be changed.

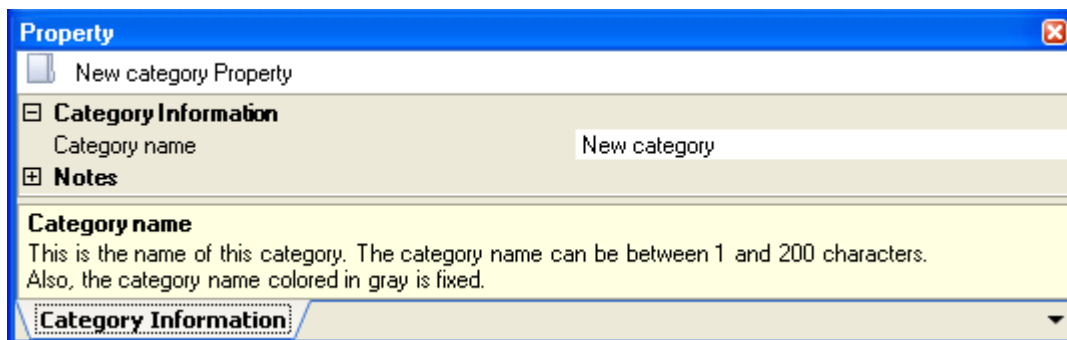
Memo	Add memos to the file. Add one item in one line. The added memos are displayed as the subproperty.		
	Default	Memo[<i>number-of-items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.	
	Restriction	Up to 256 characters Up to 256 memos can be specified.	

[Category Information] tab

This tab shows the detailed information on the category that the user added, File node, and Build tool generated files node categorized by the following and the configuration can be changed.

- (1) [Category Information]
- (2) [Notes]

Figure A.20 Property Panel: [Category Information] Tab



[Description of each category]

- (1) [Category Information]
The detailed information on the category is displayed and the configuration can be changed.

Category name	Specify the category name to categorize files. This property of the File node, and Build tool generated files node is displayed in gray and you cannot change the attribute.	
	Default	Category name of files
	How to change	Directly enter to the text box.
	Restriction	1 to 200 characters

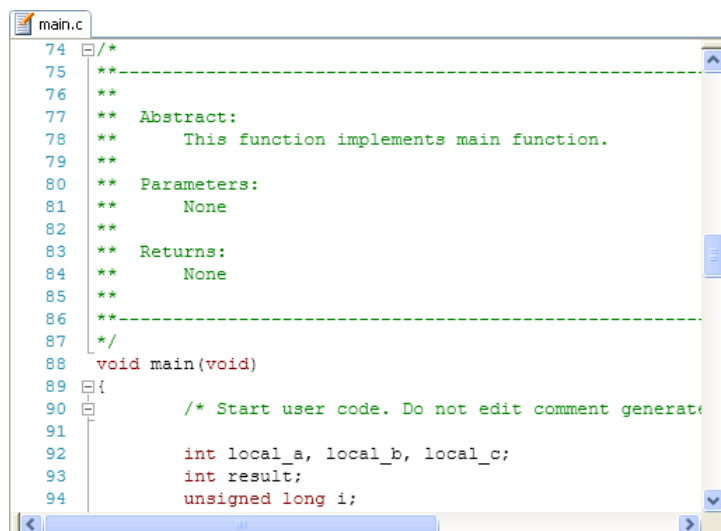
- (2) [Notes]
The detailed information on notes is displayed and the configuration can be changed.
This category of the File node, and Build tool generated files node is not displayed.

Memo	Add memos to the category of files. Add one item in one line. The added memos are displayed as the subproperty.	
	Default	Memo[<i>number-of-items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use the text box directly enter the text.
	Restriction	Up to 256 characters Up to 256 memos can be specified.

Editor panel

This panel is used to display/edit text files/source files.
See "CubeSuite+ RX Coding" for details about this panel.

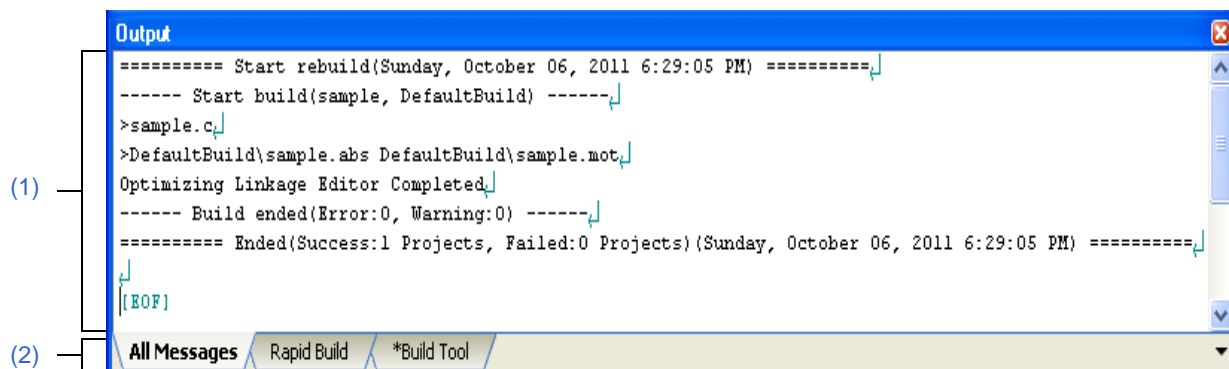
Figure A.21 Editor Panel



Output panel

This panel is used to display the message that is output from the build tool. Messages are shown individually on the tab categorized by the output tool.

Figure A.22 Output Panel



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[\[File\] menu \(only available for the Output panel\)\]](#)
- [\[\[Edit\] menu \(only available for the Output panel\)\]](#)
- [\[Context menu\]](#)

[How to open]

- From the [View] menu, select [Output].

[Description of each area]

(1) Message area

Display messages and the search results output from each tool.

In build result/search result (batch search) display, a new message is displayed deleting the previous message every time build/search is done (but not the [All Messages] tab).

Remark Up to 500000 lines of messages can be displayed. If 500001 lines or more of messages are output, then the excess lines are deleted, oldest first.


The message colors differ as follows depends on the type of the output message (the character color/background color is set in [General - Font and Color] category in the [Option dialog box](#)).

Message Type	Example (Default)			Description
Normal message	AaBbCc	Character color	Black	Information on something.
		Background color	White	
Warning	AaBbCc	Character color	Blue	Warning for the operation.
		Background color	Normal color	
Error message	AaBbCc	Character color	Red	Fatal error or operation disabled because of an error in operation.
		Background color	Light gray	

This area has the following functions.

- (a) Tag jump
When the output message is double-clicked, or the [Enter] key is pressed with the caret over the message, the [Editor panel](#) appears and the destination line number of the file is displayed.
You can jump to the line of the source file that generated the error from the error message output when building.
 - (b) Display help
help with regard to the message in the line is shown by selecting [Help for Message] in the context menu or pressing the [F1] key while the caret is in the line where the warning message or the error message is displayed.
 - (c) Save log
The contents displayed on the currently selected tab can be saved in a text file (*.txt) by selecting [Save Output - *tab name* As...] from the [File] menu and opens the [Save As dialog box](#) (messages on the tab that is not selected will not be saved).
- (2) Tab selection area
Select tabs that messages are output from.
Tabs that are displayed are as follows.

Tab Name	Description
All Messages	Shows all the messages by order of output. (Except while executing a rapid build)
Rapid Build	Shows the message output from the build tool by running a rapid build.
Build Tool	Shows the message output from the build tool by running build/rebuild/clean.

Caution Tab is not automatically switched when a new message is output on the non-selected tab.
If this is the case,  is added to the tab informing a new message is output.

[[File] menu (only available for the Output panel)]

The following items are exclusive for the [File] menu in the Output panel (other items are common to all the panels).

Save Output - <i>tab name</i>	Saves the contents on the currently selecting tab in the previously saved text file (*.txt) (see "(c) Save log "). When this item is selected for the first time after launching the program, the operation is equivalent to when selecting [Save Output - <i>tab name</i> As...].
Save Output - <i>tab name</i> As...	Opens the Save As dialog box to save the contents on the currently selecting tab in the designated text file (*.txt) (see "(c) Save log ").

[[Edit] menu (only available for the Output panel)]

The following items are exclusive to the [Edit] menu in the Output panel (other items are all invalid).

Copy	Copies the selected characters to the clipboard.
Select All	Selects all the messages displayed on this panel.
Find...	Opens the Find and Replace dialog box with the [Quick Search] tab target.
Replace...	Opens the Find and Replace dialog box with the [Whole Replace] tab target.

[Context menu]

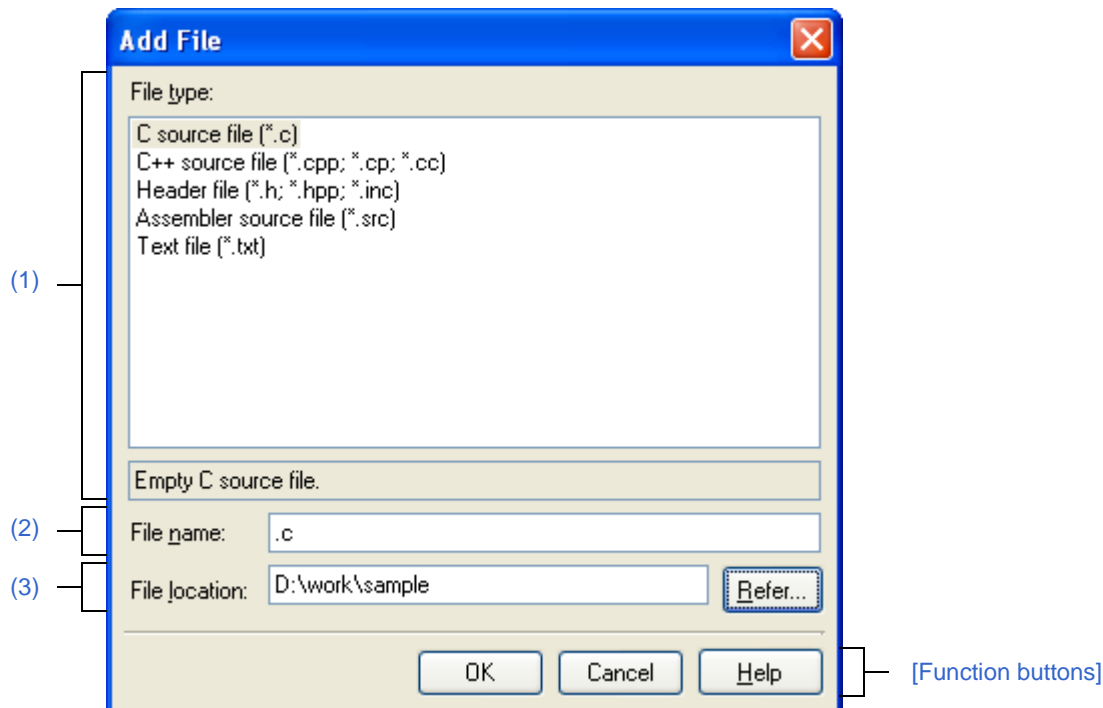
copy	Copies the selected characters to the clipboard.
Select All	Selects all the messages displayed on this panel.
Clear	Deletes all the messages displayed on this panel.

Tag Jump	Jumps to the caret line in the editor indicated by the message (file, line, and column).
Help for Message	Shows the help with regard to the message at the current caret. Note that the help is only for warning/error messages.

Add File dialog box

This dialog box is used to create a new file and add it to the project.

Figure A.23 Add File Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- From the [File] menu, select [Add] >> [Add New File...].
- On the [Project Tree panel](#), select either one of the Project node, Subproject node, File node, or category node, and then select [Add] >> [Add New File...] from the context menu.

[Description of each area]

- (1) [File type] area
 Select file types to create.
 The description is shown at the lower box when a file type is selected.
 File types to be shown are as follows.
- C source file (*.c)
 - C++ source file (*.cpp; *.cp; *.cc)
 - Header file (*.h; *.hpp; *.inc)
 - Assembler source file (*.src; *.s)
 - Python script file (*.py)
 - Text file (*.txt)

(2) [File name] area

Directly enter the name of the file to create.

The default file extension is ".txt".

Remark

If extensions are not designated, the one selected in the [File type] area is added. Also that if extensions different from the one selected in the [File type] area are designated, the one selected in the [File type] area is added as an extension (for example, if you designate "aaa.txt" as a file name and select "C source file (*.c)" as file type, the file is named as "aaa.txt.c").

(3) [File location] area

Designate the location to create a file by directly entering its path or selecting from [Refer...] button.

The default file location is the project folder path.

(a) Button

Refer...	Opens the Browse For Folder dialog box . When a folder is selected, a path is added in the text box.
----------	---

Remarks 1. When the text box is left blank, the project folder is regarded to be designated.

Remarks 2. When the relative path is used, the path is regarded to be from the project folder.

Remark

The number of characters that can be entered in the [File name] area and the [File location] area is up to 259 both for the path name and file name together. When the input violates any restriction, the following messages are shown in the tooltip in the [File name] area.

Message	Description
The file name including the path is too long. Make it within 259 characters.	The file name with the path is more than 259 characters.
The specified path contains a folder that does not exist.	The path includes the folder that does not exist.
The file name or path name is invalid. The following characters cannot be used: •, /, :, *, ?, ", <, >,	The file name with the invalid path is designated. The characters, \, /, :, *, ?, ", <, >, , cannot be used for the file name and folder name.

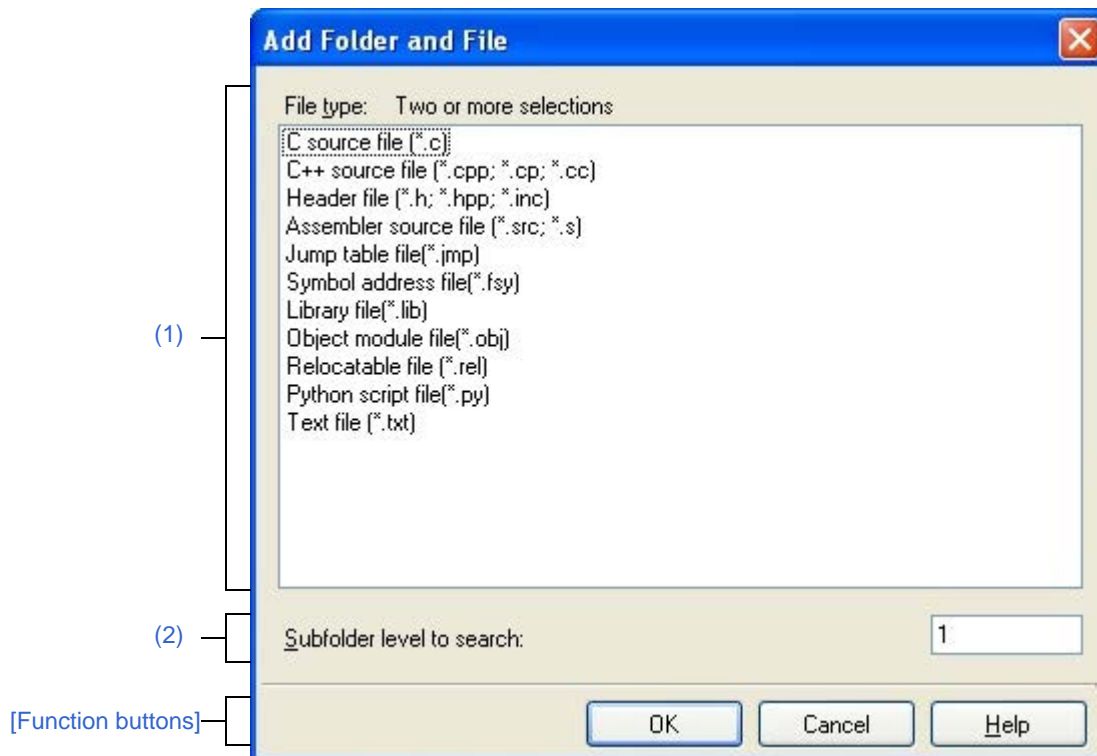
[Function buttons]

Button	Function
OK	Creates the file with the entered file name, adds it to the project, and opens with the Editor panel . Then closes this dialog box.
Cancel	Does not create a file and closes this dialog box.
Help	Displays the help of this dialog box.

Add Folder and File dialog box

This dialog box is used to add existing files and folder hierarchies to the project.
The folder is added as a category.

Figure A.24 Add Folder and File Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- Drag the folder from Explorer or the like, and drop it on the [Project Tree panel](#).

[Description of each area]

- (1) [File type] area
Select the file types to add to the project.
You can select multiple types by left clicking while holding down the [Ctrl] or [Shift] key.
If nothing is selected, it is assumed that all types are selected.
The file types displayed are shown below.
 - C source file (*.c)
 - C++ source file (*.cpp; *.cp; *.cc)
 - Header file (*.h; *.hpp; *.inc)
 - Assembler source file (*.src; *.s)
 - Jump table file (*.jmp)
 - Symbol address file (*.fsy)

- Library file (*.lib)
- Object module file (*.obj)
- Relocatable file (*.rel)
- Python script file (*.py)
- Text file (*.txt)

(2) [Subfolder level to search] area

Directly enter the number of subfolder levels to add to the project.

The default number is "1".

Remark Decimal numbers of up to 10 are allowed. When the input violates any restriction, the following messages are shown in the tooltip.

Message	Description
Fewer than 0 or more than 10 values cannot be specified.	Fewer than 0 or more than 10 subfolder levels have been specified.
Specify in decimal.	A number in other than base-10 format or a string has been specified.

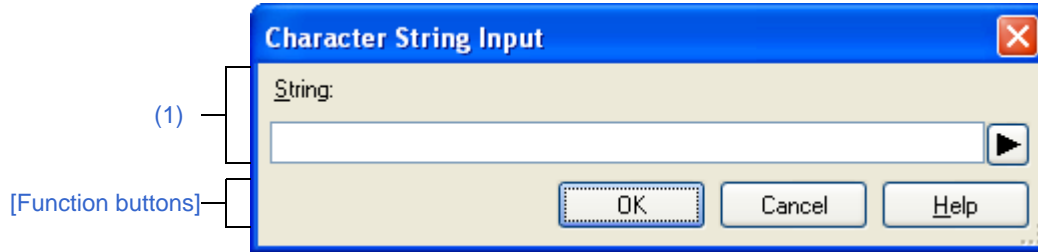
[Function buttons]

Button	Function
OK	The folder that was dragged and dropped and the files in that folder are added to the project. And then close the dialog box.
Cancel	Do not add a folder and files, and then closes this dialog box.
Help	Displays the help of this dialog box.

Character String Input dialog box

This dialog box is used to input and edit characters in one line.

Figure A.25 Character String Input Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), select the following properties, and then click the [...] button.
 - From the [\[Common Options\] tab](#), [Format of build option list] in the [Others] category.
 - From the [\[Compile Options\] tab](#), [Suppresses the number of information-level messages], [Error number of warning-level message], [Error number of information-level message], [Error number of information-level and warning-level message] in the [Source] category, [Other additional options] in the [Others] category.
 - From the [\[Assemble Options\] tab](#), [Other additional options] in the [Others] category.
 - From the [\[Link Options\] tab](#), [Execution start address] in the [Input] category, [Suppresses the number of information-level messages], [Address setting for unused vector area] in the [Output] category, [Error number of warning-level message], [Error number of information-level message], [Error number of information-level and warning-level message], [Other additional options], [Other additional options(Hex/S record/Binary data)] in the [Others] category.
 - From the [\[Librarian Options\] tab](#), [Suppresses the number of information-level messages] in the [Output] category, [Error number of warning-level message], [Error number of information-level message], [Error number of information-level and warning-level message], [Other additional options] in the [Others] category.
 - From the [\[Library Generate Options\] tab](#), [Other additional options] in the [Others] category.
 - From the [\[Individual Compile Options\(C\)\] tab](#), [Suppresses the number of information-level messages], [Error number of warning-level message], [Error number of information-level message], [Error number of information-level and warning-level message] in the [Source] category, [Other additional options] in the [Others] category.
 - From the [\[Individual Compile Options\(C++\)\] tab](#), [Suppresses the number of information-level messages], [Error number of warning-level message], [Error number of information-level message], [Error number of information-level and warning-level message] in the [Source] category, [Other additional options] in the [Others] category.
 - From the [\[Individual Assemble Options\] tab](#), [Other additional options] in the [Others] category.
- In the [General - External Tools] category of the Option dialog box, check [Require options at start-up] in the New registration area. Then the dialog box automatically opens when an external tool is launched from [Tool] menu.

[Description of each area]


- (1) Characters input area
Input characters.
- (a) [String]
Input characters in one line.

By default, the current value of the area that this dialog box is called from is reflected to this area.
You cannot start a new line.

Remark Up to 32767 characters can be entered.
When the input violates any restriction, the following messages will be shown in the tooltip.

Message	Description
More than <i>maximum number of restriction in the property that called this dialog box</i> characters cannot be specified.	The numbers of input characters exceeds the maximum number of restriction in the property that called this dialog box.

(b) Button

	The placeholders which can be specified for the area that this dialog box is called from is displayed in a popup (ascending order). If a placeholder is selected, the string will be surrounded with percentage signs ("%"), and displayed in [String].
---	--

Caution This button is displayed only when the caller of this dialog box supports placeholders.

Remark The placeholders which can be specified differ depending on the area that this dialog box is called from.
For the specific placeholder, see the description of the area that this dialog box is called from.

[Function buttons]

Button	Function
OK	Reflects the entered characters to the property that called this dialog box then closes the dialog box.
Cancel	Does not reflect the entered characters to the property that called this dialog box then closes the dialog box.
Help	Displays the help of this dialog box.

Text Edit dialog box

This dialog box is used to input and edit texts in multiple lines.

Figure A.26 Text Edit Dialog Box (When Caller Supports Placeholders)

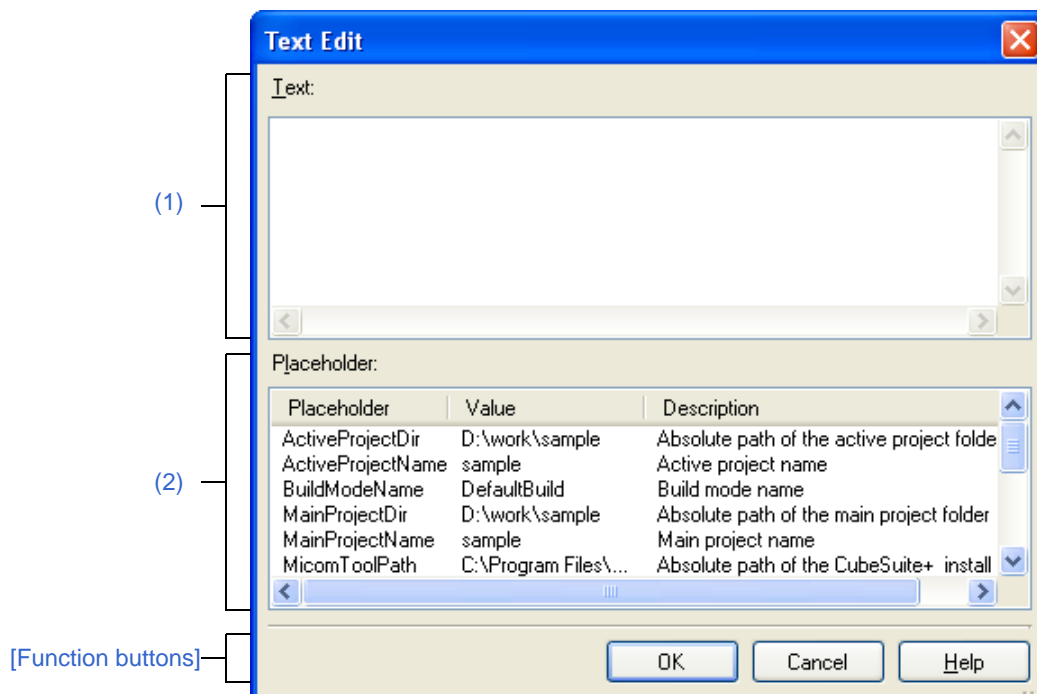
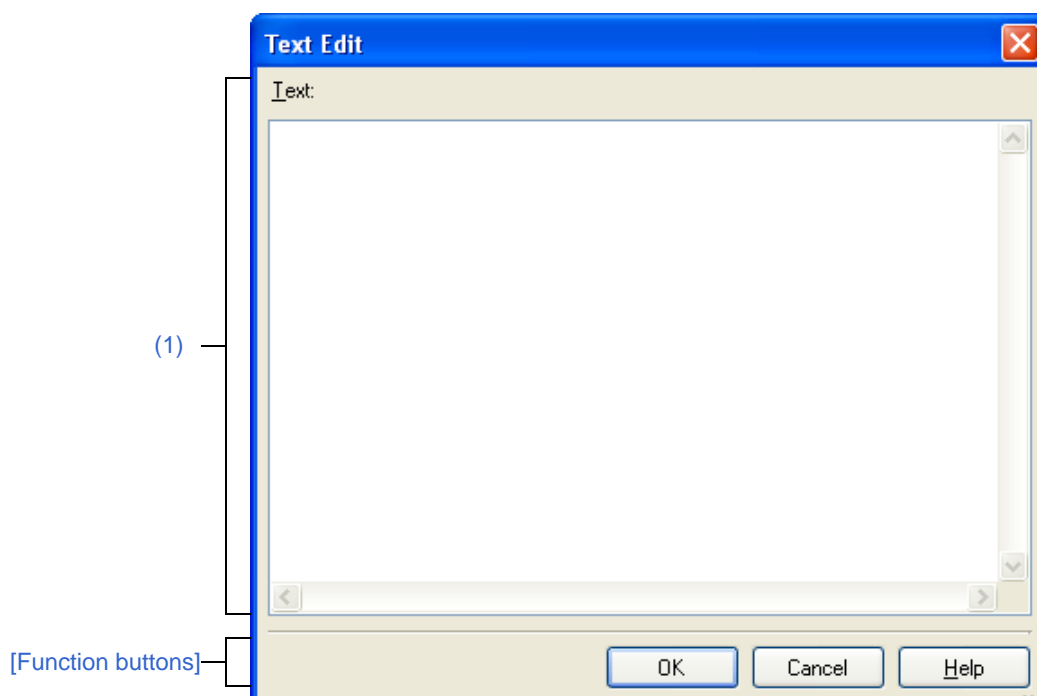


Figure A.27 Text Edit Dialog Box (When Caller Does Not Support Placeholders)



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), select the following properties, and then click the [...] button.
 - From the [\[Common Options\] tab](#), [Macro definition] in the [Frequently Used Options(for Compile)] category, [Macro definition] in the [Frequently Used Options(for Assemble)] category, [Using libraries] in the [Frequently Used Options(for Link)] category, [Memo] in the [Notes] category, and [Commands executed before build processing], [Commands executed after build processing] in the [Others] category.
 - From the [\[Compile Options\] tab](#), [Include files at the head of compiling units], [Macro definition] in the [Source] category, [Rule check exclusion file] in the [MISRA C rule check] category, and [Commands executed before compile processing], [Commands executed after compile processing] in the [Others] category.
 - From the [\[Assemble Options\] tab](#), [Macro definition] in the [Source] category, and [Commands executed before assemble processing], [Commands executed after assemble processing] in the [Others] category.
 - From the [\[Link Options\] tab](#), [Input object module file], [Using libraries], [Input binary data file], [Symbol definition] in the [Input] category, [ROM to RAM mapped section], [Specifies divide output file], [Address setting for specified vector number], [The section that outputs a jump table to branch to external definition symbols] in the [Output] category, [Division conversion file], [Target range] in the [Convert Load Module File] category, [Unreferenced symbol that disables deletion by optimization], [Same-code that disables unification regarding optimization], [Section to disable optimization], [Address range to disable optimization] in the [Optimization] category, [The specified section that outputs externally defined symbols to the file], [Section alignment] in the [Section] category, [Address range of the memory type], [Not divide the specified section] in the [Verify] category, and [Commands executed before link processing], [Commands executed after link processing] in the [Others] category.
 - From the [\[Librarian Options\] tab](#), [Input object module file], [Using libraries], [Input binary data file] in the [Input] category, and [Commands executed before link processing], [Commands executed after link processing] in the [Others] category.
 - From the [\[Library Generate Options\] tab](#), [Commands executed before library generate processing], [Commands executed after library generate processing] in the [Others] category.
 - From the [\[Individual Compile Options\(C\)\] tab](#), [Include files at the head of compiling units], [Macro definition] in the [Source] category, [Rule check exclusion] in the [MISRA C rule check] category, [Commands executed before compile processing] and [Commands executed after compile processing] in the [Others] category.
 - From the [\[Individual Compile Options\(C++\)\] tab](#), [Include files at the head of compiling units], [Macro definition] in the [Source] category, and [Commands executed before compile processing], [Commands executed after compile processing] in the [Others] category.
 - From the [\[Individual Assemble Options\] tab](#), [Macro definition] in the [Source] category, and [Commands executed before assemble processing], [Commands executed after assemble processing] in the [Others] category.
 - From the [\[File Information\] tab](#), [Memo] in the [Notes] category
 - From the [\[Category Information\] tab](#), [Memo] in the [Notes] category

[Description of each area]

(1) [Text]

Input and edit texts in multiple lines.

By default, this dialog box opens with its edit box reflecting the current value of the property selected to call the dialog box.

Remark Up to 65535 lines and 65535 characters are allowed. When the input violates any restriction, the following messages are shown in the tooltip.

Message	Description
More than <i>maximum number of restriction in the property that called this dialog box</i> characters cannot be specified. The current number of characters is displayed between brackets at the beginning of the line in excess of the limit.	The characters exceeds the maximum number of restriction in the property that called this dialog box.

- (2) [Placeholder]
The list of placeholders which can be specified for the area that this dialog box is called from is displayed (ascending order).
Double click a row to surround the placeholder with percentage signs ("%") and display it in [Text].
- (a) [Placeholder]
This area displays the placeholder.
- (b) [Value]
This area displays the string after replacement with the placeholder.
- (c) [Description]
This area displays the description of the placeholder.
- Caution** This area is displayed only when the caller of this dialog box supports placeholders.
- Remark** The placeholders which can be specified differ depending on the area that this dialog box is called from.
For the specific placeholder, see the description of the area that this dialog box is called from.

[Function buttons]

Button	Function
OK	Reflects the entered text to the text box that opened this dialog box and closed the dialog box.
Cancel	Does not reflect the entered text to the text box that opened this dialog box and closed the dialog box.
Help	Displays the help of this dialog box.

Path Edit dialog box

This dialog box is used to edit or add the path or file name including the path.

Figure A.28 Path Edit Dialog Box (When Editing Path)

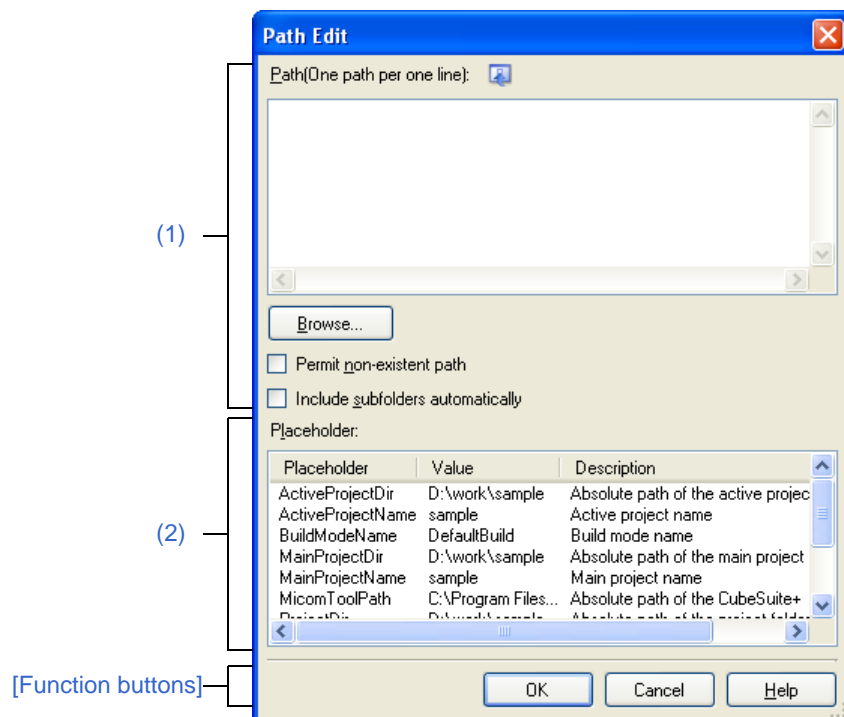
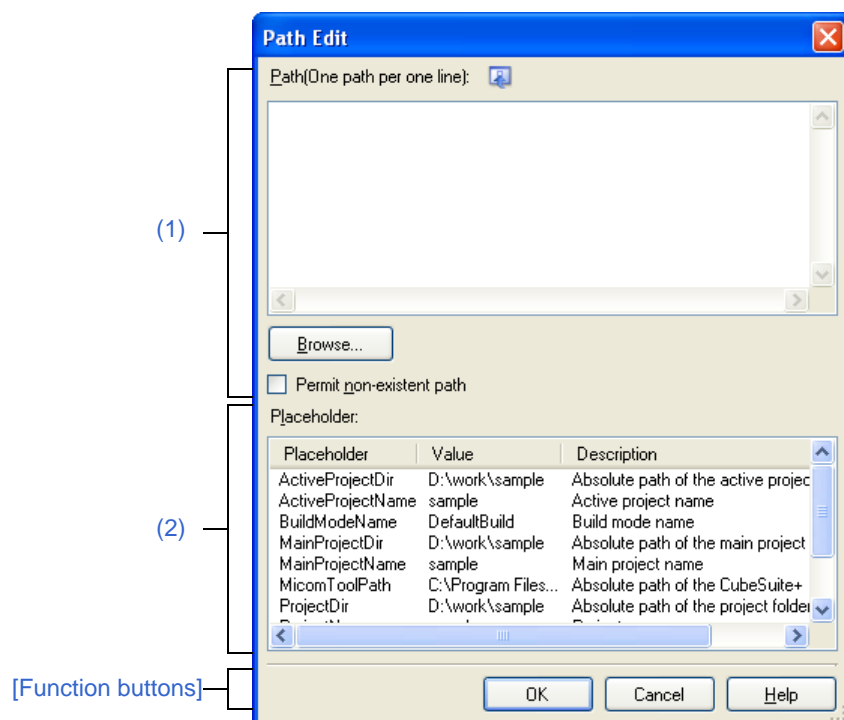


Figure A.29 Path Edit Dialog Box (When Editing File Name Including Path)



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)

- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), select the following properties, and then click the [...] button.
 - From the [\[Common Options\] tab](#), [Additional include paths] in the [Frequently Used Options(for Compile)] category, [Additional include paths] in the [Frequently Used Options(for Assemble)] category, and [Using libraries] in the [Frequently Used Options(for Link)] category
 - From the [\[Compile Options\] tab](#), [Additional include paths] in the [Preprocess] category, and [Rule check exclusion file] in the [MISRA-C:2004 Rule Check] category
 - From the [\[Assemble Options\] tab](#), [Additional include paths] in the [Preprocess] category
 - From the [\[Link Options\] tab](#), [Using libraries] in the [Library] category
 - From the [\[Librarian Options\] tab](#), [Using libraries] in the [Library] category
 - From the [\[Individual Compile Options\(C\)\] tab](#), [Additional include paths] in the [Preprocess] category, and [Rule check exclusion file] in the [MISRA-C:2004 Rule Check] category
 - From the [\[Individual Assemble Options\] tab](#), [Additional include paths] in the [Preprocess] category

[Description of each area]

- (1) Path edit area
Edit or add the path or file name including the path.
 - (a) [\[Path\(One path per one line\)\]](#)
Edit or add the path or file name including the path by directly entering it.
The path or file name including the path can be designated in multiple lines. Designate the path or file name including the path at a line.
By default, the current contents of the text box that opened this dialog box are reflected in this area.

The path can be added by one of the following methods.

- Click the [\[Browse...\]](#) button, and then select the folder in the [Browse For Folder dialog box](#).
- Drag and drop the folder using such as Explorer.

The file names including the path can be added by one of the following methods.

- Click the [\[Browse...\]](#) button, and then select the file in the [Add Excluding File dialog box](#).
- Drag and drop the file using such as Explorer.

Caution If an extremely long absolute path is specified as a relative path, an error could occur when clicking the [OK] button. In this case, designate the absolute path.

Remark Up to 10000 lines can be entered. Up to the maximum characters that are limited by the Windows OS can be entered.
When the input violates any restriction, the following messages will be shown in the tooltip.

Message	Description
Specify a path.	The line contains space characters only.
The path is too long. Specify a path with a number of characters equal to or fewer than <i>maximum number of restriction in the property that called this dialog box</i> .	The file name including the path exceeds the maximum number of restriction in the property that called this dialog box.
The specified path contains a folder that does not exist.	The path contains a folder that does not exist.

Message	Description
The file name or path name is invalid. The following characters cannot be used: \, /, :, *, ?, ", <, >,	The file name with the invalid path is designated. The following characters cannot be used for the file name and folder name: \, /, :, *, ?, ", <, >,
More than <i>maximum number of paths or files specified by the caller</i> lines cannot be specified.	The number of paths or files which have been entered exceeds the maximum number of restriction in the property that called this dialog box.

(b) Button

Browse...	- When adding the path Opens the Browse For Folder dialog box . If a folder is selected, the path will be added in [Path(One path per one line)].
-----------	---

(c) [Permit non-existent path]

When this check box is selected, the existence of the path specified in [Path(One path per one line)] or the validity of the character string specified in the path is not checked.

(d) [Include subfolders automatically]

Select this check box and then click the [Browse...] button to specify the path. The path will be added, including subfolders, to [Path(One path per one line)] (up to five levels deep).

Caution This item is displayed only when adding the path.

(2) [Placeholder] area

The list of placeholders which can be specified for the area that this dialog box is called from is displayed (ascending order).

Double click a row to surround the placeholder with percentage signs ("%") and display it in the path edit area.

(a) [Placeholder]

This area displays the placeholder.

(b) [Value]

This area displays the string after replacement with the placeholder.

(c) [Description]

This area displays the description of the placeholder.

Caution This area is displayed only when the caller of this dialog box supports placeholders.

Remark The placeholders which can be specified differ depending on the area that this dialog box is called from.
For the specific placeholder, see the description of the area that this dialog box is called from.

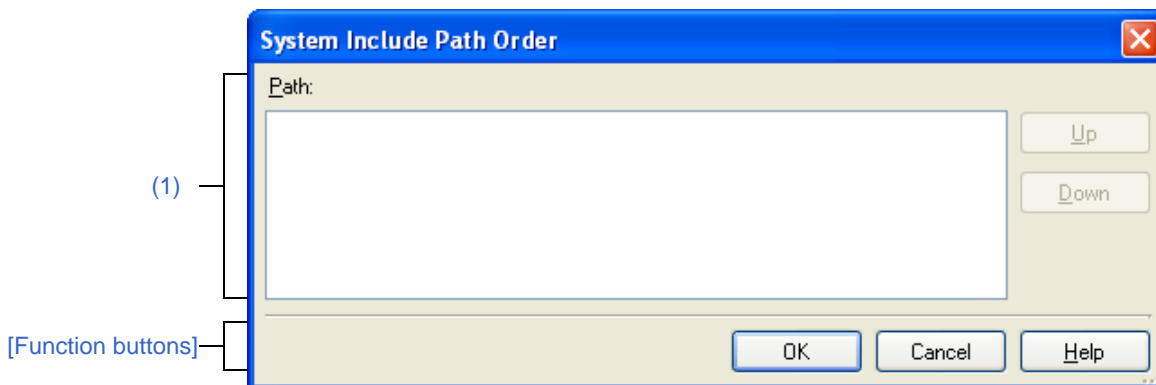
[Function buttons]

Button	Function
OK	Reflects the entered path to the text box that opened this dialog box and closes this dialog box.
Cancel	Does not reflect the entered path to the text box that opened this dialog box and closes this dialog box.
Help	Displays the help of this dialog box.

System Include Path Order dialog box

This dialog box is used to refer the system include paths specified for the compiler and set their specified sequence.

Figure A.30 System Include Path Order Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Property panel](#), select the following properties, and then click the [...] button.
 - From the [\[Common Options\] tab](#), [System include paths] in the [Frequently Used Options(for Compile)] category, and [System include paths] in the [Frequently Used Options(for Assemble)] category
 - From the [\[Compile Options\] tab](#), [System include paths] in the [Source] category
 - From the [\[Assemble Options\] tab](#), [System include paths] in the [Source] category
 - From the [\[Link Options\] tab](#), [System library file] in the [Input] category

[Description of each area]

(1) Path list display area

This area displays the list of the system include paths specified for the compiler.

(a) [Path]

This area displays the list of the system include paths in the specified sequence for the compiler.

The default order is the order that the files are registered to the project.

By changing the display order of the paths, you can set the specified order of the paths to the compiler.

To change the display order, use the [Up] and [Down] buttons, or drag and drop the path names.

Remarks 1. Move the mouse cursor over a file name to display a tooltip with the absolute path of that file.

Remarks 2. Newly added system include paths are added next to the last path of the list.

Remarks 3. When the path names are dragged and dropped, the multiple path names which are next to each other can be selected together.

(b) Button

Up	Moves the selected path to up.
Down	Moves the selected path to down.

Remark Note that above buttons are disabled when any path is not selected.

[Function buttons]

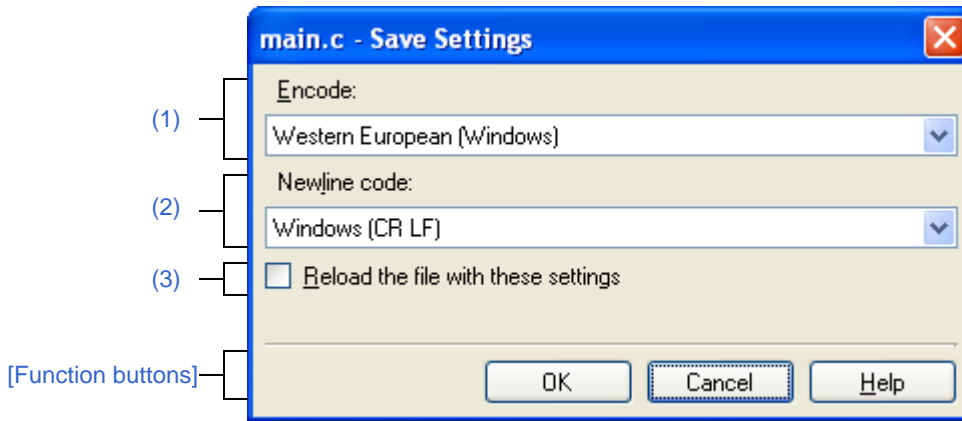
Button	Function
OK	Sets the specified order of the paths to the compiler as the display order in the Path list display area and closes this dialog box.
Cancel	Cancels the specified order of the paths and closes the dialog box.
Help	Displays the help of this dialog box.

Save Settings dialog box

This dialog box is used to set the encoding and newline code of the file that is being edited on the [Editor panel](#).

Remark The target file name is displayed on the title bar.

Figure A.31 Save Settings Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- Focus the [Editor panel](#), and then select [*file name* Save Settings...] from the [File] menu.

[Description of each area]

- (1) [Encode]

Select the encoding to be set from the drop-down list.
The items of the drop-down list are displayed according to the following sequence.
Note that the same encoding and encoding which are not supported by the current OS will not be displayed.

 - Current encoding of the file (default)
 - Default encoding of the current OS
 - Most recently used encodings (maximum 4)
 - *Popular encodings for current locale*
(e.g. for United States locale it will be:
 - Western European (Windows)
 - Unicode (UTF-8)
 - All other encodings supported by the OS (in alphabetical order)
- (2) [Newline code]

Select the newline code to be set from the drop-down list.
You can select any of items below.

 - Windows (CR LF)
 - Macintosh (CR)
 - Unix (LF)

An active newline entry is selected by default.

(3) [Reload the file with these settings]

<input checked="" type="checkbox"/>	Reloads the file with the selected encoding and newline code when the [OK] button is clicked.
<input type="checkbox"/>	Does not reload the file when the [OK] button is clicked (default).

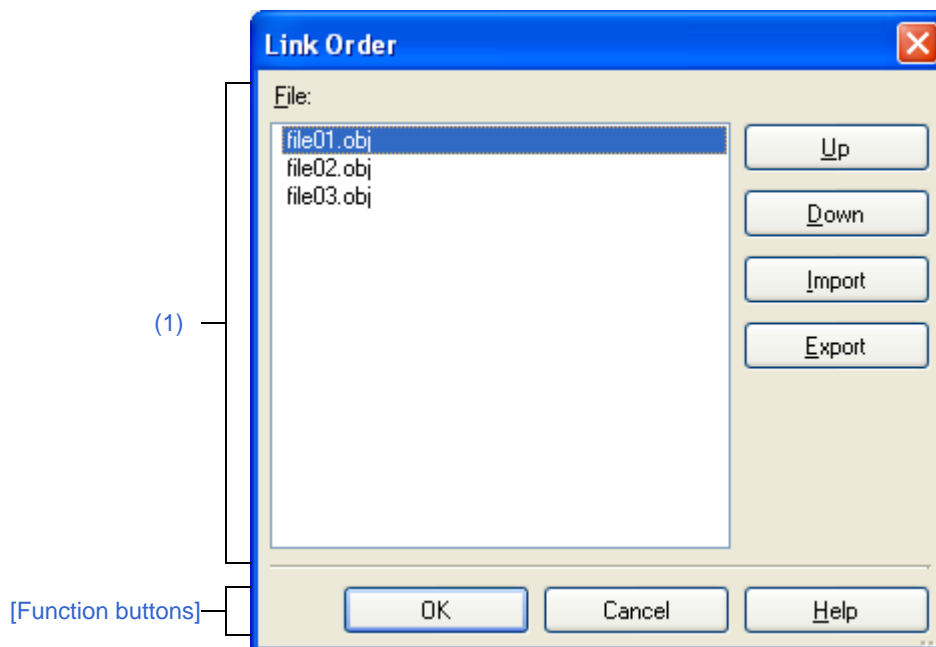
[Function buttons]

Button	Function
OK	Sets the selected encoding and newline code to the target file and closes this dialog box. If [Reload the file with these settings] is selected, sets the selected encoding and newline code to the target file and reloads the file. And then closes this dialog box.
Cancel	Cancels the settings and closes this dialog box.
Help	Displays the help of this dialog box.

Link Order dialog box

This dialog box is used to refer object module files and library files to be input to the linker and configure these link orders.

Figure A.32 Link Order Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Project Tree panel](#), select the Build tool node, and then select [Set Link Order...] from the context menu.

[Description of each area]

(1) File list display area

This area displays the file list to be input to the linker.

(a) [File]

The names of the following files are listed in the order that the files are input to the linker.

- Object module files which are generated from the source files added to the selected main project or subproject
- Object module files which are added directly to the project tree of the selected main project or subproject

The default order is the order that the files are added to the project.

By changing the display order of the files, you can set the input order of the files to the linker.

To change the display order, use the [Up] and [Down] buttons, or drag and drop the file names.

Remarks 1. When the mouse cursor is hovered over a file name, the path of the file appears in a popup. If the file is on the same drive as the project file, then it appears as the relative path; if it is on the different drive, then it appears as the absolute path.

Remarks 2. Object module files which are generated from newly added source files and newly added object module files are added after the last object module file in the list.

Remarks 3. When the file is dragged and dropped, the multiple files that are next to each other can be selected together.

(b) Button

Up	Moves the selected file to up. If any file is not selected, this button will be disabled.
Down	Moves the selected file to down. If any file is not selected, this button will be disabled.
Import	Opens the Select Import File dialog box . The description order of the file names are acquired from the selected link order specification file, and then they are reflected in [File]. If nothing is displayed in [File], this button will be disabled.
Export	Opens the Select Export File dialog box . Outputs the list of the file names displayed in [File] to the specified link order specification file. If nothing is displayed in [File], this button will be disabled.

Remark See ["2.12.2 Set the link order of files"](#) for the method of using the link order specification file.

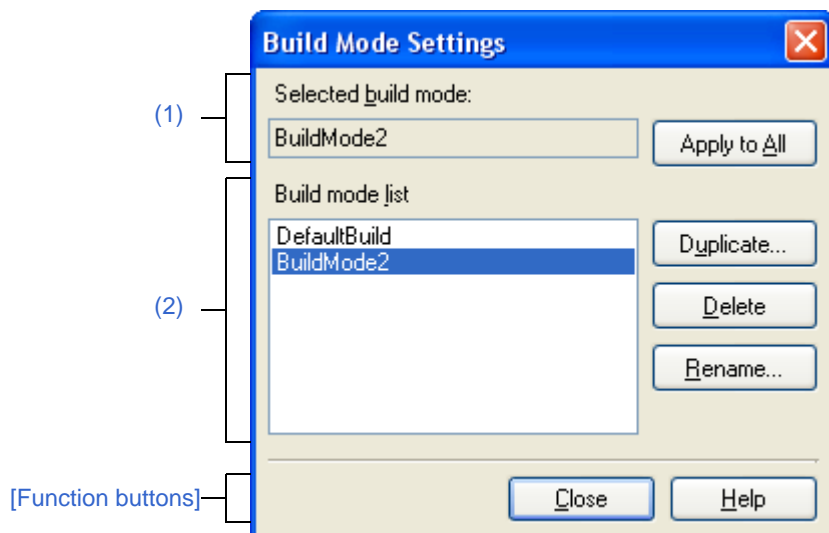
[Function buttons]

Button	Function
OK	Sets the input order of the files to the linker as the display order in [File] and closes this dialog box.
Cancel	Cancels the link order settings and closes the dialog box.
Help	Displays the help of this dialog box.

Build Mode Settings dialog box

This dialog box is used to add and delete build modes and configure the current build mode in batch.

Figure A.33 Build Mode Settings Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- From the [Build] menu, select [Build Mode Settings...].

[Description of each area]

- (1) [Selected build mode] area
Show the build mode selected in the [\[Build mode list\]](#) area.

(a) Button

Apply to All	Sets the build mode of the main project and all subprojects of the currently opened project to the currently displayed build mode.
--------------	--

- (2) [Build mode list] area
Show all the build modes that exist in the currently opening project (main project and subproject) in a list.
Current build mode in the selected project is selected by default.
The current build modes of all projects are same, the build mode is selected by default. If they are not same, "DefaultBuild" will be selected.
Note that the "DefaultBuild" is the default build mode and is always shown at the top.

(a) Button

Duplicate...	Duplicates the selected build mode. The Character String Input dialog box opens and the build mode is duplicated with the name entered and added to the main project and all the subprojects in the currently opening project. When the build mode with "*" mark does not exist in the main project or subproject and duplicate the build mode, DefaultBuild is duplicated. Up to 20 build modes can be added.
--------------	---

Delete	Deletes the selected build mode. Note that DefaultBuild cannot be deleted.
Rename...	Renames the selected build mode. Rename the build mode with entered name in the opening the Character String Input dialog box .

Caution When duplicating or renaming the build mode, the existing build mode name cannot be used.

Remarks 1. Up to 127 characters can be used as a build mode name. When the input violates any restriction, the following messages are shown in the tooltip.

Message	Description
A build mode with the same name already exists.	The entered build mode name already exists.
More than 127 characters cannot be specified.	Build mode name is too long (more than 128 characters).
The build mode name is invalid. The following characters cannot be used: •, /, :, *, ?, ", <, >,	Invalid build mode name is entered. The characters, (\, /, :, *, ?, ", <, >,) cannot be used as the name is used for the folder name.

Remarks 2. Up to 20 build modes can be added. When the input violates any restriction, the following messages are shown in the tooltip.

Message	Description
The maximum number of build modes that can be set per project/subproject is 20.	The number of build modes exceed 20.

[Function buttons]

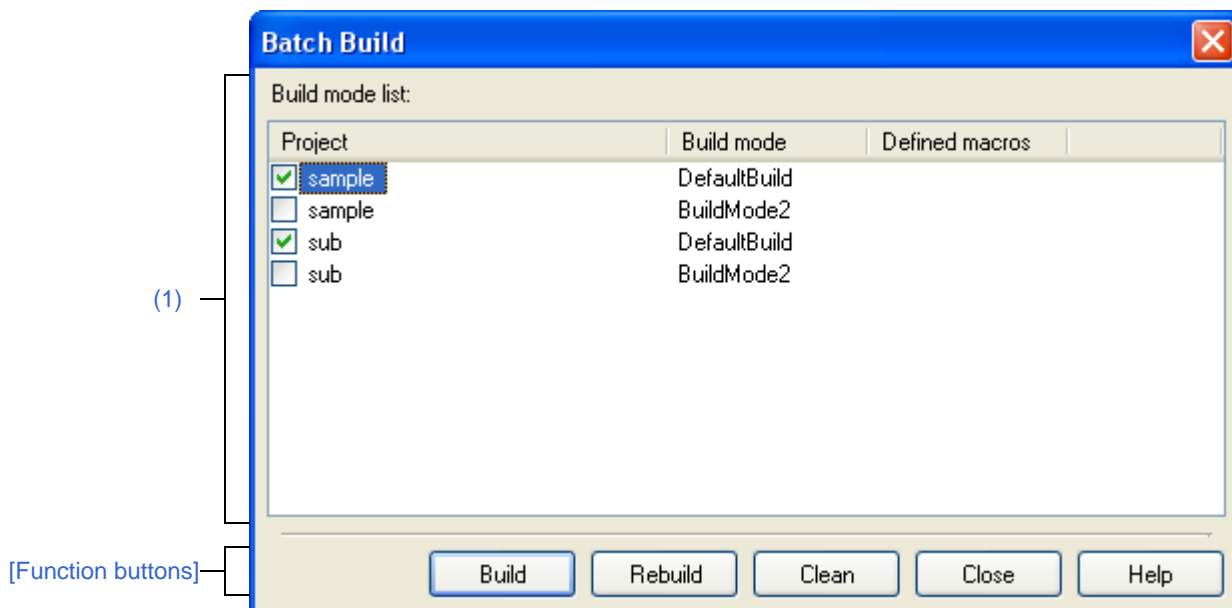
Button	Function
Close	Closes this dialog box.
Help	Displays the help of this dialog box.

Batch Build dialog box

This dialog box is used to do build, rebuild and clean process in batch with the build mode that each project (main project and subproject) has.

- Remark Order of the batch build follows the build order of the project which the subproject comes before the main project.
 When more than one build mode is selected for a main project or a subproject, all the selected build modes are built and then the next subproject or main project is built.

Figure A.34 Batch Build Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Build] menu, select [Batch Build...].

[Description of each area]

- (1) [Build mode list] area
 Show the combination list of the names of the main project and the subproject which the currently opening project has and build modes which they have.
 - (a) [Project]
 Show the main project and the subproject which the currently opening project has.
 Select the combination of the main project and subproject to build and the build modes.
 When this dialog box is opened for the first time after the project is created, all the check boxes are unchecked.
 From the second time, the previous setting is retained.
 - (b) [Build mode]
 Show build modes which the main project and subproject have.

[Function buttons]

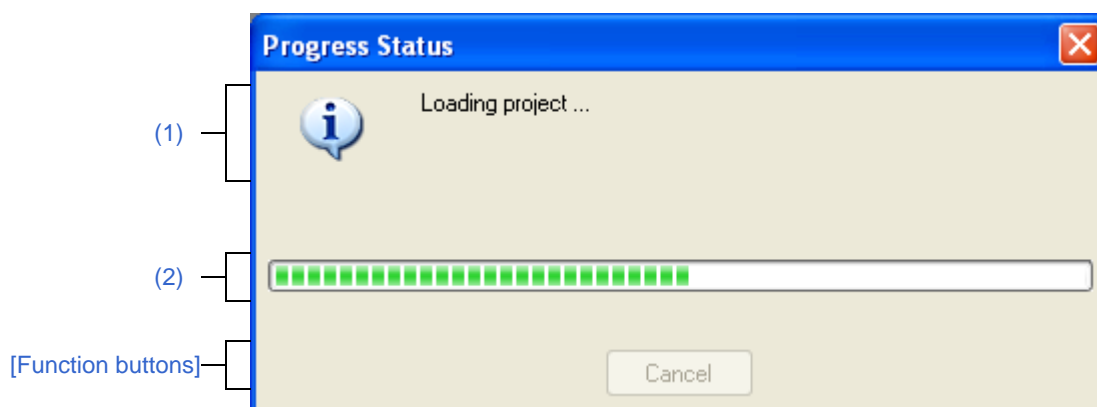
Button	Function
Build	<p>Closes this dialog box and executes a batch build of the selected projects in the respective build modes. The execution result of the build are displayed on the Output panel.</p> <p>After the batch build is complete, the build mode configuration restores to the one before this dialog box was opened.</p> <p>Note that this buttons is disabled when any project is not selected.</p>
Rebuild	<p>Closes this dialog box and executes a batch rebuild of the selected projects in the respective build modes. The execution result of the rebuild are displayed on the Output panel.</p> <p>After the batch rebuild is complete, the build mode configuration restores to the one before this dialog box was opened.</p> <p>Note that this buttons is disabled when any project is not selected.</p>
Clean	<p>Closes this dialog box and deletes the files built in the respective build modes set for the selected projects. The execution result of the clean are displayed on the Output panel.</p> <p>After the clean is complete, the build mode configuration restores to the one before this dialog was opened.</p> <p>Note that this buttons is disabled when any project is not selected.</p>
Close	Closes this dialog box.
Help	Displays the help of this dialog box.

Progress Status dialog box

This dialog box is used to show how the process has been progressed when the time consuming process is taken place.

This dialog box automatically closes when the process in progress is done.

Figure A.35 Progress Status Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- This dialog box automatically opens when a message is output while the time consuming process is in progress.

[Description of each area]

- (1) Message display area
Display the message output while process is in progress (edit not allowed).
- (2) Progress bar
The progress bar shows the current progress of the process in progress with the bar length.
When the process is 100% done (the bar gets to the right end), this dialog box automatically closed.

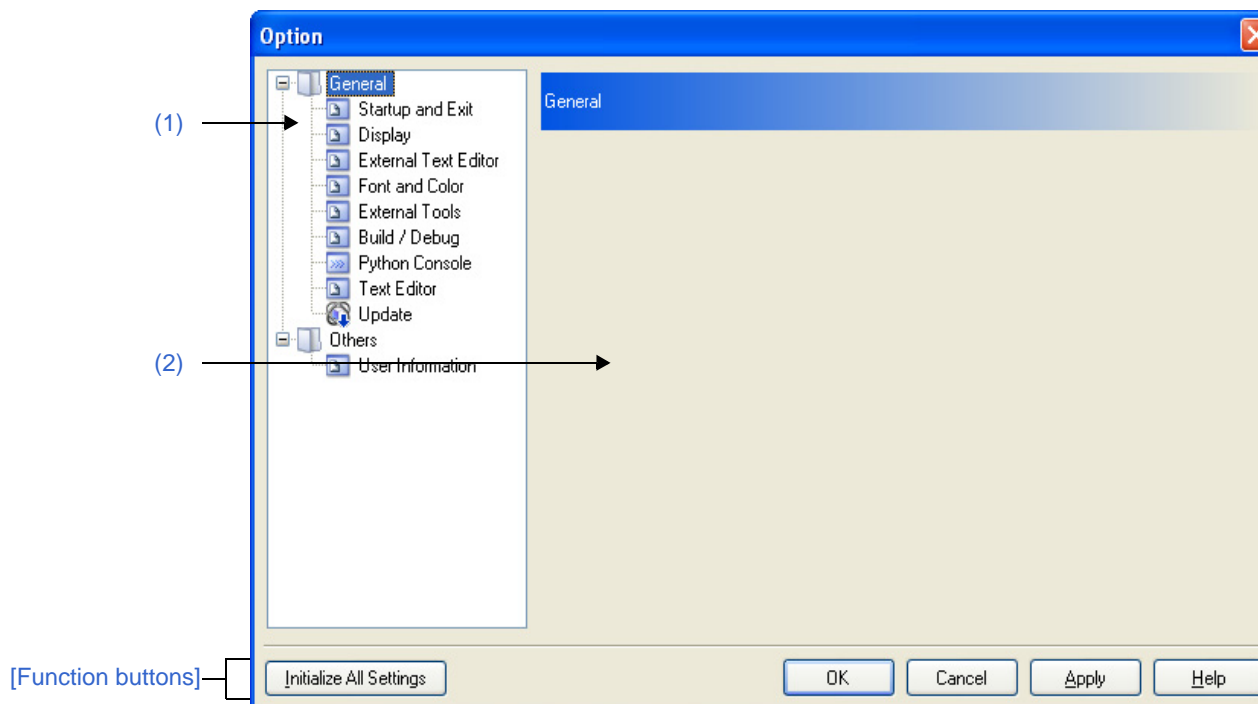
[Function buttons]

Button	Function
Cancel	Cancels the process in progress and closes this dialog box. Note that if the process termination is impossible, this button is disabled.

Option dialog box

This dialog box is used to configure the CubeSuite+ environment.
All settings made via this dialog box are saved as preferences for the current user.

Figure A.36 Option Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Options...].

[Description of each area]

- (1) Category selection area
Select the items to configure from the following categories.

Category	Description
[General - Startup and Exit] category	Configure startup and shutdown.
[General - Display] category	Configure messages from the application.
[General - External Text Editor] category	Configure the external text editor.
[General - Font and Color] category	Configure the fonts and colors shown on each panel.
[General - External Tools] category	Configure the startup of external tools.
[General - Build/Debug] category	Configure building and debugging.

Category	Description
[General - Python Console] category	Configure the Python console.
[General - Text Editor] category	Configure the text editor.
[General - Update] category	Configure update.
[Other - User Information] category	Configure user information.

Remark See "CubeSuite+ Integrated Development Environment User's Manual: Start" for details about categories other than [General - Build/Debug].

(2) Settings

This area is used to configure the various options for the selected category.

For details about configuration for a particular category, see the section for the category in question.

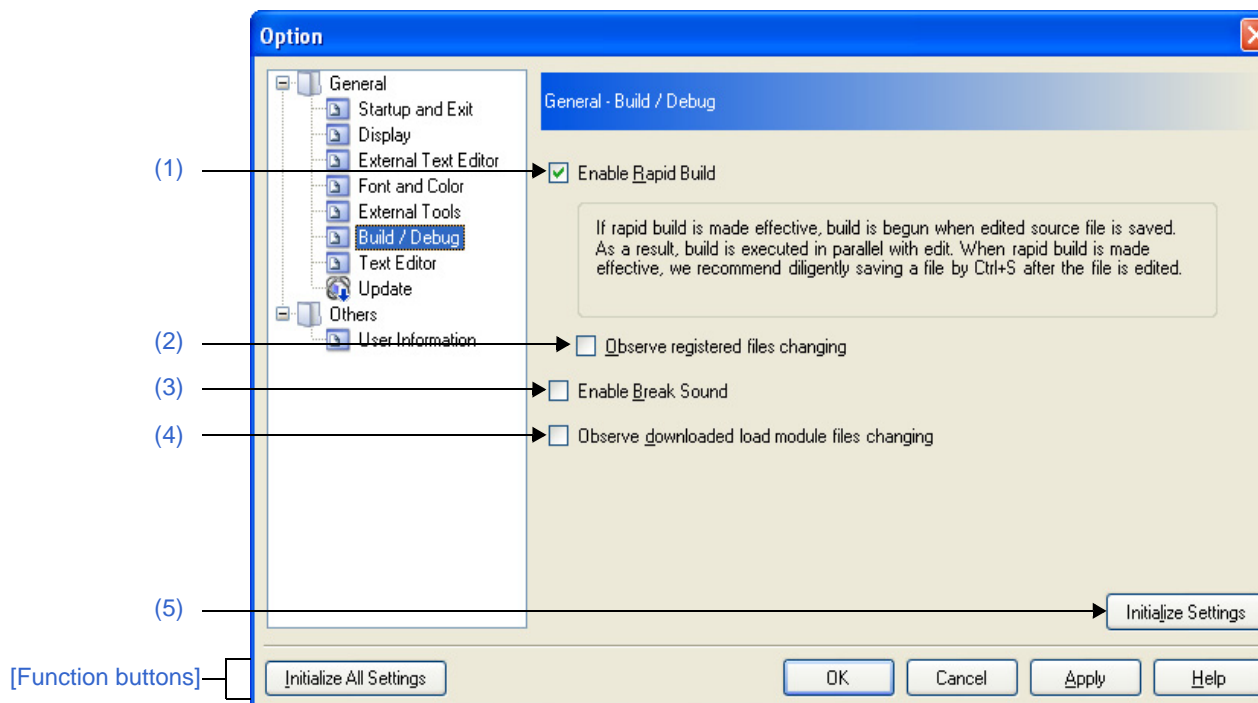
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[General - Build/Debug] category

Use this category to configure general setting relating to building and debugging.

Figure A.37 Option Dialog Box ([General - Build/Debug] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Options...].

[Description of each area]

- (1) [Enable Rapid Build]

<input checked="" type="checkbox"/>	Enable the rapid build ^{Note} feature (default).
<input type="checkbox"/>	Do not use the rapid build feature.

Note This feature automatically begins a build when the source file being edited is saved. Enabling this feature makes it possible to perform builds while editing source files. If this feature is used, we recommend saving frequently after editing source files.

- (2) [Observe registered files changing]

This item is only enabled if the [Enable Rapid Build] check box is selected.

<input checked="" type="checkbox"/>	Start a rapid build when a source file registered in the project is edited or saved by an external text editor or the like.
<input type="checkbox"/>	Do not start a rapid build when a source file registered in the project is edited or saved by an external text editor or the like (default).

Remark This item is only enabled if the [Enable Rapid Build] check box is selected.

Cautions 1. The rapid build will not finish if this item is selected, and the files to be built have been registered for automatic editing or overwriting (e.g. by commands executed before or after the build).
If the rapid build does not finish, unselect this item, and stop the rapid build.

Cautions 2. If this item is selected, a file that is registered in the project but does not exist (a file grayed out) will not be observed even if it is registered again by the Explorer etc.
To observe the file, reload the project file, or select this item again after unselecting this item and closing this dialog box.

(3) [Enable Break Sound]

<input checked="" type="checkbox"/>	Beep when the execution of a user program is halted due to a break event (hardware or software break).
<input type="checkbox"/>	Do not beep when the execution of a user program is halted due to a break event (hardware or software break) (default).

(4) [Observe downloaded load module files changing]

<input checked="" type="checkbox"/>	Observes the load module file downloaded to the debug tool for changes. When there is a change, a message dialog box confirming whether to execute the download will be displayed.
<input type="checkbox"/>	Does not observe the load module file downloaded to the debug tool for changes (default).

(5) Buttons

Initialize Settings	Return all currently displayed setting to their default values.
---------------------	---

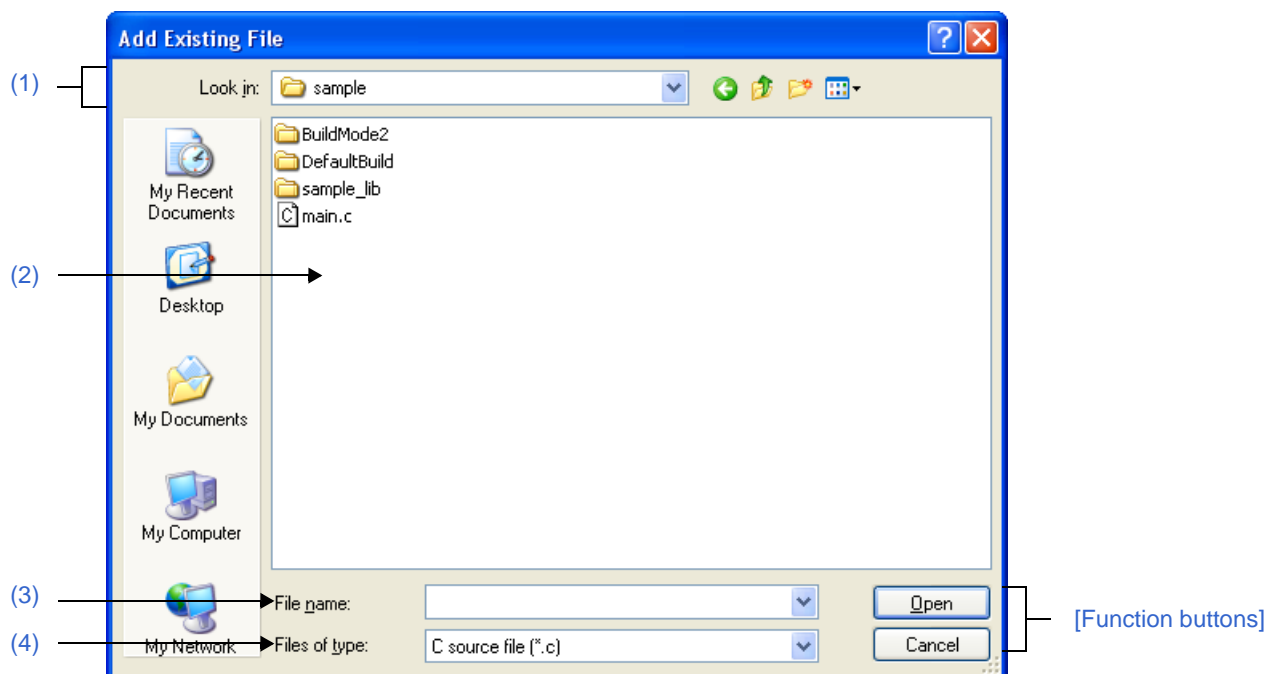
[Function buttons]

Button	Function
Initialize All Settings	Restore all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Apply all setting and closes this dialog box.
Cancel	Ignore the setting and closes this dialog box.
Apply	Apply all setting (does not close this dialog box).
Help	Display the help of this dialog box.

Add Existing File dialog box

This dialog box is used to select existing files to add to projects.

Figure A.38 Add Existing File Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- From the [File] menu, select [Add] >> [Add File...].
- On the [Project Tree panel](#), select either one of the Project node, Subproject node, File node, or file, and then select [Add] >> [Add File...] from the context menu.

[Description of each area]

- (1) [Look in] area
Select the folder that the file to add to projects exists.
The project folder is selected by default.
- (2) File list area
File list that matches to the selections in [Look in] and [Files of type] is shown.
- (3) [File name] area
Designate the file name of the file to add to projects.
- (4) [Files of type] area
Designate the file type of the file to add to projects.

C source file (*.c)	C source file
C++ source file (*.cpp; *.cp; *.cc)	C++ source file

Header file(*.h; *.hpp; *.inc)	Header file
Assembler source file(*.src, *.s)	Assembler source file
Jump table file(*.jmp)	Jump table file
Symbol address file(*.fsy)	Symbol address file
Library file(*.lib)	Library file
Object module file(*.obj)	Object module file
Relocatable file(*.rel)	Relocatable file
Text file(*.txt)	Text format
All Files(*.*)	All the format (default)

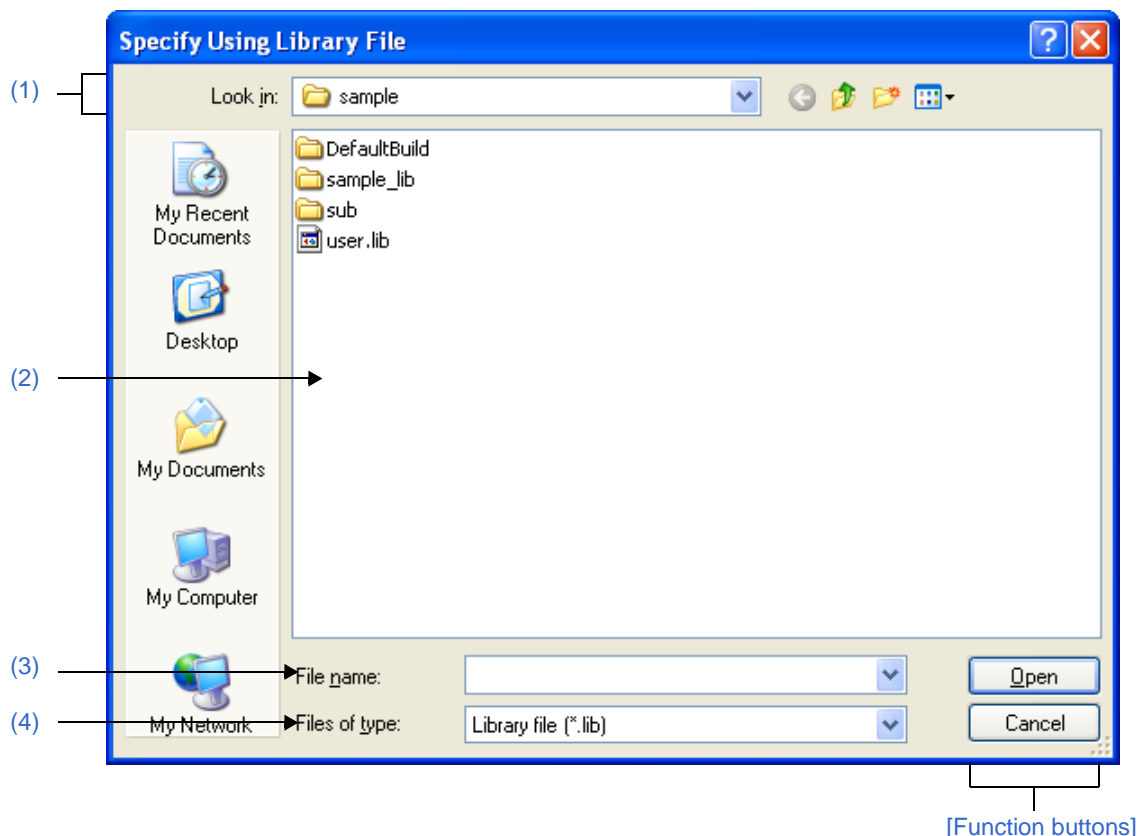
[Function buttons]

Button	Function
Open	Adds the designated file to a project.
Cancel	Closes this dialog box.

Specify Using Library File dialog box

This dialog box is used to select the library file and set it to the area that this dialog box is called from.

Figure A.39 Specify Using Library File Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), after selecting the following properties, open the [Path Edit dialog box](#) by clicking the [...] button.
 - From the [\[Common Options\] tab](#), [Using libraries] in the [Frequently Used Options(for Link)] category
 - From the [\[Link Options\] tab](#), [Using libraries] in the [Library] category
 - From the [\[Librarian Options\] tab](#), [Using libraries] in the [Library] category
- And then click the [Browse...] button in the dialog box.

[Description of each area]

- (1) [Look in] area
Select the folder where the file to be set to the area that this dialog box is called from exists.
The project folder is selected by default.
- (2) File list area
File list that matches to the selections in the [Look in] area and [File of type] area is shown.

- (3) [File name] area
Select the name of the file to be set to the area that this dialog box is called from.
- (4) [Files of type] area
Select the type of the file to be set to the area that this dialog box is called from.

Library file (*.lib)	Library file
----------------------	--------------

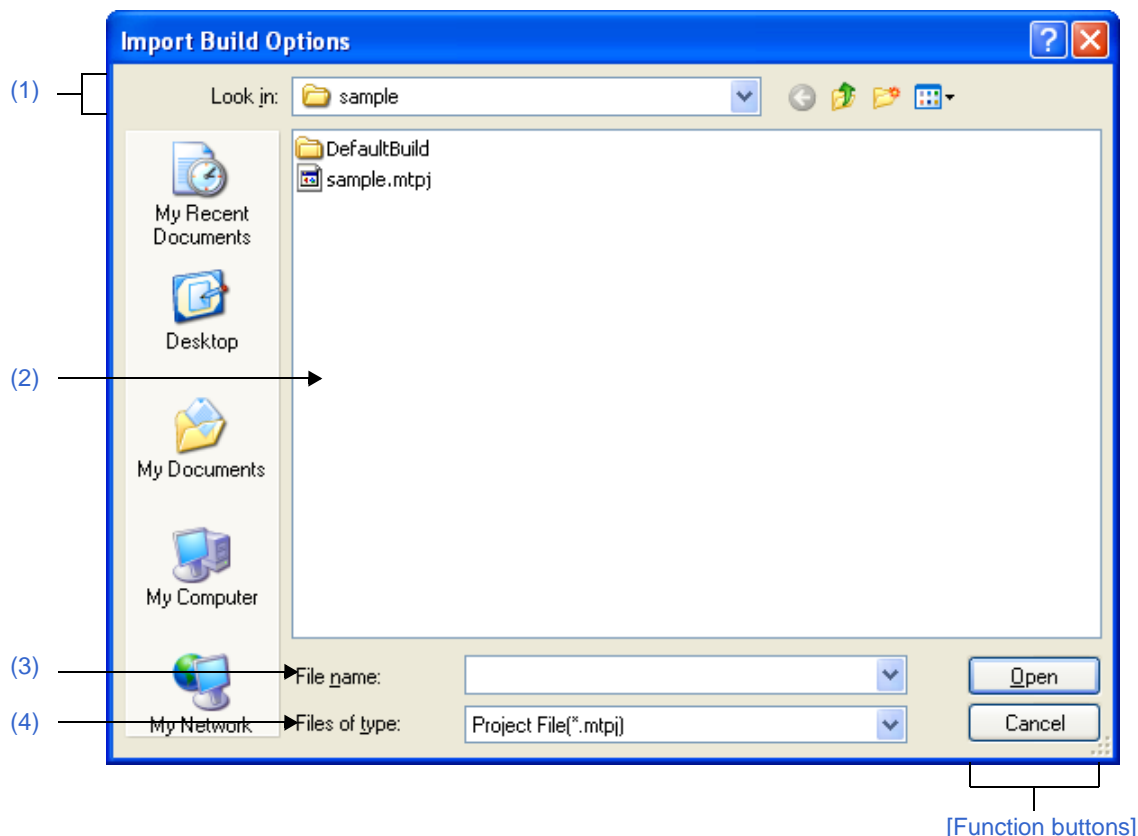
[Function buttons]

Button	Function
Open	Sets the designated file to the area that this dialog box is called from.
Cancel	Closes this dialog box.

Import Build Options dialog box

This dialog box is used to select the target project file for import the build options.

Figure A.40 Import Build Options Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Project Tree panel](#), select the Build tool node, and then select [Import Build Options...] from the context menu.

[Description of each area]

- (1) [Look in] area
Select the folder that the target project file for import the build options exists.
The current project folder is selected by default.
- (2) File list area
This area displays the list of the files which match to the selections in the [Look in] and [Files of type].
- (3) [File name] area
Specify the name of the project file.
- (4) [Files of type] area
Select the type of the project file.

Project file(*.mtpj)	Project file
----------------------	--------------

Subproject file(*.mtsp)	Subproject file
-------------------------	-----------------

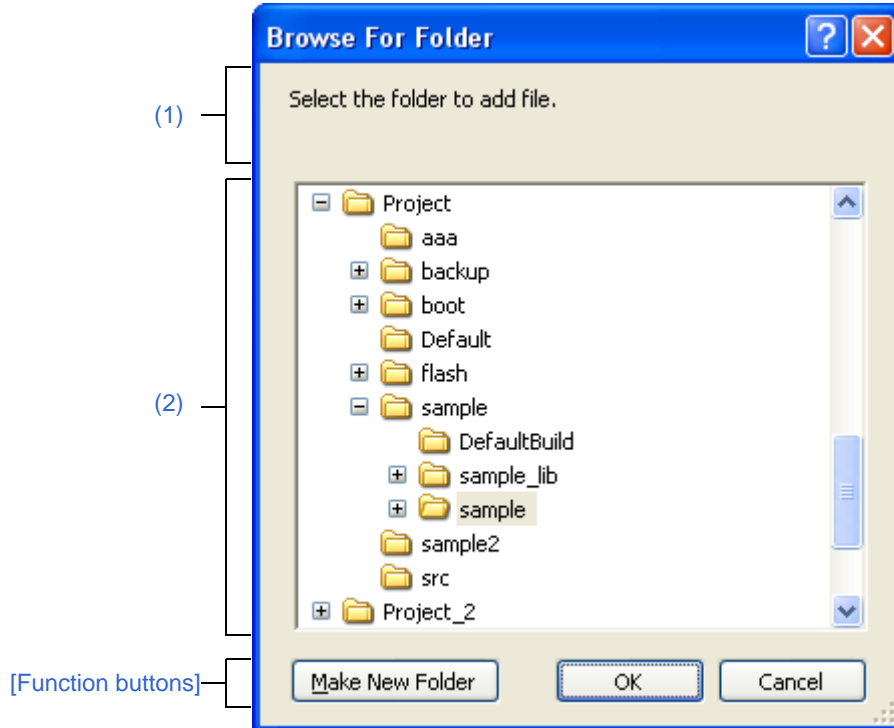
[Function buttons]

Button	Function
Open	Imports the build options of the specified project file to the current project.
Cancel	Closes this dialog box.

Browse For Folder dialog box

This dialog box is used to select a folder and retrieve it for the caller.

Figure A.41 Browse For Folder dialog box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- In the [Add File dialog box](#), click the [...] button in the [File location] area.
- In [Path Edit dialog box](#), click [...] button in the path edit area.
- On the [Property panel](#), select the following properties, and then click the [...] button.
 - From the [\[Compile Options\] tab](#), [Path of the output folder] in the [Object] category.
 - From the [\[Assemble Options\] tab](#), [Path of the output folder] in the [Object] category.
 - From the [\[Link Options\] tab](#), [Path of the output folder] in the [Object] category and [Path of the conversion file output folder] in the [Convert Load Module File] category.
 - From the [\[Librarian Options\] tab](#), [Path of the output folder] in the [Object] category.
 - From the [\[Library Generate Options\] tab](#), [Path of the output folder] in the [Object] category.
 - From the [\[Individual Compile Options\(C\)\] tab](#), [Path of the output folder] in the [Object] category.
 - From the [\[Individual Compile Options\(C++\)\] tab](#), [Path of the output folder] in the [Object] category.
 - From the [\[Individual Assemble Options\] tab](#), [Path of the output folder] in the [Object] category.

[Description of each area]

- (1) Message area

Show messages related to folders selected in this dialog box.

(2) Folder location area

Select a folder to set in the caller of the dialog box.

By default, the folder set in the caller is selected.

Remark When the area is blank or the path which does not exist is entered, "C:\Documents and Settings*user name*\My Documents" is selected instead.

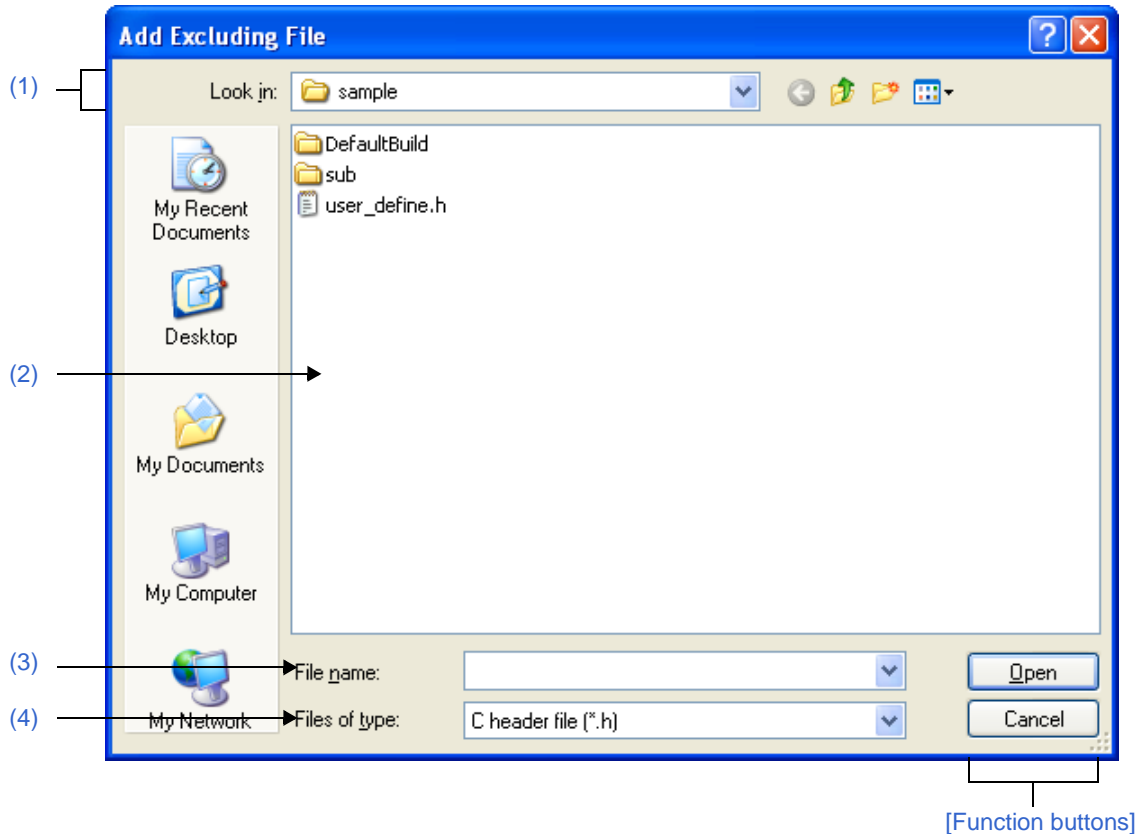
[Function buttons]

Button	Function
Make New Folder	Creates a new folder in the root of the selected folder. The default folder name is "New Folder".
OK	The designated folder path is set to the area that this dialog box is called from.
Cancel	Closes this dialog box.

Add Excluding File dialog box

This dialog box is used to select the file that will not be checked against the MISRA-C:2004 rules and set it to the area that this dialog box is called from.

Figure A.42 Add Excluding File Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), after selecting the following properties, open the [Path Edit dialog box](#) by clicking the [...] button.
 - From the [\[Compile Options\] tab](#), [Rule check exclusion file] in the [MISRA-C:2004 Rule Check] category
 - From the [\[Individual Compile Options\(C\)\] tab](#), [Rule check exclusion file] in the [MISRA-C:2004 Rule Check] category
- And then click the [Browse...] button in the dialog box.

[Description of each area]

- (1) [\[Look in\] area](#)
Select the folder where the file to be set to the area that this dialog box is called from exists.
The project folder is selected by default.
- (2) [File list area](#)
File list that matches to the selections in the [Look in] area and [File of type] area is shown.

- (3) [File name] area
Select the name of the file to be set to the area that this dialog box is called from.
- (4) [Files of type] area
Select the type of the file to be set to the area that this dialog box is called from.

C header file (*.h)	C header file (default)
C source file (*.c)	C source file
All files (*.*)	All the formats

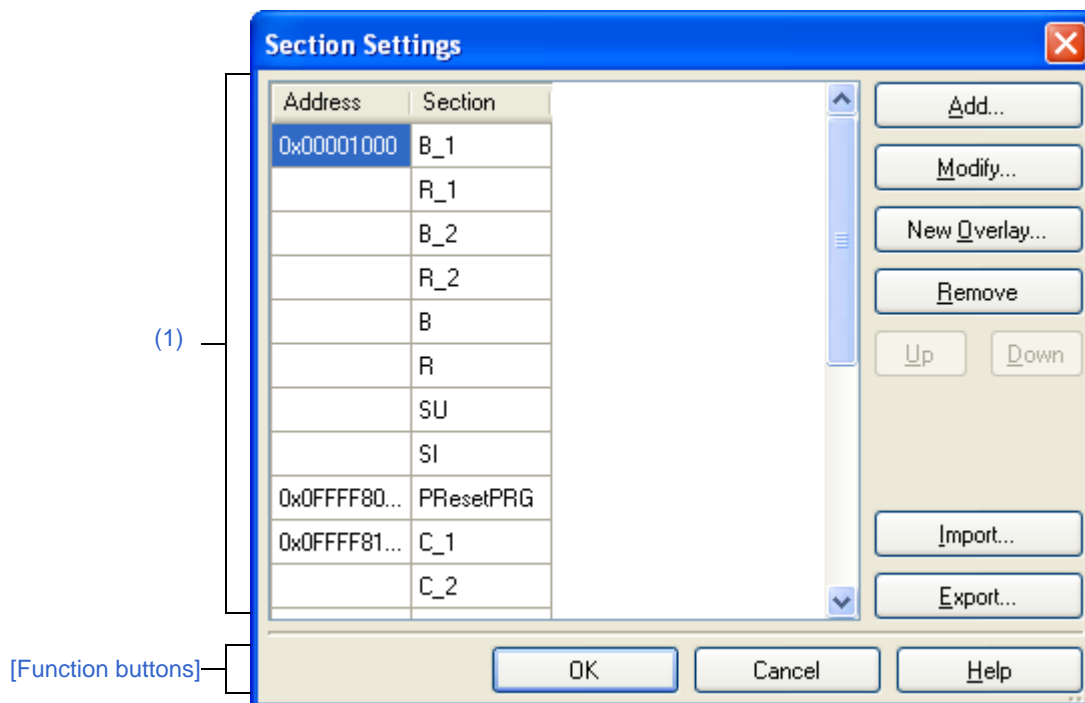
[Function buttons]

Button	Function
Open	Sets the designated file to the area that this dialog box is called from.
Cancel	Closes this dialog box.

Section Settings dialog box

This dialog box is used to add, change, or delete sections.

Figure A.43 Section Settings Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), select the following properties, and then click the [...] button.
 - From the [\[Common Options\] tab](#), [Section start address] in the [Frequently Used Options(for Link)] category.
 - From the [\[Link Options\] tab](#), [Section start address] in the [Section] category.

[Description of each area]

- (1) Address - Section area
This area displays a list of the currently configured sections.
 - (a) [Address]
This area displays the start addresses of the sections.
 - (b) [Section]
This area displays the names of the sections.
 - (c) [Overlay*n*]
This area displays the names of the sections to be overlaid. "*n*" is a number starting with "1".

(d) Button

Add...	<ul style="list-style-type: none"> - Adding a section group (address) Opens the Section Address dialog box when an address is selected in the Address - Section area. Inserts a blank row at the appropriate position among other addresses. - Adding a section Opens the Add Section dialog box when a section is selected in the Address - Section area. If there is also no blank cell below (within the section group that the selected cell belongs to), a new section is added to the bottom of the section group. The input section name is placed in a blank cell (if any) below the currently selected section.
Modify...	<ul style="list-style-type: none"> - Changing the section group (address) Opens the Section Address dialog box when an address is selected in the Address - Section area. Moves the entire group (the address and all sections that belong to the group) to the appropriate position among other addresses. - Changing the section name Opens the Modify Section dialog box when a section is selected in the Address - Section area. Replaces the currently selected section name with the one entered in the Modify Section dialog box.
New Overlay...	<ul style="list-style-type: none"> - Adding an overlay section Opens the Add Overlay dialog box. Adds the [Overlayn] column ("n" is a number starting with "1") to the Address - Section area. Places a new section in a row in the [Overlayn] column that corresponds to the section group (address) currently selected in the Address - Section area.
Remove	<ul style="list-style-type: none"> - Deleting the section group (address) Opens the Unassigned Section dialog box when an address is selected in the Address - Section area. Deletes the currently selected section from the Address - Section area. When all of the sections that belong to the section group are selected, the entire section group is deleted. - Deleting the section Deletes the currently selected section from the Address - Section area. If no sections are left in the section group after deletion, the entire section group is deleted. If no sections are left in the [Overlayn] column after deletion, the column itself is deleted.
Up	Moves up the selected section.
Down	Moves down the selected section.
Import...	Opens the Open dialog box where section-setting files can be loaded. Clicking on [OK] in the Open dialog box loads the specified file and shows its data in the Address - Section area.
Export...	Opens the Save As dialog box where section settings can be saved in files. Clicking on [OK] in the Save As dialog box stores the settings on the Address - Section area in a specified file.

[Function buttons]

Button	Function
OK	Reflects the set section to the property that called this dialog box then closes the dialog box.
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

Add Section, Modify Section, and Add Overlay dialog boxes

These dialog boxes are used to enter a new section name when adding, modifying, or overlaying a section, respectively.

Figure A.44 Add Section Dialog Box

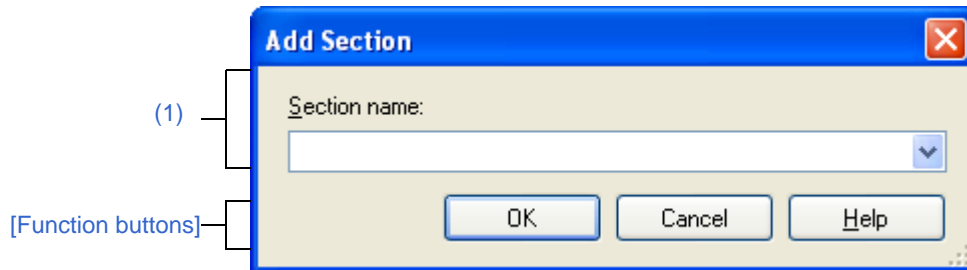


Figure A.45 Modify Section Dialog Box

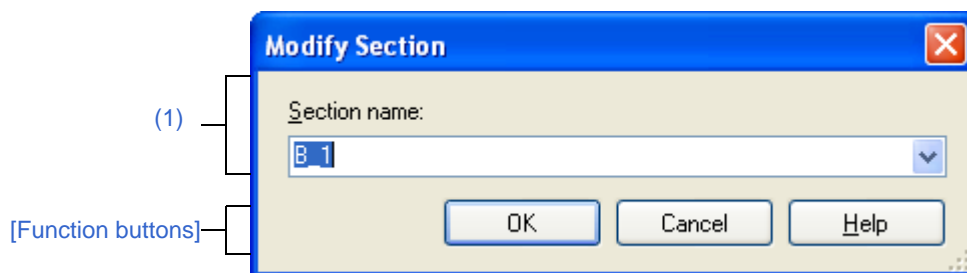
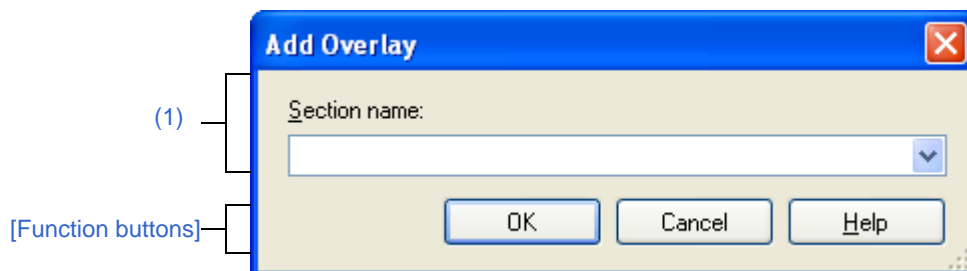


Figure A.46 Add Overlay Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- Add Section Dialog Box
 - On the [Section Settings dialog box](#), select a section in the Address - Section area, and then click the [Add] button.
- Modify Section Dialog Box
 - On the [Section Settings dialog box](#), select a section in the Address - Section area, and then click the [Modify] button.
- New Overlay Dialog Box
 - On the [Section Settings dialog box](#), select an address in the Address - Section area, and then click the [New Overlay] button.

[Description of each area]**(1) [Section name]**

Enter a section name in this area.

You can directly enter the characters into the text box or select from the drop down list.

Only ASCII characters A to Z, a to z, 0 to 9, _, and \$ can be used as section names. Use only the following characters in the field except that 0 to 9 cannot be the first letter of a section name.

The following reserved sections will be preset in the drop-down list by the linker.

\$ABS16B, \$ABS16C, \$ABS16D, \$ABS8B, \$ABS8C, \$ABS8D, \$INDIRECT, B, B\$1, B\$2, B\$4, C, C\$1, C\$2, C\$4, C\$BSEC, C\$DSEC, C\$INIT, C\$VTBL, D, D\$1, D\$2, D\$4, L, PS

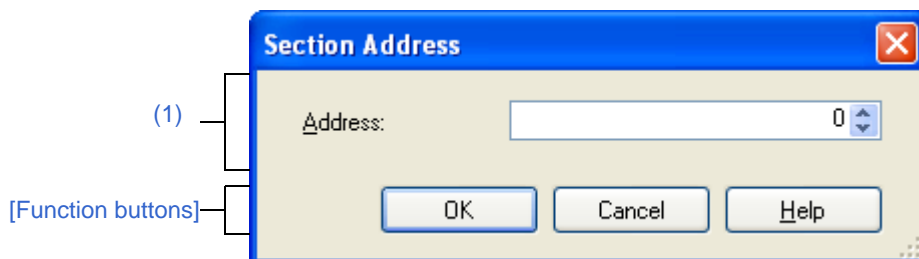
[Function buttons]

Button	Function
OK	<ul style="list-style-type: none"> - Add Section Dialog Box Closes this dialog box and adds a new section to the bottom of the section group (address) in the Address - Section area in the Section Settings dialog box dialog box. - Modify Section Dialog Box Closes this dialog box and replaces the currently selected section name in the Address - Section area in the Section Settings dialog box dialog box with the one entered. - New Overlay Dialog Box Closes this dialog box and adds the [Overlayn] column ("n" is a number starting with "1") to the Address - Section area in the Section Settings dialog box dialog box. Places a new section in a row in the [Overlayn] column that corresponds to the section group (address) currently selected in the Address - Section area.
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

Section Address dialog box

This dialog box is used to enter an address when adding or modifying a section.

Figure A.47 Section Address Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Section Settings dialog box](#), select an address in the Address - Section area, and then click the [Add] or [Modify] button.

[Description of each area]

- (1) [Address]
Enter the first address of a section in this area.
The range that can be specified for the value is 0 to FFFFFFFF (hexadecimal number). "0" is set by default.

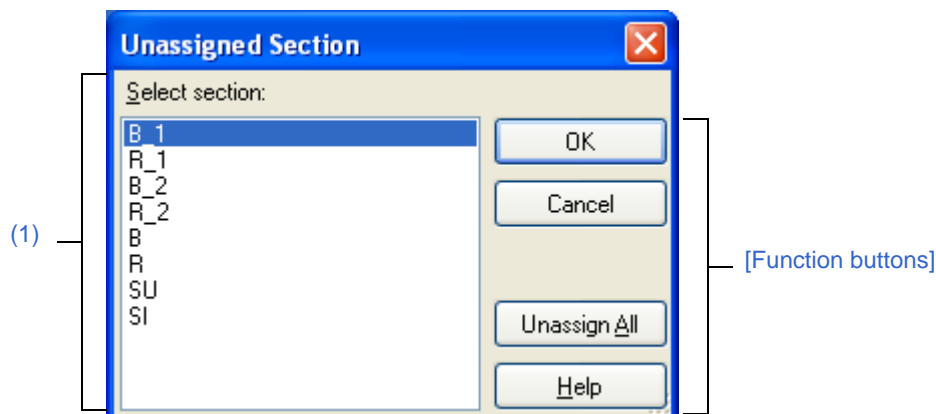
[Function buttons]

Button	Function
OK	<p>Closes this dialog box and sets the input address to the Address - Section area in the Section Settings dialog box dialog box.</p> <ul style="list-style-type: none"> - Adding a section group (address) Inserts a blank row at the appropriate position among other addresses. - Changing the section group (address) Moves the entire group (the address and all sections that belong to the group) to the appropriate position among other addresses.
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

Unassigned Section dialog box

This dialog box is used to select sections to be deleted.

Figure A.48 Unassigned Section Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Section Settings dialog box](#), select an address in the Address - Section area, and then click the [Remove] button.

[Description of each area]

- (1) [Select sections] area
 Select sections to be deleted in this area.
 You can select multiple section types by left clicking while holding down the [Ctrl] or [Shift] key.

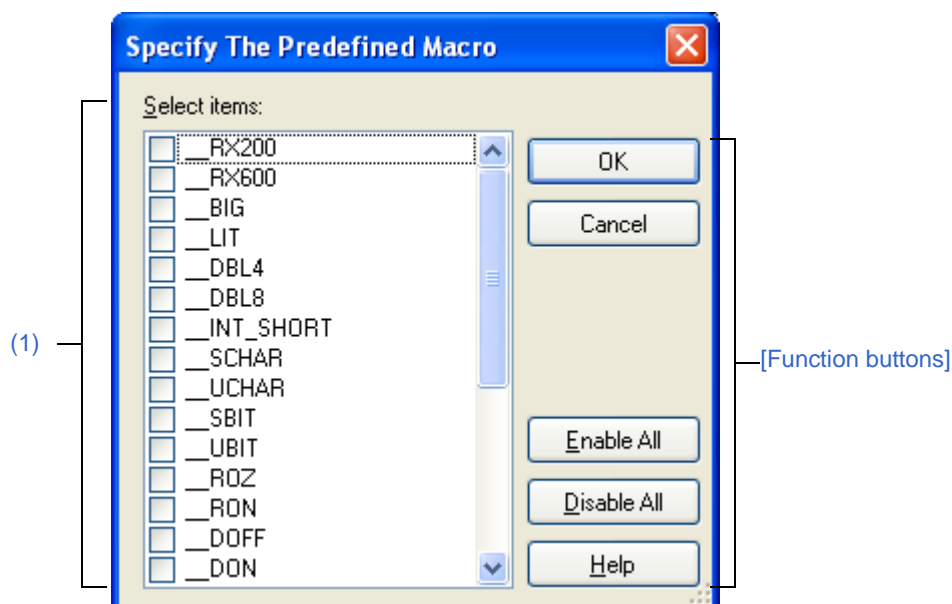
[Function buttons]

Button	Function
OK	Deletes the section selected in the [Select section] area from the Address - Section area in the Section Settings dialog box dialog box and closes the dialog box. If no sections are left in the section group (address) after deletion, the entire section group is deleted. If no sections are left in the [Overlay n] column after deletion, the column itself is deleted.
Cancel	Cancels the file selecting and closes the dialog box.
Unassigned All	Deletes all sections in the [Select section] area from the Address - Section area in the Section Settings dialog box dialog box and closes the dialog box. The entire section group (address) is deleted.
Help	Displays the help of this dialog box.

Specify The Predefined Macro dialog box

This dialog box is used to select the predefined macros to disable and set it to the area that this dialog box is called from.

Figure A.49 Specify The Predefined Macro Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), select the following properties, and then click the [...] button.
 - From the [\[Compile Options\] tab](#), [Invalidates the predefined macro] in the [Source] category.
 - From the [\[Individual Compile Options\(C\)\] tab](#), [Invalidates the predefined macro] in the [Source] category.
 - From the [\[Individual Compile Options\(C++\)\] tab](#), [Invalidates the predefined macro] in the [Source] category.

[Description of each area]

(1) [Select items]

The list of the predefined macros which can be disabled for the area that this dialog box is called from is displayed. Select the predefined macros to disable to set to the area that opened this dialog box, via check boxes.

Remark In the area that opened this dialog box, if a predefined macros to disable is already set, the check box for that macro will be selected by default.

[Function buttons]

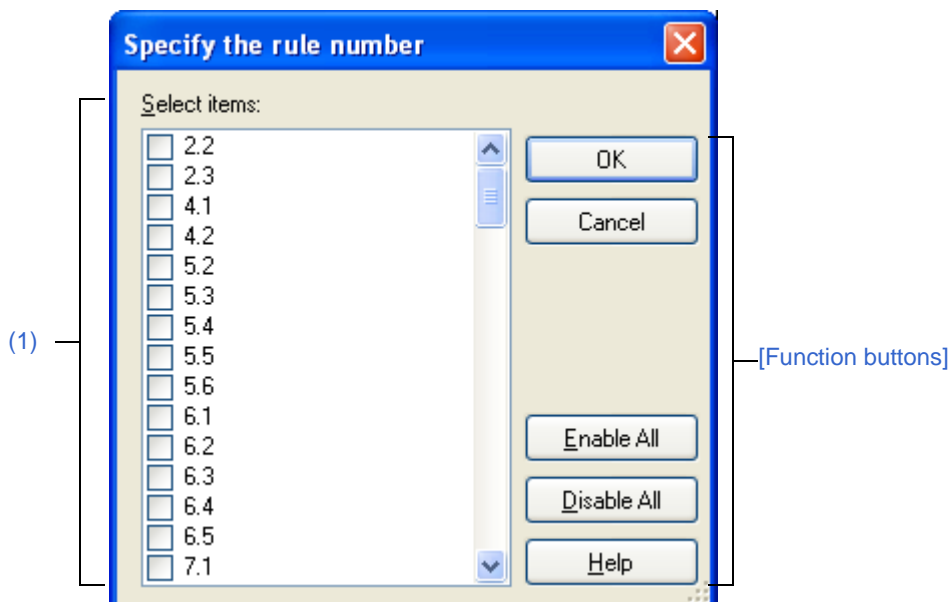
Button	Function
OK	Closes this dialog box and specifies the selected macros to the area that opened this dialog box.
Cancel	Cancels the macros selecting and closes the dialog box.

Button	Function
Enable All	Select all the macros in [Select items].
Disable All	Deselect all the macros in [Select items].
Help	Displays the help of this dialog box.

Specify the rule number dialog box

This dialog box is used to select numbers of the MISRA-C: 2004 rules and set it to the area that this dialog box is called from.

Figure A.50 Specify the rule number Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), select the following properties, and then click the [...] button.
 - From the [\[Compile Options\] tab](#), [Rule number], [Exclusion rule number], [Check rule number besides required rule], [Exclusion rule number from required rule] in the [MISRA C rule check] category.
 - From the [\[Individual Compile Options\(C\)\] tab](#), [Rule number], [Exclusion rule number], [Check rule number besides required rule], [Exclusion rule number from required rule] in the [MISRA C rule check] category.

[Description of each area]

(1) [Select items]

The list of numbers of the MISRA-C: 2004 rules which can be set for the area that this dialog box is called from is displayed (ascending order).

Select numbers to set to the area that opened this dialog box, via check boxes.

Remark In the area that opened this dialog box, if a rule number is already set, the check box for that number will be selected by default.

[Function buttons]

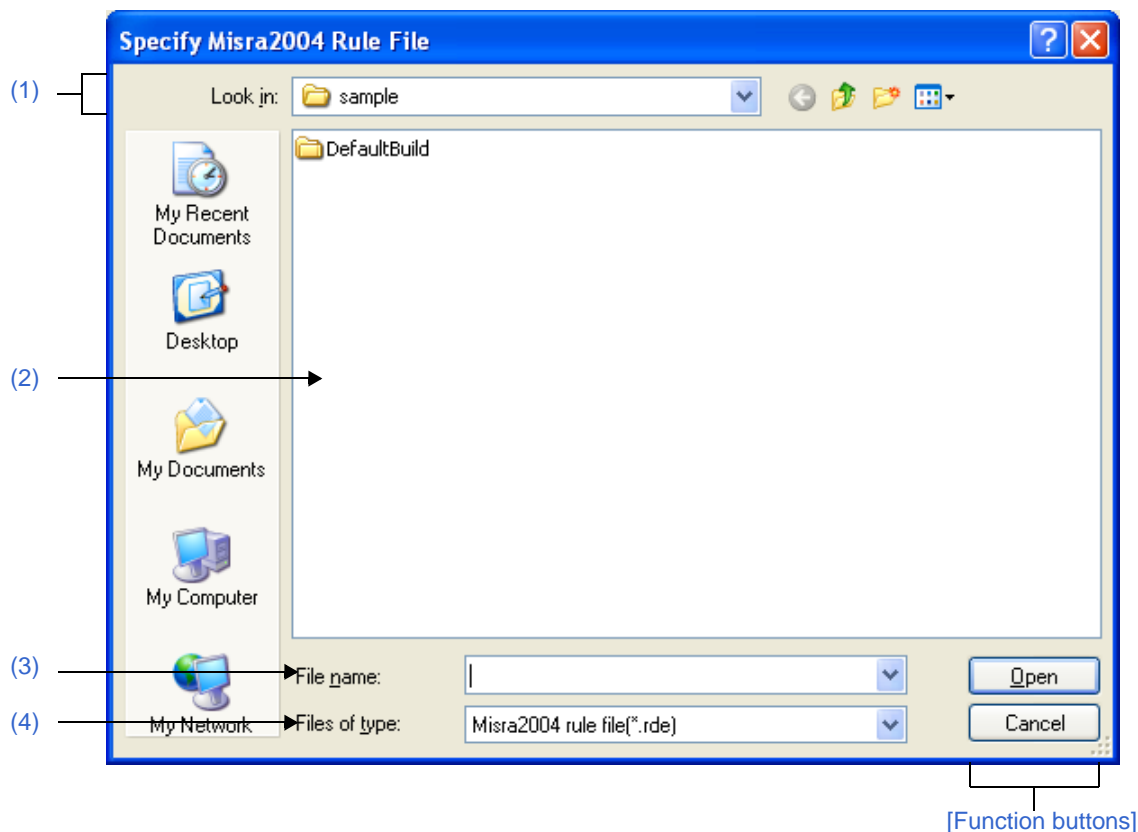
Button	Function
OK	Closes this dialog box and specifies the selected numbers to the area that opened this dialog box.

Button	Function
Cancel	Cancels the number selecting and closes the dialog box.
Enable All	Select all the rule numbers in [Select items].
Disable All	Deselect all the rule numbers in [Select items].
Help	Displays the help of this dialog box.

Specify Misra2004 Rule File dialog box

This dialog box is used to select the MISRA-C: 2004 rule file and set it to the area that this dialog box is called from.

Figure A.51 Specify Misra2004 Rule File Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- On the [Property panel](#), select the following property, and then click the [...] button.
 - From the [\[Compile Options\] tab](#), [Rule number description file] in the [MISRA C rule check] category.
 - From the [\[Individual Compile Options\(C\)\] tab](#), [Rule number description file] in the [MISRA C rule check] category.

[Description of each area]

- (1) [Look in] area
Select the folder where the file to be set to the area that this dialog box is called from exists.
The project folder is selected by default.
- (2) File list area
This area displays the list of the files which match to the selections in the [Look in] and [Files of type].
- (3) [File name] area
Select the name of the file to be set to the area that this dialog box is called from.

(4) [Files of type] area

Select the type of the file to be set to the area that this dialog box is called from.

Misra2004 rule file(*.rde)	Misra2004 rule file (default)
All files(*.*)	All the formats

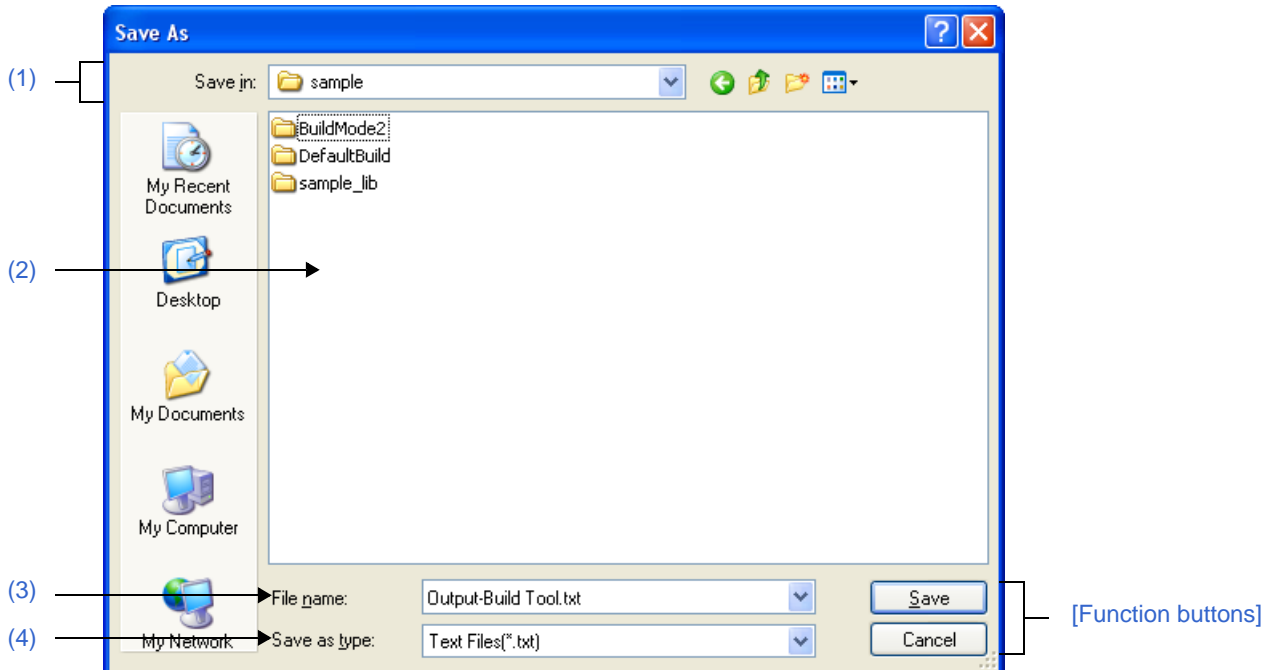
[Function buttons]

Button	Function
Open	Sets the designated file to the area that this dialog box is called from.
Cancel	Closes this dialog box.

Save As dialog box

This dialog box is used to save the editing file or contents of each panel to a file with a name.

Figure A.52 Save As Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- Focus the [Editor panel](#), and then select [Save *file name* As...] from the [File] menu.
- Focus the [Output panel](#), and then select [Save *tab name* As...] from the [File] menu.

[Description of each area]

- (1) [Save in] area
Select the folder to save the panel contents in the file.
The following folders are selected by default.
 - (a) In the [Editor panel](#)
The folder that currently editing file is saved.
 - (b) In the [Output panel](#)
The project folder is selected when the file is save for the first time. The previously selected file is selected after the second time.
- (2) File list area
File list that matches the selections in the [Save in] area and [Save as type] area is shown.
- (3) [File name] area
Specify the file name to save.

(4) [Save as type] area

(a) In the [Editor panel](#)

The following file types are displayed depend on the file type of the currently editing file.

Text file(*.txt)	Text format
C source file(*.c)	C language source file
C++ source file (*.cpp; *.cp; *.cc)	C++ language source file
Header file(*.h; *.hpp; *.inc)	Header file
Assemble file(*.src; *.s)	Assembler source file
Jump table file(*.jmp)	Jump table file
Symbol address file(*.fsy)	Symbol address file
Link order specification file(*.mtls)	Link order specification file
Map file(*.map; *.lbp)	Map file
Hex file(*.hex)	Hex file
S record file(*.mot)	S record file

(b) In the [Output panel](#)

The following file types are displayed.

Text file(*.txt)	Text format
------------------	-------------

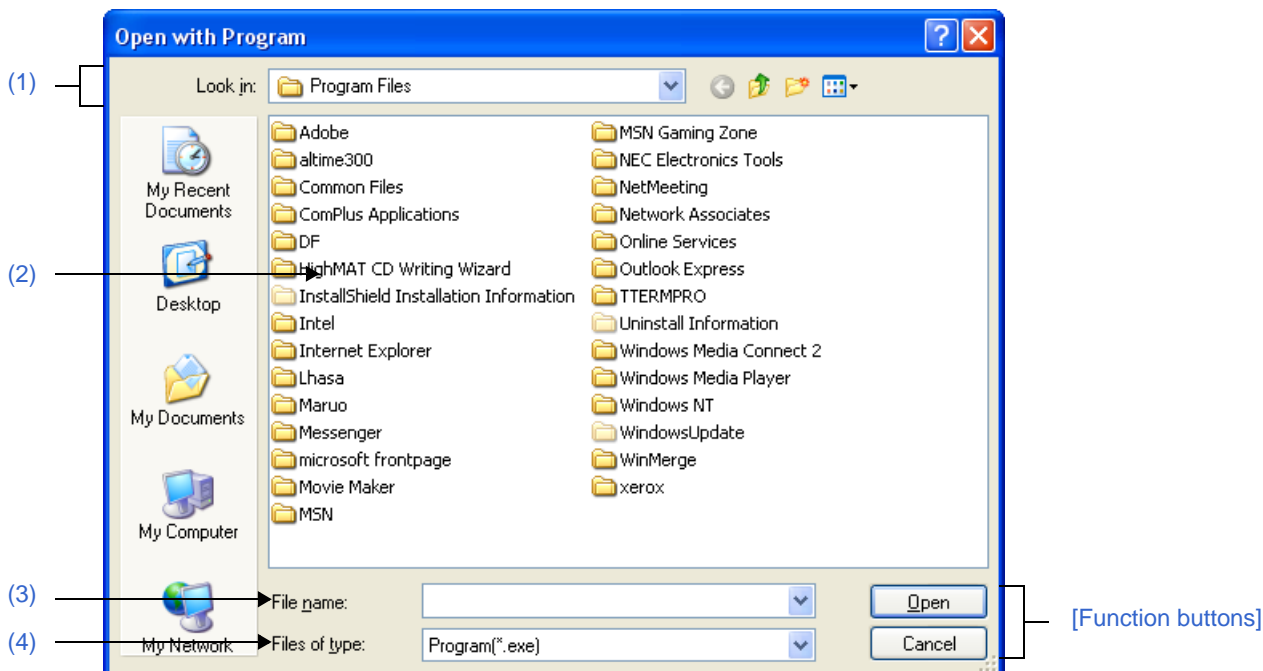
[Function buttons]

Button	Function
Save	Saves the file as the designated file name.
Cancel	Closes this dialog box.

Open with Program dialog box

This dialog box is used to select the application to open the file selected in Project Tree.

Figure A.53 Open with Program Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Project Tree panel](#), select a file, and then select [Open with Selected Application...] from the context menu.

[Description of each area]

- (1) [Look in] area
Select the folder where the application to open the file is stored.
Program folder (for Windows XP, "C:\Program Files") is selected by default.
- (2) File list area
File list that matches to the selections in the [Look in] area and [File of type] area is shown.
- (3) [File name] area
Specify the executable file name of the application to open the file.
- (4) [Files of type] area
Specify the executable file type of the application to open the file.

Program (*.exe)	Executable format (default)
All Files (*.*)	All the formats

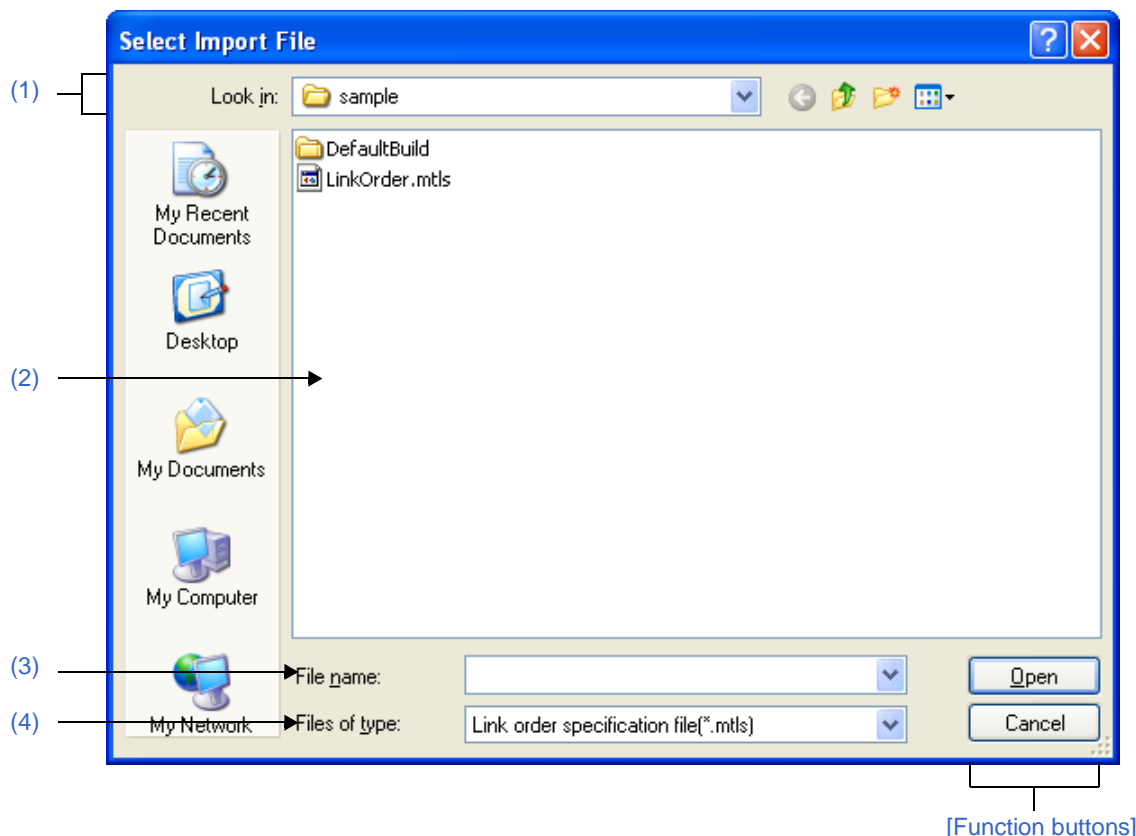
[Function buttons]

Button	Function
Open	Opens the file with the specified application.
Cancel	Closes this dialog box.

Select Import File dialog box

This dialog box is used to select a link order specification file to import to the [Link Order dialog box](#).

Figure A.54 Select Import File Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

[How to open]

- In the [Link Order dialog box](#), click the [Import] button.

[Description of each area]

- (1) [Look in] area
Select the folder that the link order specification file exists.
The project folder is selected when the file is selected for the first time. The previously selected folder is selected after the second time.
- (2) File list area
This area displays the list of the files which match to the selections in the [Look in] and [Files of type].
- (3) [File name] area
Specify the name of a link order specification file.

- (4) [Files of type] area
Select the type of the link order specification file.

Link order specification file (*.mtls)	Link order specification file
--	-------------------------------

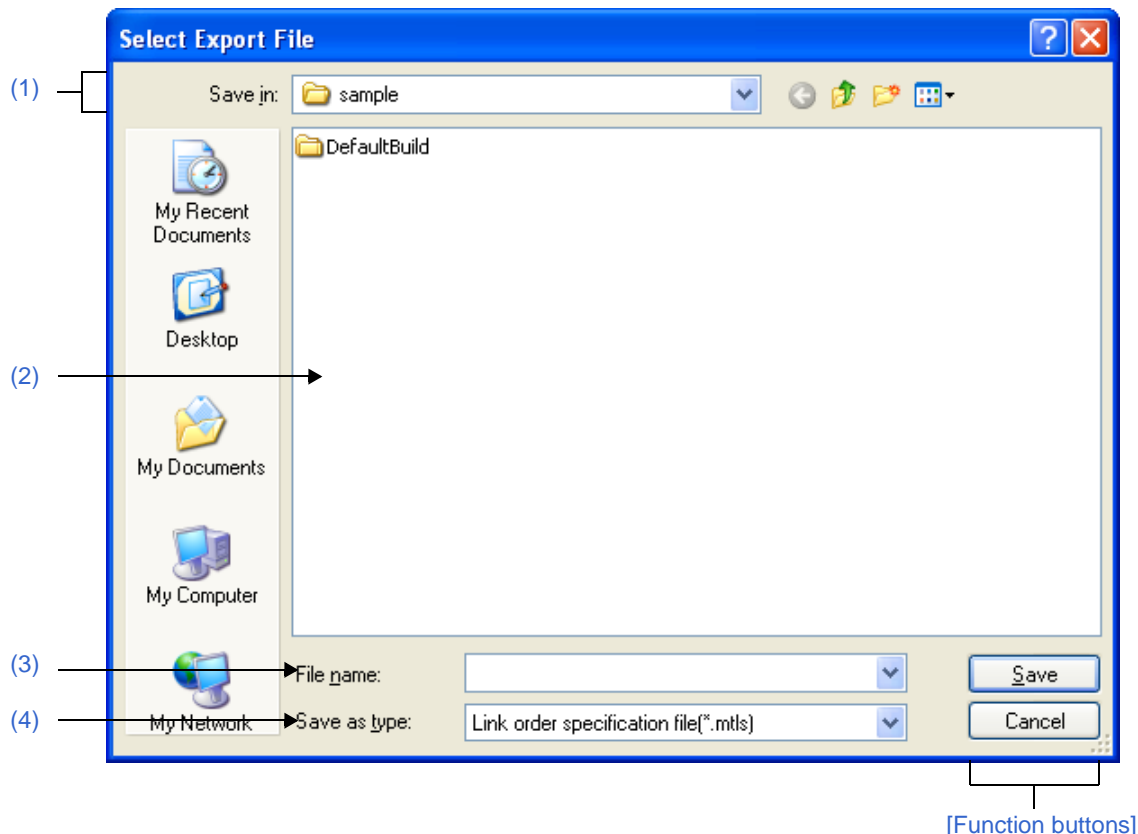
[Function buttons]

Button	Function
Open	Imports the specified file to the Link Order dialog box .
Cancel	Closes this dialog box.

Select Export File dialog box

This dialog box is used to generate a link order specification file.

Figure A.55 Select Export File Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- In the [Link Order dialog box](#), click the [Export] button.

[Description of each area]

- (1) [Save in] area
Select the folder for outputting a link order specification file.
The project folder is selected when the file is selected for the first time. The previously selected folder is selected after the second time.
- (2) File list area
This area displays the list of the files which match to the selections in the [Save in] area and [Save as type] area.
- (3) [File name] area
Specify the name of a link order specification file.

- (4) [Save as type] area
This area displays the following file type.

Link order specification file (*.mtls)	Link order specification file
--	-------------------------------

[Function buttons]

Button	Function
Save	Generates a link order specification file as the specified name.
Cancel	Closes this dialog box.

B. COMMAND REFERENCE

This appendix describes the detailed specifications of each command included in the build tool.

B.1 RX Family C/C++ Compiler

The RX family C/C++ compiler generates a file executable in the target system from the source program written in C language, C99 language, C++ language, or assembly language.

In this compiler, a single driver controls multiple phases from preprocessing to linkage.

The following describes processing in each phase.

- (1) Compiler
This processes preprocessing directives, comments, and optimization for the C source program and generates an assembly-language source file.
- (2) Preprocessor
This processes the preprocessing directives in the source program.
Only when the -P option is specified, it outputs the preprocessed file.
- (3) Parsing section
This parses the C source program and then converts it to the internal data representation for the compiler.
- (4) Optimizing section
This optimizes the internal data representation converted from the C source program.
- (5) Code generating section
This converts the internal data representation to an assembly-language source program.
- (6) Assembler
This converts the assembly-language source program to machine-language instructions and generates a relocatable object module file.
- (7) Optimizing linkage editor
This links object module files, link directive files, and library files and generates an object file (load module file) executable in the target system.

B.1.1 Input/Output Files

The following shows the files input to and output from the RX family C/C++ compiler.

Table B.1 Input/Output Files for the RX Family C/C++ Compiler

File Type	Extension	I/O	Description
C source program file	.c	Input	A source file written in C99 language. This file is created by the user.
C++ source program file	.cpp, .cp, and .cc	Input	A source file written in C++ language. This file is created by the user.
Include file	Optional	Input	A file referenced by the source file and written in C, C99, C++, or assembly language. This file is created by the user.
Preprocessor expansion file for the C program	.p	Output	A file output as a result of preprocessing applied to an input C-language or the C99-language source program. An ASCII image file. This is output when the -output=prep option is specified.
Preprocessor expansion file for the C++ program	.pp	Output	A file output as a result of preprocessing applied to an input C++-language source program. An ASCII image file. This is output when the -output=prep option is specified.
Assembly-source program file	.src	Output	An assembly-language file generated from a C, C99, or C++ source file through compilation.
	.src	Input	A source file written in assembly language.
List file for the assembly program	.lst	Output	A list file containing the assembly result information. This is output when the -listfile option is specified. The output contents are selected with the -show option.
Relocatable object program file	.obj	Output	An ELF-format file that contains the machine-language information, the relocation information about the allocation addresses of machine-language instructions, and symbol information.
Absolute load module file	.abs	Output	An ELF-format file for the object code generated as a result of linkage. This is an input file when a hex file is output.
Linkage list file	.map	Output	A list file containing the linkage result information. This is output when the -list option is specified. The output contents are selected with the -show option.
Library file	.lib	Output	A file where multiple object module files are registered.
Library list file	.lbp	Output	A list file containing the result information of generation of the library. This is output when the -list option is specified. The output contents are selected with the -show option.
Library backup file	.lbp	Output	File type for saving the contents of original library files before they are overwritten by the library generator.
Hex file (Motorola S-format file)	.mot	Output	A Motorola S-format file in the hex format converted from the load module file.
Hex file (Intel (expansion) hex format file)	.hex	Output	An Intel (expansion) file in the hex format converted from the load module file.

File Type	Extension	I/O	Description
Hex file (binary format file)	.bin	Output	A binary file in the hex format converted from the load module file.
Stack information file	.sni	Output	A stack information file. This is output when the -stack option is specified.
Debugging information file	.dbg	Output	A debugging information file. This is output when the -sdebug option is specified.
Object file including a definition specified with a file having extension td	.rti	Output	An object file including a definition specified with a file having extension td.
Calling information file	.cal	Output	A calling information file. This is output by CallWalker.
External symbol assignment information file	.bls	Output	An external symbol assignment information file. This is output at linkage when the -map option is specified.
	.bls	Input	An external symbol assignment information file. This is specified as an input file for the -map option at compilation.
Jump table file (assembly language)	.jmp	Output	An assembler source file for the jump table that branches the external definition symbol. This is output when the -jump_entries_for_pic option is specified.
Symbol address file (assembly language)	.fsy	Output	An assembler source file that describes the external definition symbol in an assembler directive. This is output when the -fsymbol option is specified.
C++ language function support file	.td, .ti, .pi, and .ii	Output	An information file that supports the C++ language function.

B.1.2 Operating Instructions

This section describes how to operate the RX family C/C++ compiler.

The commands will take options from left to right on the command-line. When two or more options with conflicted meanings are selected and it will take neither error nor warning, the right-side option will be enabled. This results are different according to each options. For more details, please confirm each options' descriptions.

(1) Operating Tools

(a) Compiler (ccrx)

ccrx is the startup command for the compile driver.

Compilation, assemble, and linkage can be performed using this command.

When the extension of an input file is ".s", ".src", ".S", or ".SRC", the compiler interprets the file as an assembly-language file (.src, .s) and initiates the assembler.

A file with an extension other than those above is compiled as a C/C++ source file (.c, .cpp).

Two or more input files can be specified at the same time. Cases where two or more C/C++ language source files are specified as input files at the same time are referred to as "batch compilation."

[Command description format]

```
ccrx [ $\Delta$ <option> ...][ $\Delta$ <file name>[  $\Delta$ <option> ...] ...]
      <option>: -<option>[= $\Delta$ <suboption>[= $\Delta$ <suboption>]][, ...]
```

(b) Assembler (asrx)

asrx is the startup command for the assembler.

[Command description format]

```
asrx [Δ<option> ...][ Δ<file name>[ Δ<option> ...] ...]
      <option>: -<option>[=<suboption>][, ...]
```

(c) Optimizing Linkage Editor (rlink)

rlink is the startup command for the optimizing linkage editor.

The optimizing linkage editor has the following functions as well as the linkage processing.

- Optimizes relocatable files at linkage
- Generates and edits library files
- Converts files into Motorola S type files, Intel hex type files, and binary files

[Command description format]

```
rlink [Δ<option> ...][ Δ<file name>[ Δ<option> ...] ...]
      <option>: -<option>[=<suboption>][, ...]
```

(d) Library Generator (lbgrx)

lbgrx is the startup command for the library generator.

[Command description format]

```
lbgrx [Δ<option> ...]
      <option>: -<option>[=<suboption>][, ...]
```

(2) Command Description Examples

(a) Compilation, Assemble, and Linkage by One Command

Perform all steps below by a single command.

- Compile C/C++ source files (tp1.c and tp2.c) in **ccrx**.
- After compilation, assemble the files in **asrx**.
- After assemble, link the files in **rlink** to generate an absolute file (tp.abs).

[Command description]

```
ccrx -isa=rxv1 -output=abs=tp.abs tp1.c tp2.c
```

- Remarks 1. When the output type specification of the **output** option is changed to **-output=sty**, the file after linkage will be generated as a Motorola S type file.
- Remarks 2. An intermediate file generated during the absolute file generation process (assembly-language file or relocatable file) is not saved. Only a file of the type specified by the **output** option is to be generated.
- Remarks 3. In order to specify assemble options and linkage options that are valid for only the assembler and optimizing linkage editor in **ccrx**, use the **-asmcmd**, **-lnkcmd**, **-asmopt**, and **-lnkopt** options.
- Remarks 4. Object files that are to be linked are allocated from address 0. The order of the sections is not guaranteed. In order to specify the allocation address or section allocation order, specify options for the optimizing linkage editor using the **-lnkcmd** and **-lnkopt** options.

(b) Compilation and Assemble by One Command

Perform all steps below by a single command, and initiate the linker with another command to generate tp.abs.

- Compile C/C++ source files (tp1.c and tp2.c) in **ccrx**.
- After compilation, assemble the files in **asrx** to generate relocatable files (tp1.obj and tp2.obj).

[Command description]

```
ccrx -isa=rxv1 -output=obj tp1.c tp2.c
rlink -form=abs -output=tp.abs -subcommand=cmd.sub tp1.obj tp2.obj
```

- Remarks 1. When the **-output=obj** option is specified in **ccrx**, **ccrx** generates relocatable files.
- Remarks 2. In order to change relocatable file names, their C/C++ source files have to be input in **ccrx**, one file each.

Remarks 3. When the **form** option in **rlink** is changed to **-form=sty**, the file after linkage will be generated as a Motorola S type file.

(c) Compilation, Assemble, and Linkage by Separate Commands

Individually perform each step below by a single command.

- Compile C/C++ source files (tp1.c and tp2.c) in **ccrx** to generate assembly-language files (tp1.src and tp2.src).
- Assemble the assembly-language files (tp1.src and tp2.src) in **asrx** to generate relocatable files (tp1.obj and tp2.obj).
- Link the relocatable files (tp1.obj and tp2.obj) in **rlink** to generate an absolute file (tp.abs).

[Command description]

```
ccrx -isa=rxv1 -output=src tp1.c tp2.c
asrx tp1.src tp2.src
rlink -form=abs -output=tp.abs -subcommand=cmd.sub tp1.obj tp2.obj
```

Remark When the **-output=src** option is specified in **ccrx**, **ccrx** generates assembly-language files.

(d) Assemble and Linkage by One Command

Perform all steps below by a single command.

- Assemble assembly-language files (tp1.src and tp2.src) in **asrx**.
- After assemble, link the files in **rlink** to generate an absolute file (tp.abs).

[Command description]

```
ccrx -isa=rxv1 -output=abs=tp.abs tp1.src tp2.src
```

Remark Object files that are to be linked are allocated from address 0. The order of the sections is not guaranteed. In order to specify the allocation address or section allocation order, specify options for the optimizing linkage editor using the **-lnkcmd** and **-lnkopt** options.

(e) Assemble and Linkage by Separate Commands

Individually perform each step below by a single command.

- Assemble assembly-language files (tp1.src and tp2.src) in **asrx** to generate relocatable files (tp1.obj and tp2.obj).
- Link the relocatable files (tp1.obj and tp2.obj) in **rlink** to generate an absolute file (tp.abs).

[Command description 1]

```
ccrx -isa=rxv1 -output=obj tp1.src tp2.src
rlink -form=abs -output=tp.abs -subcommand=cmd.sub tp1.obj tp2.obj
```

[Command description 2]

```
asrx -isa=rxv1 tp1.src tp2.src
rlink -form=abs -output=tp.abs -subcommand=cmd.sub tp1.obj tp2.obj
```

(3) Set options in CubeSuite+

This section describes how to set options for the RX family C/C++ compiler from CubeSuite+.

On the CubeSuite+'s [Project Tree panel](#), select the Build Tool node. Next, select the [View] menu >> [Property]. The [Property panel](#) opens.

(a) For application projects

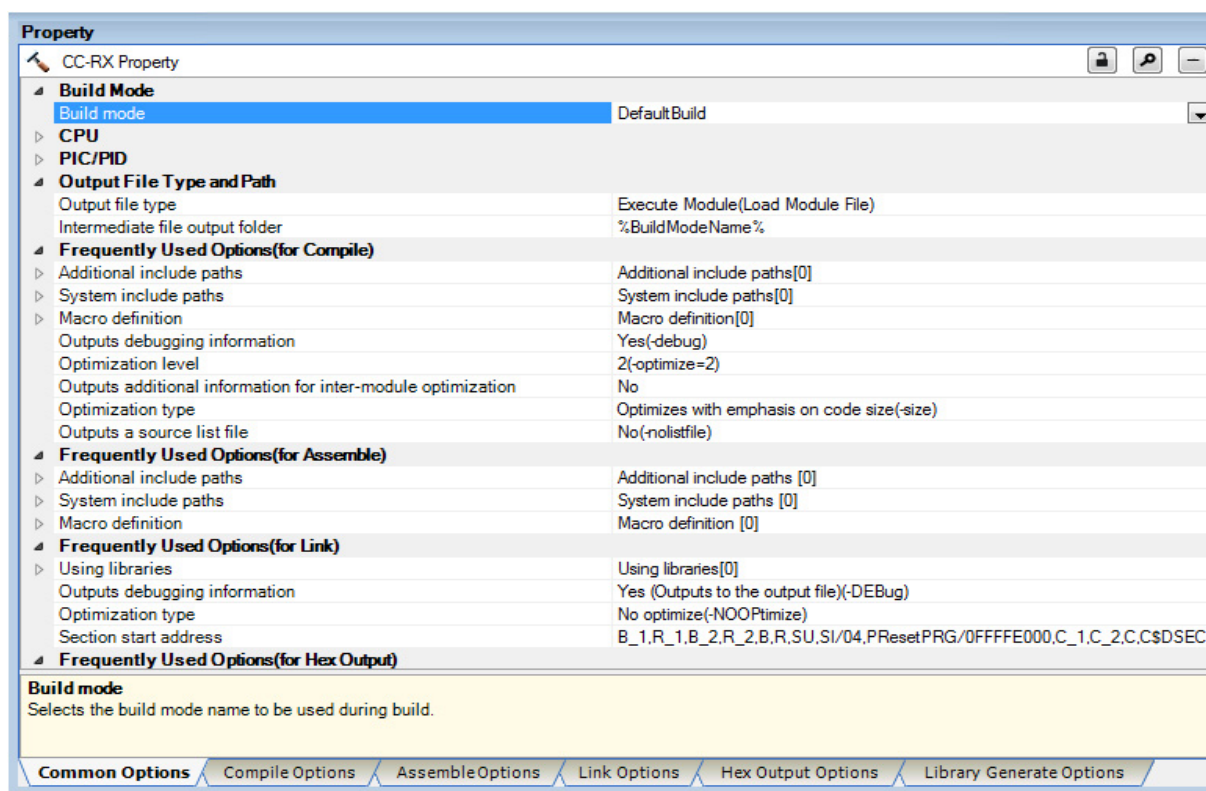
Select the [\[Common Options\] tab](#)/[\[Compile Options\] tab](#)/[\[Assemble Options\] tab](#)/[\[Link Options\] tab](#)/[\[Library Generate Options\] tab](#).

(b) For library projects

Select the [\[Common Options\] tab](#)/[\[Compile Options\] tab](#)/[\[Assemble Options\] tab](#)/[\[Librarian Options\] tab](#).

You can set the various options for the RX family C/C++ compiler by setting the necessary properties in this tab.

Figure B.1 Property Panel



- (4) Environment Variables (Command Prompt)
Environment variables are listed below.

Table B.2 Environment Variables

No.	Environment Variable	Description	Default When Specification is Omitted
1	path	Specifies a storage directory for the execution file.	Specification cannot be omitted.
2	BIN_RX	Specifies the directory in which ccrx is stored.	<ccrx storage directory> Specification cannot be omitted when the lbgrx command is used.
3	ISA_RX *1	Selects an instruction-set architecture. <Instruction-set architectures> RXV1 RXV2	No value is set when the specification is omitted.
4	INC_RX	Specifies a directory in which an include file of the compiler is stored.	<ccrx storage directory> \\.include
5	INC_RXA	Specifies a directory in which an include file of the assembler is stored.	No value is set when the specification is omitted.
6	TMP_RX	Specifies a directory in which a temporary file is generated.	%TEMP% when the ccrx command is used.

No.	Environment Variable	Description	Default When Specification is Omitted
7	HLNK_LIBRARY1 HLNK_LIBRARY2 HLNK_LIBRARY3	Specifies a default library name for the optimizing linkage editor. Libraries which are specified by a library option are linked first. Then, if there is an unresolved symbol, the default libraries are searched in the order of 1, 2, 3.	No value is set when the specification is omitted.
8	HLINK_TMP	Specifies a folder in which the optimizing linkage editor generates temporary files. If HLNK_TMP is not specified, the temporary files are created in the current folder.	No value is set when the specification is omitted.
9	HLINK_DIR	Specifies an input file storage folder for the optimizing linkage editor. The search order for files which are specified by the input or library option is the current folder, then the folder specified by HLNK_DIR. However, when a wild card is used in the file specification, only the current folder is searched.	No value is set when the specification is omitted.
10	CPU_RX *1	Specifies the CPU type. <CPU types> RX600 RX200	No value is set when the specification is omitted.

*1) When both ISA_RX and CPU_RX are defined, ISA_RX takes precedence.

B.1.3 Options

This section describes the options for the RX family C/C++ compiler in each processing phase.

Compile phase: Refer to [B.1.3.1 Compile Options](#).

Assembly phase: Refer to [B.1.3.2 Assembler Command Options](#).

Link phase: Refer to [B.1.3.3 Optimizing Linkage Editor \(rlink\) Options](#).

Library generation phase: Refer to [B.1.3.4 Library Generator Options](#).

B.1.3.1 Compile Options

The types and explanations for options of the compile phase are shown below.

Classification	Option	Description
Source Options	-lang	Specifies the language to assume in compiling the source file.
	-include	Specifies the names of folders that hold include files.
	-preinclude	Specifies the names of files to be included at the head of each compiling unit.
	-define	Specifies macro definitions.
	-undefine	Specifies disabling of predefined macros.
	-message	Information-level messages are output.
	-nomessage	Specifies the numbers of information-level messages to be disabled.
	-change_message	Changes the levels of compiler output messages.
	-file_inline_path	Specifies the names of folders that hold files for inter-file inline expansion.
	-comment	Selects permission for comment (<code>/* */</code>) nesting.
	-check	Checks compatibility with an existing program.
	-misra2004	Checks the source code against the MISRA-C: 2004 rules.
	-ignore_files_misra	Selects files that will not be checked against the MISRA-C: 2004 rules.
	-check_language_extension	Enables complete checking against the MISRA-C: 2004 rules for parts of the code where this would otherwise be suppressed due to use of an extended specification.
Object Options	-output	Selects the output file type.
	-noline	Selects the non-output of #line in preprocessor expansion.
	-debug	Debugging information is output to the object files.
	-nodebug	Debugging information is not output to the object files.
	-section	Changes section names to be changed.
	-stuff	Variables are allocated to sections that match their alignment values.
	-nostuff	Alignment values of variables are ignored in allocating the variables to sections.
	-instalign4	Instructions at branch destinations are aligned with 4-byte boundaries.
	-instalign8	Instructions at branch destinations are aligned with 8-byte boundaries.
	-noinstalign	Instructions at branch destinations have no specific alignment.
	-nouse_div_inst	Generates code in which no DIV, DIVU, or FDIV instructions are used for division and modular division.

Classification	Option	Description
List Options	-listfile	A source list file is output.
	-nolistfile	A source list file is not output.
	-show	Specifies the contents of the source list file.
Optimize Options	-optimize	Selects the optimization level.
	-goptimize	Outputs additional information for inter-module optimization.
	-speed	Optimization is with emphasis on execution performance.
	-size	Optimization is with emphasis on code size.
	-loop	Specifies a maximum number for loop-expansion.
	-inline	Inline expansion is processed automatically.
	-noinline	Inline expansion is not processed automatically.
	-file_inline	Specifies a file for inter-file inline expansion.
	-case	Selects the method of expansion for switch statements.
	-volatile	External variables are handled as if they are all volatile qualified.
	-novolatile	External variables are handled as if none of them have been declared volatile .
	-const_copy	Enables constant propagation of const qualified external variables.
	-noconst_copy	Disables constant propagation of const qualified external variables.
	-const_div	Divisions and remainders of integer constants are converted into instruction sequences.
	-noconst_div	Divisions and remainders of integer constants are not converted into instruction sequences.
	-library	Selects the method for the execution of library functions.
	-scope	Selects division of the ranges for optimization into multiple sections before compilation.
	-noscope	Selects non-division of the ranges for optimization into multiple sections before compilation.
	-schedule	Pipeline processing is considered in scheduling instructions.
	-noschedule	Scheduling is not applied to instruction execution.
	-map	All access to external variables is optimized.
	-smap	Access to external variables is optimized as defined in the file to be compiled.
	-nomap	Access to external variables is not optimized.
	-approxdiv	Division of floating-point constants is converted into multiplication.
	-enable_register	Variables with the register storage class specification are given preference for allocation to registers.

Classification	Option	Description
Optimize Options	-simple_float_conv	Part of the type conversion processing between the floating-point type and the integer type is omitted.
	-fpu	Floating-point calculation instructions are used.
	-nofpu	Floating-point calculation instructions are not used.
	-alias	Optimization is performed in consideration of the types of data indicated by pointers.
	-float_order	The orders of operations in floating-point expressions are modified for optimization.
	-ip_optimize	Selects global optimization.
	-merge_files	The results of compiling multiple source files are output to a single object file.
	-whole_program	Makes the compiler perform optimization on the assumption that all source files have been input.
Microcontroller Options	-isa	Selects the instruction-set architecture.
	-cpu	Selects the microcontroller type.
	-endian	Selects the endian type.
	-round	Selects the rounding method for floating-point constant operations.
	-denormalize	Selects the operation when denormalized numbers are used to describe floating-point constants.
	-dbl_size	Selects the precision of the double and long double types.
	-int_to_short	Replaces the int type with the short type and the unsigned int type with the unsigned short type.
	-signed_char	Variables of the char type are handled as signed char .
	-unsigned_char	Variables of the char type are handled as unsigned char .
	-signed_bitfield	The sign bits of bit-fields are taken as signed .
	-unsigned_bitfield	The sign bits of bit-fields are taken as unsigned .
	-auto_enum	Selects whether or not the sizes for enumerated types are automatically selected.
	-bit_order	Selects the order of bit-field members.
	-pack	Specifies one as the boundary alignment value for structure members and class members.
	-unpack	Aligns structure members and class members to the alignment boundaries for the given data types.
	-exception	Enables the exception handling function.
	-noexception	Disables the exception handling function.
	-rtti	Selects enabling or disabling of C++ runtime type information (dynamic_cast or typeid).
	-fint_register	Selects a general register for exclusive use with the fast interrupt function.

Classification	Option	Description
Microcontroller Options	-branch	Selects the maximum size or no maximum size for branches.
	-base	Specifies the base registers for ROM and RAM.
	-patch	Selects avoidance or non-avoidance of a problem specific to the CPU type.
	-pic	Enables the PIC function.
	-pid	Enables the PID function.
	-nouse_pid_register	The PID register is not used in code generation.
	-save_acc	The contents of ACC are saved and restored in interrupt functions.
Assemble and Linkage Options	-asmcmd	Specifies a subcommand file for asrx options.
	-lnkcmd	Specifies a subcommand file for rlink options.
	-asmopt	Specifies asrx options.
	-lnkopt	Specifies rlink options.
Other Options	-logo	Selects the output of copyright information.
	-nologo	Selects the non-output of copyright information.
	-euc	The character codes of input programs are interpreted as EUC codes.
	-sjis	The character codes of input programs are interpreted as SJIS codes.
	-latin1	The character codes of input programs are interpreted as ISO-Latin1 codes.
	-utf8	The character codes of input programs are interpreted as UTF-8 codes.
	-big5	The character codes of input programs are interpreted as BIG5 codes.
	-gb2312	The character codes of input programs are interpreted as GB2312 codes.
	-outcode	Selects the character coding for an output assembly-language file.
	-subcommand	Specifies a file for including command options.

Source Options

< [Compile Options](#) / [Source Options](#) >

The following source options are available.

- [-lang](#)
- [-include](#)
- [-preinclude](#)
- [-define](#)
- [-undefine](#)
- [-message](#)
- [-nomessage](#)
- [-change_message](#)
- [-file_inline_path](#)
- [-comment](#)
- [-check](#)
- [-misra2004](#)
- [-ignore_files_misra](#)
- [-check_language_extension](#)

-lang

< [Compile Options](#) / [Source Options](#) >

[Format]

<code>-lang= { c cpp ecpp c99 }</code>
--

- [Default]

If this option is not specified, the compiler will compile the program file as a C++ source file when the extension is **cpp**, **cc**, or **cp**, and as a C (C89) source file for any other extensions. However, if the extension is **src** or **s**, the program file is handled as an assembly-language file regardless of whether this option is specified.

[Description]

- This option specifies the language of the source file.
- When the **lang=c** option is specified, the compiler will compile the program file as a C (C89) source file.
- When the **lang=cpp** option is specified, the compiler will compile the program file as a C++ source file.
- When the **lang=ecpp** option is specified, the compiler will compile the program file as an Embedded C++ source file.
- When the **lang=c99** option is specified, the compiler will compile the program file as a C (C99) source file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Language of the C source file\]](#) and [\[Language of the C++ source file\]](#) in the [\[Source\]](#) category

[Remarks]

- The Embedded C++ language specification does not support a **catch**, **const_cast**, **dynamic_cast**, **explicit**, **mutable**, **namespace**, **reinterpret_cast**, **static_cast**, **template**, **throw**, **try**, **typeid**, **typename**, **using**, multiple inheritance, or virtual base class. If one of these classes is written in the source file, the compiler will display an error message.
- Always specify the **lang=ecpp** option when using an EC++ library.
- In batch compilation (when multiple C/C++ language source files are input to the compiler at the same time), the individual C or C++ language source files must be in the same language. Thus, separate the C and C++ language source files in accord with the languages to be specified and then perform batch compilation by specifying this option for the group in each of the languages.

-include

< [Compile Options](#) / [Source Options](#) >

[Format]

`-include=<path name>[,...]`

[Description]

- This option specifies the name of the path to the folder that stores the include file.
- Multiple path names can be specified by separating them with a comma (,).
- Searching for files with names enclosed in "<" and ">" proceeds in order of the folders specified by the **include** option, the folders specified by environment variable **INC_RX**.
- Searching for files with names enclosed in double quotation marks (") proceeds in order of the storage folder of the file for which the #include statement is made, the folders specified by the **include** option, the folders specified by environment variable **INC_RX**.
- If two or more folders are specified in the **include** option, searching proceeds in order of specifications of the path-names for the folders on the command line (from left to right).
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Additional include paths\]](#) and [\[System include paths\]](#) in the [Source] category

[Remarks]

- If this option is specified for more than one time, all specified path names are valid.

-preinclude

< [Compile Options](#) / [Source Options](#) >

[Format]

```
-preinclude=<file name>[,...]
```

[Description]

- This option includes the specified file contents at the head of the compiling unit. Multiple file names can be specified by separating them with a comma (,).
- If there is more than one folder specified by the **preinclude** option, search is performed in turn starting from the left-most folder.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Include files at the head of compiling units\]](#) in the [Source] category

[Remarks]

- If this option is specified for more than one time, all specified files will be included.

-define

< [Compile Options](#) / [Source Options](#) >

[Format]

```
-define=<sub>[,...]  
    <sub>: <macro name> [= <string>]
```

[Description]

- This option provides the same function as **#define** specified in the source file.
- <string> can be defined as a macro name by specifying <macro name>=<string>.
- When only <macro name> is specified as a suboption, the macro name is assumed to be defined. Names or integer constants can be written in <string>.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Macro definition\]](#) in the [Source] category

[Remarks]

- If the macro name specified by this option has already been defined in the source file by **#define**, **#define** takes priority.
- If this option is specified for more than one time, all specified macro names are valid.

-undefine

< [Compile Options](#) / [Source Options](#) >

[Format]

```
-undefine=<sub>[ , ... ]  
    <sub>: <macro name>
```

[Description]

- This option invalidates the predefined macro of <macro name>.
- Multiple macro names can be specified by separating them with a comma (,).
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Invalidates the predefined macro\]](#) in the [Source] category

[Remarks]

- For the specifiable predefined macros, refer to Predefined Macros.
- If this option is specified for more than one time, all specified macro names will be undefined.

-message

< [Compile Options](#) / [Source Options](#) >

[Format]

-message

[Description]

- This option outputs the information-level messages.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Enables information-level message output\]](#) in the [Source] category

[Remarks]

- Message output from the assembler or optimizing linkage editor cannot be controlled by this option. Message output from the optimizing linkage editor can be controlled by using the **Inkcmd** option to specify the **message** or **nomessage** option of the optimizing linkage editor.

-nomessage

< [Compile Options](#) / [Source Options](#) >

[Format]

```
-nomessage [= <error number> [- <error number>]][, ...]
```

[Description]

- When the **nomessage** option is specified, output of the information-level messages is disabled. When an error number is specified as a suboption, the output of the specified information-level message will be disabled. Multiple error numbers can be specified by separating them with a comma (.).
- A range of error numbers to be disabled can be specified by using a hyphen (-), that is, in the form of <error number>-<error number>.
- Error numbers are specified by the five lower-order digits (i.e. five digits from the right) of message numbers without the prefix "M" (information).
Example: To change the level of information message M0523009
-nomessage=23009
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Enables information-level message output\]](#), [\[Suppresses the number of information-level messages\]](#) in the [\[Source\]](#) category

[Remarks]

- Message output from the assembler or optimizing linkage editor cannot be controlled by this option. Message output from the optimizing linkage editor can be controlled by using the **Inkcmd** option to specify the **message** or **nomessage** option of the optimizing linkage editor.
- If the **nomessage** option is specified for more than one time, output for all specified error numbers will be disabled.
- This option is only specifiable for messages with number 0510000 to 0549999 (including the component number).
- This option can only be used to suppress the output of messages 0520000 to 0529999. The output of other messages is not suppressed even if their numbers are specified with this option. If you wish to suppress the output of such messages, also use **-change_message** to change them to information messages.

-change_message

< [Compile Options](#) / [Source Options](#) >

[Format]

```
-change_message = <sub>[,...]  
                <sub>: <error level>[=<error number>[- <error number>][,...]]  
                <error level>: { information | warning | error }
```

[Description]

- This option changes the message level of information-level and warning-level messages.
- Multiple error numbers can be specified by separating them with a comma (,).
- Error numbers are specified by the five lower-order digits (i.e. five digits from the right) of the message numbers without the prefix "M" (information) or "W" (warning).
Example: To change the level of information message M0523009
-change_message=error=23009
- Although this option may change the types of some messages (e.g. error (E) or warning (W)), the meaning of the message indicated by the component or message number remains the same.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\]](#) tab, [\[Changes the warning-level messages to information-level messages\]](#), [\[Error number of warning-level message\]](#), [\[Changes the information-level messages to warning-level messages\]](#), [\[Error number of information-level message\]](#), [\[Changes the information-level and warning-level messages to error-level messages\]](#), [\[Error number of information-level and warning-level message\]](#) in the [\[Source\]](#) category

[Example]

```
change_message=information=error number
```

- Warning-level messages with the specified error numbers are changed to information-level messages.

```
change_message=warning=error number
```

- Information-level messages with the specified error numbers are changed to warning-level messages.

```
change_message=error=error number
```

- Information-level and warning-level messages with the specified error numbers are changed to error-level messages.

```
change_message=information
```

- All warning-level messages are changed to information-level messages.

```
change_message=warning
```

- All information-level messages are changed to warning-level messages.

```
change_message=error
```

- All information-level and warning-level messages are changed to error-level messages.

[Remarks]

- The output of messages which have been changed to information-level messages can be disabled by the **nomessage** option.

- Message output from the assembler or optimizing linkage editor cannot be controlled by this option. Message output from the optimizing linkage editor can be controlled by using the **Inkcmd** option to specify the **message** or **nomessage** option of the optimizing linkage editor.
- If this option is specified for more than one time, all specified error numbers are valid.
- Only the levels of warning and information messages can be controlled by this option. Specification of the option for a message not at these levels is ignored.
- This option is not usable to control the level of MISRA2004 detection messages (labeled M) that appear when the **misra2004** option has been specified.

-file_inline_path

< [Compile Options](#) / [Source Options](#) >

[Format]

`-file_inline_path=<path name>[,...]`

[Description]

- This option is not available in V.2.00. Any specification of this option will simply be ignored and will not lead to an error due to compatibility with former versions.

-comment

< [Compile Options](#) / [Source Options](#) >

[Format]

```
-comment = { nest | nonest }
```

[Description]

- When **comment=nest** is specified, nested comments are allowed to be written in the source file.
- When **comment=nonest** is specified, writing nested comments will generate an error.
- The default for this option is **comment=nonest**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Permits comment \(/ * */\) nesting\]](#) in the [\[Source\]](#) category

[Example]

- When **comment=nest** is specified, the compiler handles the above line as a nested comment; however, when **comment=nonest** is specified, the compiler assumes (1) as the end of the comment.

```
/* This is an example of /* nested */ comment */  
                                (1)
```

-check[< Compile Options / Source Options >](#)**[Format]**`-check = { nc | ch38 | shc }`**[Description]**

- This option checks the specified options and source file parts which will affect the compatibility when this compiler uses a C/C++ source file that has been coded for the R8C and M16C family C compilers, H8, H8S, and H8SX family C/C++ compilers, and SuperH family C/C++ compilers.
- For **check=nc**, the compatibility with the R8C and M16C family C compilers is checked. Checking will be for the following options and types:
 - Options: signed_char, signed_bitfield, bit_order=left, endian=big, and dbl_size=4
 - inline, enum type, #pragma BITADDRESS, #pragma ROM, #pragma PARAMETER, and asm()
 - Assignment of a constant outside the **signed short** range to the **int** or **signed int** type or assignment of a constant outside the **unsigned short** range to the **int** or **unsigned int** type while **-int_to_short** is not specified
 - Assignment of a constant outside both of the **signed short** and **unsigned short** ranges to the **long** or **long long** type
 - Comparison expression between a constant outside the **signed short** range and the **int**, **short**, or **char** type (except the **signed char** type)
- For **check=ch38**, the compatibility with the H8, H8S, and H8SX family C/C++ compilers is checked. Checking will be for the following options and types:
 - Options: unsigned_char, unsigned_bitfield, bit_order=right, endian=little, and dbl_size=4
 - __asm and #pragma unpack
 - Comparison expression with a constant greater than the maximum value of **signed long**
 - Assignment of a constant outside the **signed short** range to the **int** or **signed int** type or assignment of a constant outside the **unsigned short** range to the **int** or **unsigned int** type while **-int_to_short** is not specified
 - Assignment of a constant outside both of the **signed short** and **unsigned short** ranges to the **long** or **long long** type
 - Comparison expression between a constant outside the signed short range and the **int**, **short**, or **char** type (except the **signed char** type)
- For **check=shc**, the compatibility with the SuperH family C/C++ compilers is checked. Checking will be for the following options and types:
 - Options: unsigned_char, unsigned_bitfield, bit_order=right, endian=little, dbl_size=4, and round=nearest
 - #pragma unpack
 - **volatile** qualified variables
 - Confirm the following notes for the displayed items.
 - Options: The settings which are not defined in the language specification and depend on implementation differ in each compiler. Confirm the settings of the options that were output in a message.
 - Extended specifications: There is a possibility that extended specifications will affect program operation. Confirm the descriptions on the extended specifications that were output in a message.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Checks the compatibility with an existing program\]](#) in the [\[Source\]](#) category

[Remarks]

- When **dbl_size=4** is enabled, the results of type conversion related to floating-point numbers and the results of library calculation may differ from those in the R8C and M16C family C compilers, H8, H8S, and H8SX family C/C++ compilers, and SuperH family C/C++ compilers. When **dbl_size=4** is specified, this compiler handles **double** type and **long double** type as 32 bits, but the R8C and M16C family C compilers (**fdouble_32**), H8, H8S, and H8SX family C/C++ compilers (**double=float**), and SuperH family C/C++ compilers (**double=float**) handle only **double** type as 32 bits.
- The result of a binary operation (addition, subtraction, multiplication, division, comparison, etc.) with **unsigned int** type and **long** type operands may differ from that in the SuperH family C/C++ compilers. In this compiler, the types of the operands are converted to the **unsigned long** type before operation. However, in the SuperH family C/C++ compilers (only when **strict_ansi** is not specified), the types of the operands are converted to the **signed long long** type before operation.
- The data size of reading from and writing to a **volatile** qualified variable may differ from that in the SuperH family C/C++ compilers. This is because a **volatile** qualified bit field may be accessed in a size smaller than that of the declaration type in this compiler. However, in the SuperH family C/C++ compilers, a **volatile** qualified bit field is accessed in the same size as that of the declaration type.
- This option does not output a message regarding allocation of structure members and bit field members. When an allocation-conscious declaration is made, refer to 3.1.4 Internal Data Representation and Areas in User's Manual: RX Coding.
- In the R8C and M16C family C compilers (**fextend_to_int** is not specified), the generated code has been evaluated without performing generalized integer promotion by a conditional expression. Accordingly, operation of such a code may differ from a code generated by this compiler.

-misra2004

< [Compile Options](#) / [Source Options](#) >

[Format]

```
-misra2004 = {
  all
  | apply=<rule number>[,<rule number>,...]
  | ignore=<rule number>[,<rule number>,...]
  | required
  | required_add=<rule number>[,<rule number>,...]
  | required_remove=<rule number>[,<rule number>,...]
  | <filename> }
```

[Description]

- This option enables checking against the MISRA-C: 2004 rules and to select specific rules to be used.
- When **misra2004=all**, the compiler checks the source code against all of the rules that are supported.
- When **misra2004=apply=<rule number>[,<rule number>,...]**, the compiler checks the source code against the rules with the selected numbers.
- When **misra2004=ignore=<rule number>[,<rule number>,...]**, the compiler checks the source code against the rules other than those with the selected numbers.
- When **misra2004=required**, the compiler checks the source code against the rules of the "required" type.
- When **misra2004=required_add=<rule number>[,<rule number>,...]**, the compiler checks the source code against the rules of the "required" type and the rules with the selected numbers.
- When **misra2004=required_remove=<rule number>[,<rule number>,...]**, the compiler checks the source code against the rules other than those with the selected numbers among the rules of the "required" type.
- When **misra2004=<filename>**, the compiler checks the source code against the rules with the numbers written in the specified file. One rule number is written per line in the file. Each rule number must be specified by using a decimal value and a period (".").
- When checking of a line of code against the MISRA-C: 2004 rules leads to detection of a violation, a message in the following format will appear.
Filename (line number): M0523028 Rule number: Message
- When **-misra2004=<filename>** is used more than once, only the last specification is valid.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Apply rule\]](#), [\[Rule number description file\]](#), [\[Rule number\]](#), [\[Exclusion rule number\]](#), [\[Check rule number besides required rule\]](#), [\[Exclusion rule number from required rule\]](#) in the [\[MISRA C rule check\]](#) category

[Remarks]

- **-misra2004** can be used multiple times. If two or more types (which follow **-misra2004=**) are specified, however, only the type of the last specification is valid.
... -misra2004=ignore=2.2 -misra2004=apply=2.3
-misra2004=required_add=4.1 -misra2004=apply=4.2
-misra2004=apply=5.2 ...
In this example, **ignore**, **apply**, and **required_add** are specified, but only **apply** (used in the last two cases) is valid. The compiler will check the source code against rules 4.2 and 5.2.
- When the number of an unsupported rule is specified for **<rule number>**, the compiler detects error C0523031 and stops the processing.

- When the file specified in **misra2004=<filename>** cannot be opened, the compiler detects error C0523029. When rule numbers are not extractable from the specified file, the compiler detects error C0523030. Processing by the compiler stops in both cases.
- This option is ignored when **cpp**, **c99**, or **ecpp** is selected for the **lang** option or when **output=prep** is specified at the same time.
- This option supports the MISRA-C: 2004 rules listed below.
 - [Required]
 - 2.2 2.3
 - 4.1 4.2
 - 5.2 5.3 5.4
 - 6.1 6.2 6.4 6.5
 - 7.1
 - 8.1 8.2 8.3 8.5 8.6 8.7 8.11 8.12
 - 9.1 9.2 9.3
 - 10.1 10.2 10.3 10.4 10.5 10.6
 - 11.1 11.2 11.5
 - 12.3 12.4 12.5 12.7 12.8 12.9 12.10 12.12
 - 13.1 13.3 13.4
 - 14.2 14.3 14.4 14.5 14.6 14.7 14.8 14.9 14.10
 - 15.1 15.2 15.3 15.4 15.5
 - 16.1 16.3 16.5 16.6 16.9
 - 18.1 18.4
 - 19.3 19.6 19.8 19.11 19.14 19.15
 - 20.4 20.5 20.6 20.7 20.8 20.9 20.10 20.11 20.12
 - [Not required]
 - 5.5 5.6
 - 6.3
 - 11.3 11.4
 - 12.1 12.6 12.11 12.13
 - 13.2
 - 17.5
 - 19.7 19.13
- For source programs that use extended functions such as **#pragma**, checking against these rules will be suppressed under some conditions. For details, refer to the section on the **check_language_extension** option.

-ignore_files_misra

< [Compile Options](#) / [Source Options](#) >

[Format]

`-ignore_files_misra=<filename>[,<filename>,...]`

[Description]

- This option selects files that will not be checked against the MISRA-C: 2004 rules.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Rule check exclusion file\]](#) in the [\[MISRA C rule check\]](#) category

[Remarks]

- If a single option is specified more than once in the command line, all specifications are valid.
- This option is ignored when the -misra2004 option has not been specified.
- <filename> is ignored when the specified file is not to be compiled.

-check_language_extension

< [Compile Options](#) / [Source Options](#) >

[Format]

```
-check_language_extension
```

[Description]

- This option enables complete checking against the MISRA-C: 2004 rules for parts of the code where it would otherwise be suppressed due to individual extensions from the C/C++ language specification.
- With the default `misra2004` option, the compiler does not proceed with checking against the MISRA-C: 2004 rules under the condition given below. To enable complete checking, specify the **check_language_extension** option.
- Condition:
 - The function has no prototype declaration (rule 8.1) and **#pragma entry** or **#pragma interrupt** is specified for it.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Outputs message of the enhanced key word and extended specifications\]](#) in the [\[MISRA C rule check\]](#) category

[Example]

```
#pragma interrupt vfunc
extern void service(void);
void vfunc(void)
{
    service();
}
```

- Function **vfunc**, for which **#pragma interrupt** is specified, has no prototype declaration. Even when this function is compiled with **-misra2004=all** specified, the message on rule 8.1 is not displayed unless the **check_language_extension** option is specified.

[Remarks]

This option is ignored when the **-misra2004** option has not been specified.

Object Options

< [Compile Options](#) / [Object Options](#) >

The following object options are available.

- [-output](#)
- [-noline](#)
- [-debug](#)
- [-nodebug](#)
- [-section](#)
- [-stuff](#)
- [-nostuff](#)
- [-instalign4](#)
- [-instalign8](#)
- [-noinstalign](#)
- [-nouse_div_inst](#)

-output

< [Compile Options](#) / [Object Options](#) >

[Format]

```
-output = <sub> [=<file name>]
        <sub>: { prep | src | obj | abs | hex | sty }
```

- [Default]

The default for this option is **output=obj**.

[Description]

- This option specifies the output file type.
- The suboptions and output files are shown in the following table.
- If no **<file name>** is specified, a file will be generated with an extension, that is shown in the following table, appended to the source file name input at the beginning.

Table B.3 Suboption Output Format

Suboption	Output File Type	Extension When File Name is Not Specified
prep	Source file after preprocessed	C (C89, C99) source file: p C++ source file: pp
src	Assembly-language file	src
obj	Relocatable file	obj
abs	Absolute file	abs
hex	Intel hex type file	hex
sty	Motorola S type file	mot

Note Relocatable files are files output from the assembler.
Absolute files, Intel hex type files, and Motorola S type files are files output from the optimizing linkage editor.

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Output file type\]](#) in the [\[Object\]](#) category

[Remarks]

- An intermediate file used to generate a file of the specified type is stored in the specified folder; however, when no folder has been specified, the intermediate file is stored in the current folder.

-noline

< [Compile Options](#) / [Object Options](#) >

[Format]

-noline

[Description]

- This option disables **#line** output during preprocessor expansion.

[Remarks]

- This option is validated when the **output=prep** option has not been specified.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Output file type\]](#) in the [\[Object\]](#) category

-debug

< [Compile Options](#) / [Object Options](#) >

[Format]

-debug

[Description]

- When the **debug** option is specified, debugging information necessary for C-source debugging is output. The **debug** option is valid even when an optimize option is specified.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Outputs debugging information\]](#) in the [Object] category

-nodebug

< [Compile Options](#) / [Object Options](#) >

[Format]

-nodebug

[Description]

- When the **nodebug** option is specified, no debugging information is output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Outputs debugging information\]](#) in the [Object] category

-section

< [Compile Options](#) / [Object Options](#) >

[Format]

```
-section = <sub>[,...]  
    <sub>: { P = <section name> |  
           C = <section name> |  
           D = <section name> |  
           B = <section name> |  
           L = <section name> |  
           W = <section name> }
```

[Description]

- This option specifies the section name.
- **section=P=<section name>** specifies the section name of a program area.
- **section=C=<section name>** specifies the section name of a constant area.
- **section=D=<section name>** specifies the section name of an initialized data area.
- **section=B=<section name>** specifies the section name of an uninitialized data area.
- **section=L=<section name>** specifies the section name of a literal area.
- **section=W=<section name>** specifies the section name of a **switch** statement branch table area.
- <section name> must be alphabetic, numeric, underscore (_), or \$. The first character must not be numeric.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Section name of program area\]](#), [\[Section name of constant area\]](#), [\[Section name of initialized data area\]](#), [\[Section name of uninitialized data area\]](#), [\[Section name of literal area\]](#), and [\[Section name of switch statement branch table area\]](#) in the [\[Object\]](#) category

[Remarks]

- The default for this option is **section=P=P,C=C,D=D,B=B,L=L,W=W**.
- In the same way as in V. 1.00, if you want to output the literal area in the **C** section rather than output a separate **L** section, select **section=L=C**.
- Except for changing the **L** section to the same section name as that of the **C** section, the same section name cannot be specified for the sections for different areas.
- For the translation limit of the section name length, refer to Translation Limits.

-stuff

< [Compile Options](#) / [Object Options](#) >

[Format]

```
-stuff
```

[Description]

- When the **stuff** option is specified, all variables are allocated to 4-byte, 2-byte, or 1-byte boundary alignment sections depending on the alignment value (see table B-4).

Table B.4 Correspondences between Variables and Their Output Sections When stuff Option is Specified

Variable Type	Alignment Value for Variable	Section to Which Variable Belongs
const qualified variables	4	C
	2	C_2
	1	C_1
Initialized variables	4	D
	2	D_2
	1	D_1
Uninitialized variables	4	B
	2	B_2
	1	B_1
switch statement branch table	4	W
	2	W_2
	1	W_1

- **C**, **D**, and **B** are the section names specified by the **section** option or **#pragma section**. **W** is the section name specified by the **section** option. The data contents allocated to each section are output in the order they were defined, except that variables that do not have the initial value are output after those that have the initial value in section C.

[Example]

```
int a;
char b=0;
const short c=0;
struct {
    char x;
    char y;
} ST;
```

	.SECTION		C_2,ROMDATA,ALIGN=2
	.glb	_c	
_c:	.word		0000H
	.SECTION		D_1,ROMDATA
	.glb	_b	
_b:	.byte		00H
	.SECTION		B,DATA,ALIGN=4
	.glb	_a	
_a:	.blk1	1	
	.SECTION		B_1,DATA
	.glb	_ST	
_ST	.blkb	2	

[Remarks]

- The **stuff** option has no effect for sections other than **B**, **D**, **C**, and **W**.

-nostuff< [Compile Options](#) / [Object Options](#) >**[Format]**

```
-nostuff [= <section type>[,...]]
        <section type>: { B | D | C | W }
```

[Description]

- When the **nostuff** option is specified, the compiler allocates the variables belonging to the specified <section type> to 4-byte boundary alignment sections. When **<section type>** is omitted, variables of all section types are applicable.
- **C**, **D**, and **B** are the section names specified by the **section** option or **#pragma** section. **W** is the section name specified by the section option. The data contents allocated to each section are output in the order they were defined, except that variables that do not have the initial value are output after those that have the initial value in section C.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\]](#) tab, [\[Allocates uninitialized variables to 4-byte boundary alignment sections\]](#), [\[Allocates initialized variables to 4-byte boundary alignment sections\]](#), [\[Allocates const qualified variables to 4-byte boundary alignment sections\]](#), and [\[Allocates switch statement branch tables to 4-byte boundary alignment sections\]](#) in the [\[Object\]](#) category

[Example]

```
int a;
char b=0;
const short c=0;
struct {
    char x;
    char y;
} ST;
```

```

        .SECTION          C,ROMDATA,ALIGN=4
        .glb              _c
_c:
        .word              0000H
        .SECTION          D,ROMDATA,ALIGN=4
        .glb              _b
_b:
        .byte              00H
        .SECTION          B,DATA,ALIGN=4
        .glb              _a
_a:
        .blk1              1
        .glb              _ST
_ST
        .blkb              2
```

[Remarks]

- The **nostuff** option cannot be specified for sections other than **B**, **D**, **C**, and **W**.

-instalign4

< [Compile Options](#) / [Object Options](#) >

[Format]

```
-instalign4[={loop|inmostloop}]
```

[Description]

- This option aligns instructions at branch destinations.
- When the **instalign4** option is specified, the instruction at the location address is aligned to the 4-byte boundary.
- Instruction alignment is performed only when the instruction at the specified location exceeds the address which is a multiple of the alignment value (4)*¹.
- The following three types of branch destination can be selected by specifying the suboptions of **-instalign4***².
 - No specification: Head of function and **case** and **default** labels of **switch** statement
 - inmostloop**: Head of each inmost loop, head of function, and **case** and **default** labels of **switch** statement
 - loop**: Head of each loop, head of function, and **case** and **default** labels of **switch** statement
- When this option is selected, the alignment value of the program section is changed from 1 to 4 (for **instalign4**) or 8 (for **instalign8**).
- This option aims to efficiently operate the instruction queues of the RX CPU and improve the speed of program execution by aligning the addresses of branch destination instructions.
This option has specifications targeting the following usage.
- **instalign4**: When attempting to improve the speed of CPUs with a 32-bit instruction queue (mainly RX200 Series)
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Adjustment for instruction in branch\]](#) in the [\[Object\]](#) category

Notes 1. This is when the instruction size is equal to or smaller than the alignment value. If the instruction size is greater than the alignment value, alignment is performed only when the number of exceeding points is two or more.

Notes 2. Alignment is adjusted only for the branch destinations listed above; alignment of the other destinations is not adjusted. For example, when **loop** is selected, alignment of the head of a **loop** is adjusted but alignment is not adjusted at the branch destination of an **if** statement that is used in the loop but does not generate a loop.

-instalign8

< [Compile Options](#) / [Object Options](#) >

[Format]

```
-instalign8[={loop|inmostloop}]
```

[Description]

- This option aligns instructions at branch destinations.
- When the **instalign8** option is specified, the instruction at the location address is aligned to the 8-byte boundary.
- Instruction alignment is performed only when the instruction at the specified location exceeds the address which is a multiple of the alignment value (8)*¹.
- The following three types of branch destination can be selected by specifying the suboptions of **-instalign4** and **-instalign8***².
 - No specification: Head of function and **case** and **default** labels of **switch** statement
 - inmostloop**: Head of each inmost loop, head of function, and **case** and **default** labels of **switch** statement
 - loop**: Head of each loop, head of function, and **case** and **default** labels of **switch** statement
- When these options are selected, the alignment value of the program section is changed from 1 to 4 (for **instalign4**) or 8 (for **instalign8**).
- These options aim to efficiently operate the instruction queues of the RX CPU and improve the speed of program execution by aligning the addresses of branch destination instructions.
This option has specifications targeting the following usage.
- **instalign8**: When attempting to improve the speed of CPUs with a 64-bit instruction queue (mainly RX600 Series)
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Adjustment for instruction in branch\]](#) in the [\[Object\]](#) category

Notes 1. This is when the instruction size is equal to or smaller than the alignment value. If the instruction size is greater than the alignment value, alignment is performed only when the number of exceeding points is two or more.

Notes 2. Alignment is adjusted only for the branch destinations listed above; alignment of the other destinations is not adjusted. For example, when **loop** is selected, alignment of the head of a loop is adjusted but alignment is not adjusted at the branch destination of an **if** statement that is used in the loop but does not generate a loop.

[Example]

- <C source file>

```
dlong a;
int f1(int num)
{
    return (num+1);
}
void f2(void)
{
    a = 0;
}
void f3(void)
{
}
```

- <Output code>

[When compiling with -instalgn8 specified]

In the example shown below, the head of each function is aligned so that the instruction does not exceed the 8-byte boundary.

In 8-byte boundary alignment of instructions, the address will not be changed unless the target instruction exceeds the 8-byte boundary. Therefore, only the address of function f2 is actually aligned.

```
.SECTION P, CODE, ALIGN=8
.INSTALIGN 8
_f1:                                ; Function f1, address = 0000H
    ADD    #01H, R1                ; 2 bytes
    RTS                                         ; 1 byte
.INSTALIGN 8
_f2:                                ; Function f2, address = 0008H
                                         ; Note: Alignment is performed.
                                         ; When a 6-byte instruction is placed at
                                         ; 0003H, it exceeds the 8-byte boundary.
                                         ; Thus, alignment is performed.
    MOV.L   #_a, R4                ; 6 bytes
    MOV.L   #0, [R4]               ; 3 bytes
    RTS                                         ; 1 byte
.INSTALIGN 8
_f3:                                ; Function f3, address = 0012H
    RTS
.END
```

-noinstalign

< [Compile Options](#) / [Object Options](#) >

[Format]

-noinstalign

[Description]

- This option does not align instructions at branch destinations.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\]](#) tab, [\[Adjustment for instruction in branch\]](#) in the [\[Object\]](#) category

-nouse_div_inst

< [Compile Options](#) / [Object Options](#) >

[Format]

`-nouse_div_inst`

[Description]

- This option generates code in which no DIV, DIVU, or FDIV instructions are used for division and modular division operations in the program.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Generates divisions and residues with DIV, DIVU, and the FDIV instruction\]](#) in the [\[Object\]](#) category

[Remarks]

- This option calls the equivalent runtime functions instead of DIV, DIVU, or FDIV instructions. This may lower code efficiency in terms of required ROM capacity and speed of execution.

List Options

< [Compile Options](#) / [List Options](#) >

The following list options are available.

- [-listfile](#)
- [-nolistfile](#)
- [-show](#)

-listfile

< [Compile Options](#) / [List Options](#) >

[Format]

```
-listfile[={<file name>|<path name>}]
```

[Description]

- These options specify whether to output a source list file.
- When the **listfile** option is specified, a source list file is output. <file name> can also be specified.
- An existing folder can also be specified as <path name> instead of <file name>. In such a case, a source list file with the file extension **.lst** and the name of the source file being compiled or assembled is output to the folder selected as <path name>.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Outputs a source list file\]](#) in the [List] category
- [Remarks]
- A linkage list cannot be output by this option. In order to output a linkage list, specify the **list** option of the optimizing linkage editor by using the **Inkcmd** option.
- Information output from the compiler is written to the source list. For the source list file format, refer to Assemble List File.
- When you use <path name>, create the folder in advance. If the folder specified as <path name> does not exist, the compiler will assume that <file name> is selected.

-nolistfile

< [Compile Options](#) / [List Options](#) >

[Format]

<code>-nolistfile</code>

[Description]

- When the **nolistfile** option is specified, no source list file is output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Outputs a source list file\]](#) in the [List] category

-show

< [Compile Options](#) / [List Options](#) >

[Format]

```
-show=<sub>[,...]  
  <sub>: { source | conditionals | definitions | expansions }
```

[Description]

- This option sets the source list file contents.
- The suboptions and specified contents are shown in the following table.

Table B.5 Suboption Specifications

Suboption	Description
source	Outputs the C/C++ source file.
conditionals	Outputs also the statements for which the specified condition is not satisfied in conditional assembly.
definitions	Outputs the information before .DEFINE replacement.
expansions	Outputs the assembler macro expansion statements.

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\]](#) tab, [\[Outputs the C/C++ source file\]](#), [\[Outputs the statements unsatisfied in conditional assembly\]](#), [\[Outputs the information before .DEFINE replacement\]](#), and [\[Outputs the assembler macro expansion statements\]](#) in the [\[List\]](#) category

[Remarks]

- This option is valid only when the **listfile** option has been specified.
- Information output from the compiler is written to the source list. For the source list file format, refer to Assemble List File.

Optimize Options

< [Compile Options](#) / [Optimize Options](#) >

The following optimize options are available.

- [-optimize](#)
- [-goptimize](#)
- [-speed](#)
- [-size](#)
- [-loop](#)
- [-inline](#)
- [-noinline](#)
- [-file_inline](#)
- [-case](#)
- [-volatile](#)
- [-novolatile](#)
- [-const_copy](#)
- [-const_div](#)
- [-noconst_div](#)
- [-library](#)
- [-scope](#)
- [-noscope](#)
- [-schedule](#)
- [-noschedule](#)
- [-map](#)
- [-nomap](#)
- [-approxdiv](#)
- [-enable_register](#)
- [-simple_float_conv](#)
- [-fpu](#)
- [-nofpu](#)
- [-alias](#)
- [-float_order](#)
- [-ip_optimize](#)
- [-merge_files](#)
- [-whole_program](#)

-optimize

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-optimize = { 0 | 1 | 2 | max }
```

[Description]

- This option specifies the optimization level.
- When **optimize=0** is specified, the compiler does not optimize the program. Accordingly, the debugging information may be output with high precision and source-level debugging is made easier.
- When **optimize=1** is specified, the compiler partially optimizes the program by automatically allocating variables to registers, integrating the function exit blocks, integrating multiple instructions which can be integrated, etc. Accordingly, the code size may become smaller than when compiled with the **optimize=0** specification.
- When **optimize=2** is specified, the compiler performs overall optimization. However, the optimization contents to be performed slightly differ depending on whether the **size** option or **speed** option has been selected.
- When **optimize=max** is specified, the compiler performs optimization as much as possible. For example, the optimization scope is expanded to its maximum extent, and if the **speed** option is specified, loop expansion is possible on a large scale. Though the advantages of optimization can be expected, there may be side effects, such as longer compilation time, and if the **speed** option is specified, significantly increased code size.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Optimization level\]](#) in the [Optimization] category
 - From the [\[Library Generate Options\] tab](#), [\[Optimization level\]](#) in the [Optimization] category

[Remarks]

- If the default is not included in the description of an optimize option, this means that the default varies depending on the **optimize** option and **speed** or **size** option specifications. For details on the default, refer to the **speed** or **size** option.

-goptimize

< [Compile Options](#) / [Optimize Options](#) >

[Format]

`-goptimize`

[Description]

- This option generates the additional information for inter-module optimization in the output file.
- At linkage, inter-module optimization is applied to files for which this option has been specified.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Outputs additional information for inter-module optimization\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Outputs additional information for inter-module optimization\]](#) in the [\[Optimization\]](#) category

-speed< [Compile Options](#) / [Optimize Options](#) >**[Format]**`-speed`**[Description]**

- When the **speed** option is specified, optimization will be performed with emphasis on execution performance.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Optimization type\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Optimization type\]](#) in the [\[Optimization\]](#) category

[Remarks]

- When the **speed** option is specified, the following options are automatically specified based on the **optimize** option specification.
- When **optimize=max** is specified

	Loop Expansion	Inline Expansion	Converting Constant Division into Multiplication	Scheduling Instructions	Constant Propagation of const Qualified Variables	Dividing Optimizing Ranges	Optimizing External Variable Accesses	Optimization Considering the Type of the Data Indicated by the Pointer
speed	loop=8	inline=250	const_div	schedule	const_copy	noscope	map* nomap*	alias=ansi

Note The default is **map** when a C/C++ source program has been specified for input and **output=abs** or **output=mot** has been specified for output. For any other case, the default is **nomap**.

- When **optimize=2** is specified

	Loop Expansion	Inline Expansion	Converting Constant Division into Multiplication	Scheduling Instructions	Constant Propagation of const Qualified Variables	Dividing Optimizing Ranges	Optimizing External Variable Accesses	Optimization Considering the Type of the Data Indicated by the Pointer
speed	loop=2	inline=100	const_div	schedule	const_copy	scope	nomap	alias=noansi

- When **optimize=0** or **optimize=1** is specified

	Loop Expan- sion	Inline Expan- sion	Converting Constant Division into Multiplica- tion	Schedul- ing Instruc- tions	Constant Propaga- tion of const Qual- ified Vari- ables	Dividing Optimiz- ing Ranges	Optimiz- ing Exter- nal Variable Accesses	Optimiza- tion Consid- ering the Type of the Data Indi- cated by the Pointer
speed	loop=1	noinline	const_div	nosched- ule	noconst_ copy	scope	nomap	alias=noansi

-size< [Compile Options](#) / [Optimize Options](#) >**[Format]**`-size`**[Description]**

- When the **size** option is specified, optimization will be performed with emphasis on code size.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Optimization type\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Optimization type\]](#) in the [\[Optimization\]](#) category

[Remarks]

- When the **size** option is specified, the following options are automatically specified based on the **optimize** option specification. Note however that if one of the following options is specified otherwise explicitly, that specified option becomes valid.
- When **optimize=max** is specified

	Loop Expansion	Inline Expansion	Converting Constant Division into Multiplication	Scheduling Instructions	Constant Propagation of const Qualified Variables	Dividing Optimizing Ranges	Optimizing External Variable Accesses	Optimization Considering the Type of the Data Indicated by the Pointer
size	loop=1	inline=0	noconst_div	schedule	const_copy	noscope	map* nomap*	alias=ansi

Note The default is **map** when a C/C++ source program has been specified for input and **output=abs** or **output=mot** has been specified for output. For any other case, the default is **nomap**.

- When **optimize=2** is specified

	Loop Expansion	Inline Expansion	Converting Constant Division into Multiplication	Scheduling Instructions	Constant Propagation of const Qualified Variables	Dividing Optimizing Ranges	Optimizing External Variable Accesses	Optimization Considering the Type of the Data Indicated by the Pointer
size	loop=1	noinline	noconst_div	schedule	const_copy	scope	nomap	alias=noansi

- When **optimize=0** or **optimize=1** is specified

	Loop Expan- sion	Inline Expan- sion	Converting Constant Division into Multiplica- tion	Schedul- ing Instruc- tions	Constant Propaga- tion of const Qual- ified Vari- ables	Dividing Optimiz- ing Ranges	Optimiz- ing Exter- nal Variable Accesses	Optimiza- tion Consid- ering the Type of the Data Indi- cated by the Pointer
size	loop=1	noinline	noconst_div	nosched- ule	noconst_ copy	scope	nomap	alias=noansi

-loop

< [Compile Options](#) / [Optimize Options](#) >

[Format]

<code>-loop[=<numeric value>]</code>
--

- [Default]

The default for this option is **loop=2**.

[Description]

- This option specifies whether to optimize loop expansion.
- When the **loop** option is specified, the compiler expands loop statements (**for**, **while**, and **do-while**).
- The maximum expansion factor can be specified by <numeric value>. An integer from 1 to 32 can be specified for <numeric value>. If no <numeric value> is specified, 2 will be assumed.
- The default for this option is determined based on the **optimize** option and **speed** or **size** option specifications. For details, refer to the **speed** or **size** option.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Loop expansion\]](#) and [\[Expansion maximum number\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Loop expansion\]](#) and [\[Expansion maximum number\]](#) in the [\[Optimization\]](#) category

[Remarks]

- This option is invalid when **optimize=0** or **optimize=1**.

-inline

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-inline[=<numeric value>]
```

- [Default]

The default for this option is inline=100.

[Description]

- These options specify whether to automatically perform inline expansion of functions.
- A value from 0 to 65535 is specifiable as **<numeric value>**.
- When the **inline** option is specified, the compiler automatically performs inline expansion. However, inline expansion is not performed for the functions specified by **#pragma noline**. The user is able to use **inline=<numeric value>**, to specify the allowed increase in the function's size due to the use of inline expansion. For example, when **inline=100** is specified, inline expansion will be performed until the function size has increased by 100% (size is doubled).
- The default for this option is determined based on the **optimize** option and **speed** or **size** option specifications. For details, refer to the **speed** or **size** option.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Performs inline expansion automatically\]](#) and [\[Maximum increasing rate of function size\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Performs inline expansion automatically\]](#) and [\[Maximum increasing rate of function size\]](#) in the [\[Optimization\]](#) category

[Remarks]

- Inline expansion is attempted for all functions for which **#pragma inline** has been specified or with an **inline** specifier whether other options have been specified or not. To perform inline expansion for a function for certain, specify **#pragma inline** for the function. Even though this option has been selected or an **inline** specifier has been specified for the function, if the compiler judges that the efficiency is degraded by inline expansion, it will not perform it in some cases.

-noinline

< [Compile Options](#) / [Optimize Options](#) >

[Format]

`-noinline`

[Description]

- When the **noinline** option is specified, automatic inline expansion is not performed.

[Remarks]

- Inline expansion is attempted for all functions for which **#pragma inline** has been specified or with an **inline** specifier whether other options have been specified or not.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Performs inline expansion automatically\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Performs inline expansion automatically\]](#) in the [\[Optimization\]](#) category

-file_inline

< [Compile Options](#) / [Optimize Options](#) >

[Format]

`-file_inline=<file name>[,...]`

- [Default]
None

[Description]

- This option is not available in V.2.00. Any specification of this option will simply be ignored and will not lead to an error due to compatibility with former versions.

[Remarks]

- For C (C99) source files, **-merge_files** can be used instead of **-file_inline**. Add the file that was used with **-file_inline** (including the file path if **-file_inline_path** was used together with it) as one of the source files to be merged.
- There are some points to be noted regarding **-merge_files**. Refer to [Remarks] of the **-merge_files** option.

-case

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-case={ ifthen | table | auto }
```

- [Default]

The default for this option is **case=auto**.

[Description]

- This option specifies the expansion method of the **switch** statement.
- When **case=ifthen** is specified, the **switch** statement is expanded using the **if_then** method, which repeats, for each **case** label, comparison between the value of the evaluation expression in the **switch** statement and the **case** label value. If they match, execution jumps to the statement of the **case** label. This method increases the object code size depending on the number of case labels in the **switch** statement.
- When **case=table** is specified, the **switch** statement is expanded by using the table method, where the **case** label jump destinations are stored in a branch table so that a jump to the statement of the **case** label that matches the expression for evaluation in the **switch** statement is made through a single access to the branch table. With this method, the size of the branch table increases with the number of **case** labels in the **switch** statement, but the performance in execution remains the same. The branch table is output to a section for areas holding **switch** statements for branch tables.
- When **case=auto** is specified, the compiler automatically selects the **if_then** method or table method.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Expansion method of the switch statement\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Expansion method of the switch statement\]](#) in the [\[Optimization\]](#) category

[Remarks]

- The branch table created when **case=table** has been specified will be output to section **W** when the **nostuff** option is specified and will be output to section **W**, **W_2**, or **W_1** according to the size of the **switch** statement when the **nostuff** option is not specified.

-volatile

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-volatile

[Description]

- When **volatile** is specified, all external variables are handled as if they were **volatile** qualified. Accordingly, the access count and access order for external variables are exactly the same as those written in the C/C++ source file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Handles external variables as if they are volatile qualified\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Handles external variables as if they are volatile qualified\]](#) in the [\[Optimization\]](#) category

[Remarks]

- Debugging tools for RX do not display the volatile declaration added to individual variables by this option.

-novolatile

< [Compile Options](#) / [Optimize Options](#) >

[Format]

`-novolatile`

[Description]

- When **novolatile** is specified, the external variables which are not **volatile** qualified are optimized. Accordingly, the access count and access order for external variables may differ from those written in the C/C++ source file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Handles external variables as if they are volatile qualified\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Handles external variables as if they are volatile qualified\]](#) in the [\[Optimization\]](#) category

-const_copy

< [Compile Options](#) / [Optimize Options](#) >

[Format]

`-const_copy`

- [Default]

The default for this option is **const_copy** when the **optimize=2** or **optimize=max** option has been specified.

[Description]

- When **const_copy** is specified, constant propagation is performed even for **const** qualified global variables.
- The default for this option is **const_copy** when the **optimize=2** or **optimize=max** option has been specified.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Performs the constant propagation of const qualified external variables\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Performs the constant propagation of const qualified external variables\]](#) in the [\[Optimization\]](#) category

[Remarks]

- **const** qualified variables in a C++ source file cannot be controlled by this option (constant propagation is always performed).

-noconst_copy

< [Compile Options](#) / [Optimize Options](#) >

[Format]

`-noconst_copy`

- [Default]

The default for this option is **noconst_copy** when the **optimize=1** or **optimize=0** option has been specified.

[Description]

- When **noconst_copy** is specified, constant propagation is disabled for **const** qualified global variables.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Performs the constant propagation of const qualified external variables\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Performs the constant propagation of const qualified external variables\]](#) in the [\[Optimization\]](#) category

[Remarks]

- **const** qualified variables in a C++ source file cannot be controlled by this option (constant propagation is always performed).

-const_div

< [Compile Options](#) / [Optimize Options](#) >

[Format]

`-const_div`

- [Default]

The default for this option is **const_div** when the **speed** option has been specified.

[Description]

- When **const_div** is specified, calculations for division and remainders of integer constants in the source file are converted into sequences of multiplication or bitwise operation (shift or bitwise AND operations) instructions.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Conversion method of the divisions and residues of integer constants\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Conversion method of the divisions and residues of integer constants\]](#) in the [\[Optimization\]](#) category

[Remarks]

- Constant multiplication that can be performed through only shift operations and division and residue that can be performed through only bitwise AND operations cannot be controlled by the **const_div** option.

-noconst_div

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-noconst_div

- [Default]

The default for this option is **noconst_div** when the **size** option has been specified.

[Description]

- When **noconst_div** is specified, the corresponding division and remainder instructions are used for calculating division and remainders of integer constants in the source file (except divisions and remainders of unsigned integers by powers of two).
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Conversion method of the divisions and residues of integer constants\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Conversion method of the divisions and residues of integer constants\]](#) in the [\[Optimization\]](#) category

[Remarks]

- Constant multiplication that can be performed through only shift operations and division and residue that can be performed through only bitwise AND operations cannot be controlled by the **noconst_div** option.

-library

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-library = { function | intrinsic }
```

- [Default]

The default for this option is **library=intrinsic**.

[Description]

- When **library=function** is specified, all library functions are called.
- When **library=intrinsic** is specified, instruction expansion is performed for **abs()**, **fabsf()**, and library functions which can use string manipulation instructions.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Expansion method of the library function\]](#) in the [Optimization] category
 - From the [\[Library Generate Options\] tab](#), [\[Expansion method of the library function\]](#) in the [Optimization] category

[Remarks]

- When -library=intrinsic and -isa=rxv2 are selected at the same time, calls of the sqrtf function or the sqrt function (when -dbl_size=4) are expanded as FSQRT instructions. Note, however, that no value is set for errno in such cases.

-scope

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-scope

- [Default]

The default for this option is **scope** when the **optimize=max** option has been specified.

[Description]

- When the **scope** option is specified, the optimizing ranges of the large-size function are divided into many sections before compilation.
- Use this option at performance tuning because it affects the object performance depending on the program.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Divides the optimizing ranges into many sections before compilation\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Divides the optimizing ranges into many sections before compilation\]](#) in the [\[Optimization\]](#) category

-noscope

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-noscope

- [Default]

The default for this option is **noscope** when the **optimize=max** option has been specified.

[Description]

- When the **noscope** option is specified, the optimizing ranges are not divided before compilation. When the optimizing range is expanded, the object performance is generally improved although the compilation time is delayed. However, if registers are not sufficient, the object performance may be lowered. Use this option at performance tuning because it affects the object performance depending on the program.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Divides the optimizing ranges into many sections before compilation\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Divides the optimizing ranges into many sections before compilation\]](#) in the [\[Optimization\]](#) category

-schedule

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-schedule

- [Default]

The default for this option is **schedule** when the **optimize=2** or **optimize=max** option has been specified.

[Description]

- When the **schedule** option is specified, instructions are scheduled taking into consideration pipeline processing.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Schedules the instruction taking into consideration pipeline processing\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Schedules the instruction taking into consideration pipeline processing\]](#) in the [\[Optimization\]](#) category

-noschedule

< [Compile Options](#) / [Optimize Options](#) >

[Format]

<code>-noschedule</code>

- [Default]

The default for this option is **noschedule** when the **optimize=1** or **optimize=0** option has been specified.

[Description]

- When the **noschedule** option is specified, instructions are not scheduled. Basically, processing is performed in the same order the instructions have been written in the C/C++ source file.
- This option is equivalent to the following property in CubeSuite+.
From the [\[Compile Options\] tab](#), [\[Schedules the instruction taking into consideration pipeline processing\]](#) in the [Optimization] category
From the [\[Library Generate Options\] tab](#), [\[Schedules the instruction taking into consideration pipeline processing\]](#) in the [Optimization] category

-map

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-map[= <file name>]
```

- [Default]

The default for this option is map when the **optimize=max** option has been specified.

[Description]

- This option optimizes accesses to global variables.
- When the **map** option is specified, a base address is set by using an external symbol-allocation information file created by the optimizing linkage editor, and a code that uses addresses relative to the base address for accesses to global or static variables is generated.
- When accesses to external variables are to be optimized by the map option, how the **map** option is used differs according to the specification of the **output** option.
- [**output=abs** or **output=mot** is specified]
Specify only **map** (not necessary when **optimize=max** is specified). Compilation and linkage are automatically performed twice, and a code in which the base address is set based on external symbol allocation information is generated.
- [**output=obj** is specified]
Compile the source file once without specifying these options, create an external symbol-allocation information file by specifying **map=<file name>** at linkage by the optimizing linkage editor, and then compile the source file again by specifying **map=<file name>** in **ccrx**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Optimizes accesses to external variables\]](#) in the [Optimization] category
 - From the [\[Library Generate Options\] tab](#), [\[Optimizes accesses to external variables\]](#) in the [Optimization] category

[Example]

- <C source file>

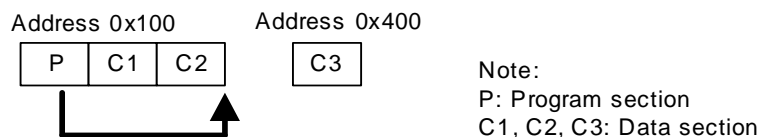
```
long A,B,C;
void func()
{
    A = 1;
    B = 2;
    C = 3;
}
```

- <Output code>

```
_func:
MOV.L    #_A,R4      ; Sets the address of A as the base address.
MOV.L    #1,[R4]
MOV.L    #2,4[R4]    ; Accesses B using the address of A as the base.
MOV.L    #3,8[R4]    ; Accesses C using the address of A as the base.
```

[Remarks]

- When the order of the definitions of global variables or static variables has been changed, a new external symbol-allocation information file must be created. If any option other than the **map** option in the previous compilation differs from the one in the current compilation, or if any contents of a function are changed, correct operation is not guaranteed. In such a case, a new external symbol-allocation information file must be created.
- This option is only valid for the compilation of C/C++ source programs. It does not apply to programs that have been compiled with the **output=src** specification or to programs written in assembly language.
- When the **map** option and **smap** option are specified simultaneously, the **map** option is valid.
- When continuous data sections are allocated after a program section, optimization of external variable accesses may be disabled or may not be performed sufficiently. For performing optimization to a maximum extent in a case in which multiple sections are allocated continuously, allocate the program section at the end. An example is shown below.



- In the above example, section **P** is allocated from address 0x100, sections **C1** and **C2** are allocated immediately after section **P**, and section **C3** is allocated from address 0x400. Since sections **C1** and **C2** are allocated continuously after section **P**, section **P** should be allocated behind section **C2**. Section **C3** is not involved because it is not allocated continuously.

-smap

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-smap

[Description]

- When the **smap** option is specified, a base address is set for global or static variables defined in the file to be compiled, and a code that uses addresses relative to the base address for accesses to those variables is generated.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Optimizes accesses to external variables\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Optimizes accesses to external variables\]](#) in the [\[Optimization\]](#) category

[Example]

- <C source file>

```
long A,B,C;
void func()
{
    A = 1;
    B = 2;
    C = 3;
}
```

- <Output code>

```
_func:
    MOV.L    #_A,R4      ; Sets the address of A as the base address.
    MOV.L    #1,[R4]
    MOV.L    #2,4[R4]    ; Accesses B using the address of A as the base.
    MOV.L    #3,8[R4]    ; Accesses C using the address of A as the base.
```

[Remarks]

- This option is only valid for the compilation of C/C++ source programs. It does not apply to programs that have been compiled with the **output=src** specification or to programs written in assembly language.
- When the **map** option and **smap** option are specified simultaneously, the **map** option is valid.

-nomap

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-nomap
```

- [Default]

The default for this option is **nomap** when the **optimize=0**, **optimize=1**, or **optimize=2** option has been specified.

[Description]

- When the **nomap** option is specified, accesses to external variables are not optimized.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Optimizes accesses to external variables\]](#) in the [Optimization] category
 - From the [\[Library Generate Options\] tab](#), [\[Optimizes accesses to external variables\]](#) in the [Optimization] category

[Example]

- <C source file>

```
long A,B,C;
void func()
{
    A = 1;
    B = 2;
    C = 3;
}
```

- <Output code>

```
_func:
    MOV.L    #__A,R4
    MOV.L    #1,[R4]
    MOV.L    #__B,R4
    MOV.L    #2,[R4]
    MOV.L    #__C,R4
    MOV.L    #3,[R4]
```

-approxdiv

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-approxdiv
```

- [Default]

When this option is omitted, division of floating-point constants into multiplications of the corresponding reciprocals as constants is not performed.

[Description]

- This option converts divisions of floating-point constants into multiplications of the corresponding reciprocals as constants.
- To be specific, when there is an expression of (variable ÷ divisor) with the divisor being a constant, a code with the expression converted into (variable × reciprocal of divisor) will be generated.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Converts floating-point constant division into multiplication\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Converts floating-point constant division into multiplication\]](#) in the [\[Optimization\]](#) category

[Remarks]

- When this option is specified, the execution performance of floating-point constant division will be improved. The precision and order of operations may, however, be changed, so take care on this point.

-enable_register

< [Compile Options](#) / [Optimize Options](#) >

[Format]

`-enable_register`

[Description]

- This option is not available in V.2.00. Any specification of this option will simply be ignored and will not lead to an error due to compatibility with former versions.

-simple_float_conv

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-simple_float_conv
```

[Description]

- This option omits part of the type conversion processing for the floating type.
- When this option is selected, the generation code that performs type conversion of the next floating-point number changes.
 - a) Type conversion from 32-bit floating type to unsigned integer type
 - b) Type conversion from unsigned integer type to 32-bit floating type
 - c) Type conversion from integer type to 64-bit floating type via 32-bit floating type
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Omits a check of the range for conversion between the floating type and unsigned integer type\]](#) in the [\[Optimization\]](#) category
 - From the [\[Library Generate Options\] tab](#), [\[Omits a check of the range for conversion between the floating type and unsigned integer type\]](#) in the [\[Optimization\]](#) category

[Example]

- < a) Type conversion from 32-bit floating type to unsigned integer type>

```
unsigned long func1(float f)
{
    return ((unsigned long)f);
}
```

When this option is not specified:

```
_func1:
        FCMP    #4F000000H,R1
        BLT     L12
        FADD    #0CF800000H,R1
L12:
        FTOI    R1,R1
        RTS
```

- < b) Type conversion from unsigned integer type to 32-bit floating type>

```
float func2(unsigned long u)
{
    return ((float)u);
}
```

When this option is not specified:

```
_func2:
    BTST    #31,R1
    BEQ     L15
    SHLR    #1,R1,R14
    AND     #1,R1
    OR      R14,R1
    ITOF    R1,R1
    FADD    R1,R1
    BRA     L16

L15:
    ITOF    R1,R1

L16:
    RTS
```

- < c) Type conversion from integer type to 64-bit floating type via 32-bit floating type>

Note Does not apply when the `dbl_size=8` specification is not valid.

```
double func3(long l)
{
    return (double)(float)l;
}
```

When this option is not specified:

```
_func3:
    ITOF    R1,R1
    BRA     __COM_CONVfd
```

When this option is specified:

```
BRA     __COM_CONV32sd
```

[Remarks]

- When this option is specified, code performance of the relevant type conversion processing is improved. The conversion result may, however, differ from C/C++ language specifications, so take care on this point.
- This option of c) is invalid when **optimize=0**.

-fpu

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-fpu

- [Default]

The default for this option is **fpu** when the Instruction-code set as the ISA ^{*1}.

The default for this option is **nofpu** (when RX200 is selected as the target CPU ^{*2}) or **fpu** (in other cases).

Note

*1) This means a selection by the **-isa** option or the ISA_RX environment variable.

*2) This means a selection by the **-cpu** option or the CPU_RX environment variable.

[Description]

- When the **fpu** option is specified, a code using FPU instructions is generated.
- This option is equivalent to the following property in CubeSuite+.
 - From the , [[Common Options](#) tab], [[Uses floating-point operation instructions](#)] in the [CPU] category

[Remarks]

- For details of the FPU instructions, refer to the RX Family Software Manual.
- When RX200 is selected as the CPU, an error will occur if fpu is specified.

-nofpu

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-nofpu

- [Default]

The default for this option is **fpu** when the Instruction-code set as the ISA ^{*1}.

The default for this option is **nofpu** (when RX200 is selected as the target CPU ^{*2}) or **fpu** (in other cases).

Note

*1) This means a selection by the **-isa** option or the ISA_RX environment variable.

*2) This means a selection by the **-cpu** option or the CPU_RX environment variable.

[Description]

- When the **nofpu** option is specified, a code not using FPU instructions is generated.
- This option is equivalent to the following property in CubeSuite+.
 - From the , [[Common Options](#) tab], [[Uses floating-point operation instructions](#)] in the [CPU] category

[Example]

-

[Remarks]

- For details of the FPU instructions, refer to the RX Family Software Manual.

-alias

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-alias = { noansi | ansi }
```

- [Default]

The default for this option is alias=noansi.

[Description]

- This option selects whether to perform optimization with consideration for the type of the data indicated by the pointer.
- When alias=ansi is specified, based on the ANSI standard, optimization considering the type of the data indicated by the pointer is performed. Although the performance of object code is generally better than when alias=noansi is specified, the results of execution may differ according to whether alias=ansi or alias=noansi is specified.
- In the same way as in V. 1.00, ANSI-standard based optimization in consideration of the type of data indicated by pointers is not performed when alias=noansi is specified.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Performs optimization considering the type of the data indicated by the pointer\]](#) in the [Optimization] category
 - From the [\[Library Generate Options\] tab](#), [\[Performs optimization considering the type of the data indicated by the pointer\]](#) in the [Optimization] category

[Example]

```
long x;
long n;
void func(short * ps)
{
    n = 1;
    *ps = 2;
    x = n;
}
```

- [When alias=noansi is specified]

Note The value of **n** is reloaded at (A) since it is regarded that there is a possibility of the value of **n** being rewritten by ***ps = 2**.

```
_func:
    MOV.L    #_n,R4
    MOV.L    #1,[R4]    ; n = 1;
    MOV.W    #2,[R1]    ; *ps = 2;
    MOV.L    [R4],R5    ; (A) n is reloaded
    MOV.L    #_x,R4
    MOV.L    R5,[R4]
    RTS
```

- [When alias=ansi is specified]

Note The value used in assignment at **n = 1** is reused at (B) because it is regarded that the value of **n** will not change at ***ps = 2** since ***ps** and **n** have different types.
(If the value of **n** is changed by ***ps = 2**, the result is also changed.)

```
_func:
    MOV.L    #_n,R4
    MOV.L    #1,[R4]    ; n = 1;
    MOV.W    #2,[R1]    ; *ps = 2;
    MOV.L    #_x,R4
    MOV.L    #1,[R4]    ; (B) Value used in assignment at n = 1 is reused
    RTS
```

[Remarks]

- When **optimize=0** or **optimize=1** is valid and the **alias** option is specified, the **alias=ansi** specification will be ignored and code will always be generated as if **alias=noansi** has been selected.

-float_order

< [Compile Options](#) / [Optimize Options](#) >

[Format]

<code>-float_order</code>

- [Default]

If this option is omitted, optimization of modification of the operation order in a floating-point expression is not performed.

[Description]

- This option is not available in V.2.00. Any specification of this option will simply be ignored and will not lead to an error due to compatibility with former versions.

-ip_optimize

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-ip_optimize
```

[Description]

- This option applies global optimization including
 - optimization that utilizes interprocedural alias analysis and
 - propagation of constant parameters and return values.

[Example]

Examples 1.

- <C source code>

```
static int func1(int *a, int *b) {
    *a=0;
    *b=1;
    return *a;
}
int x[2];
int func2() {
    return func1(x, x+1);
}
```

- <Output assembly code without ip_optimize>

```
; -optimize=2 -size
__$func1:
    MOV.L #00000000H, [R1]
    MOV.L #00000001H, [R2]
    MOV.L [R1], R1
    RTS
_func2:
    MOV.L #_x,R1
    ADD #04H, R1, R2
    BRA __$func1
```

- <Output assembly code with ip_optimize>

```
; -optimize=2 -size
__$func1:
    MOV.L #00000000H, [R1]
    MOV.L #00000001H, [R2]
    MOV.L #00000000H, R1
    RTS
_func2:
    MOV.L #_x,R1
    ADD #04H, R1, R2
    BRA __$func1
```

Examples 2.

- <C source code>

```
static int func(int x, int y, int z) {  
    return x-y+z;  
}  
int func2() {  
    return func(3,4,5);  
}
```

- <Output assembly code without ip_optimize>

```
__$func:  
    ADD R3, R1  
    SUB R2, R1  
    RTS  
_func2:  
    MOV.L #00000005H, R3  
    MOV.L #00000004H, R2  
    MOV.L #00000003H, R1  
    BRA __$func
```

- <Output assembly code with ip_optimize>

```
__$func:  
    MOV.L #00000004H, R1  
    RTS  
_func2:  
    MOV.L #00000005H, R3  
    MOV.L #00000004H, R2  
    MOV.L #00000003H, R1  
    BRA __$func
```

[Remarks]

- Inter-file optimization is also applied when this option is used with **merge_files**.

-merge_files

< [Compile Options](#) / [Optimize Options](#) >

[Format]

```
-merge_files
```

[Description]

- This option allows the compiler to compile multiple C source files and output the results to a single object file.
- The name of the object file is specified by the output option. If no name is specified, the filename will be that of the first source file plus a filename extension that corresponds to the selected output format.
- If **src** or **obj** is selected as the output format, the compiler also generates blank files that have the names of the other source files with the given filename extension attached.

[Example]

```
ccrx -merge_files -output=src=files.obj file1.c file2.c file3.c
```

files.obj is the object file. Blank files **file1.obj**, **file2.obj**, and **file3.obj** are also generated.

[Remarks]

- This option is invalid when only one source file is to be compiled or when the **output** option has been used to specify **prep** as the output format.
- Inter-file in-line expansion is applied when this option is used with the **inline** option.
- This option is not available for files to be compiled in C++ or EC++.
- The following restrictions apply to programs that include static functions or static variables.
 - If you wish to use the [Watch] window of the debugger to view a static variable that has the same name as a variable in another file, specify the variable name as well as the filename. The debugger cannot identify the variable without a filename.
 - When two or more files contain static variables with the same name and **rlink** is used to overlay sections to which the files belong, the debugger's facility to display overlay sections taking precedence over other sections is not available.
 - The names of static variables and static functions written in the link map file (.map) are those converted by the compiler (i.e., not original ones).
- Any differences (e.g. type specifier) in declarations of the same variable may lead to an error in compilation.

-whole_program

< [Compile Options](#) / [Optimize Options](#) >

[Format]

-whole_program

[Description]

- This option makes the compiler perform optimization on the assumption that all source files have been input.

[Remarks]

- When this option is specified, do not include C++ language source files among the input files.
- When this option is specified, do not specify **-lang=cpp** or **-lang=ecpp**.
- Specifying this option also makes the **ip_optimize** option effective, and if multiple source files are input, the **merge_files** option is also effective.
- When this option is specified, compilation is on the assumption that the conditions listed below are satisfied. Correct operation is not guaranteed otherwise.
 - Values and addresses of **extern** variables defined in the target source files will not be modified or referred to by other files.
 - Functions within the target source file will not be called from within other files, although functions in other files can be called from within the target source files.

[Example]

```
[wp.c]
extern void g(void);
int func(void)
{
    static int a = 0;
    a++;      // (1) Write a value to a.
    g();      // (2) Call g().
    return a; // (3) Call a.
}
```

[Without whole_program]

The compiler assumes that (2) will change the value of a since function g() may call function func(), and generates a code to read the value of a in (3).

```
_func:
    PUSH.L R6
    MOV.L  #__a$1,R6
    MOV.L  [R6],R14
    ADD    #1,R14
    MOV.L  R14,[R6]      ; (1)
    BSR    _g            ; (2)
    MOV.L  [R6],R1      ; (3)
    RTSD   #4,R6-R6
```

[With whole_program]

The compiler assumes that function g() will not call function func() and thus (2) will not change the value of a. As a result, the compiler does not read the value of a in (3) and instead generates a code to use the value written to a in (1).

```
_func:
    PUSH.L R6
    MOV.L  #__a$1,R14
    MOV.L  [R14],R6
    ADD    #1,R6
    MOV.L  R6,[R14]      ; (1)
    BSR    _g            ; (2)
    MOV.L  R6,R1         ; (3)
    RTSD   #4,R6-R6
```

Microcontroller Options

< [Compile Options / Microcontroller Options](#) >

The following microcontroller options are available.

- [-isa](#)
- [-cpu](#)
- [-endian](#)
- [-round](#)
- [-denormalize](#)
- [-dbl_size](#)
- [-int_to_short](#)
- [-signed_char](#)
- [-unsigned_char](#)
- [-signed_bitfield](#)
- [-unsigned_bitfield](#)
- [-auto_enum](#)
- [-bit_order](#)
- [-pack](#)
- [-unpack](#)
- [-exception](#)
- [-noexception](#)
- [-rtti](#)
- [-fint_register](#)
- [-branch](#)
- [-base](#)
- [-patch](#)
- [-pic](#)
- [-pid](#)
- [-nouse_pid_register](#)
- [-save_acc](#)

-isa

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-isa={ rxv1 | rxv2 }
```

- [Default]

The default for this option is determined based on the environment variable **ISA_RX**.

[Description]

- This option is used to select an instruction-set architecture (RXv1 or RXv2) for use in generating instruction codes.
- When **-isa=rxv1** is specified, an instruction code for the RXv1 instruction-set architecture is generated.
- When **-isa=rxv2** is specified, an instruction code for the RXv2 instruction-set architecture is generated.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Instruction-set architecture\]](#) in the [CPU] category

[Remarks]

- When neither the **-nofpu** nor **-fpu** option has been selected, specifying the **-isa** option automatically selects the **-fpu** option.
- Omitting the **-isa** option will lead to an error if neither the **-cpu** option nor one of the environment variables (CPU_RX or ISA_RX) is specified.
- The **-isa** and **-cpu** options cannot be specified at the same time.

-cpu

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-cpu={ rx600 | rx200 }
```

- [Default]

The default for this option is determined based on the environment variable **CPU_RX**.

[Description]

- This option specifies the microcontroller type for the instruction code to be generated.
- When **cpu=rx600** is specified, an instruction code for the RX600 Series is generated.
- When **cpu=rx200** is specified, an instruction code for the RX200 Series is generated.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Microcontroller type\]](#) in the [CPU] category

[Remarks]

- This option is for compatibility with earlier products.
- For upcoming RX-family MCUs, the isa option will be used instead of the cpu option to select an instruction-set architecture. In developing new applications, use the isa option where possible.
- The cpu option can be replaced by the -isa, -fpu and -nofpu option as follows.
 - -cpu=rx600 ==> -isa=rxv1 -fpu
 - -cpu=rx200 ==> -isa=rxv1 -nofpu
- When **cpu=rx200** is specified, the **nofpu** option is automatically selected.
- **cpu=rx200** and the **fpu** option cannot be specified at the same time.
- When **cpu=rx600** is specified while neither the **nofpu** option nor the **fpu** option has been specified, the **fpu** option is automatically selected.
- Omitting the cpu option will lead to an error if neither the -isa option nor one of the environment variables (CPU_RX or ISA_RX) is specified.
- The cpu and isa options cannot be specified at the same time.

-endian

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-endian={ big | little }
```

- [Default]

The default for this option is **endian=little**.

[Description]

- When **endian=big** is specified, data bytes are arranged in big endian.
- When **endian=little** is specified, data bytes are arranged in little endian.
- The endian type can also be specified by the **#pragma endian** extension. If both this option and a **#pragma** extension are specified, the **#pragma** specification takes priority.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Endian type for data\]](#) in the [CPU] category

-round

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

<code>-round={ zero nearest }</code>
--

- [Default]

The default for this option is **round=nearest**.

[Description]

- This option specifies the rounding method for floating-point constant operations.
- When **round=zero** is specified, values are rounded to zero.
- When **round=nearest** is specified, values are rounded to the nearest value.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\]](#) tab, [\[Rounding method for floating-point constant operations\]](#) in the [CPU] category

[Remarks]

- This option does not affect the method of rounding for floating-point operations during program execution.
- The default selection of this option does not affect the selection of the **fpu** and **nofpu** options.

-denormalize

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

<code>-denormalize={ off on }</code>
--

- [Default]

The default for this option is **denormalize=off**.

[Description]

- This option specifies the operation when denormalized numbers are used to describe floating-point constants.
- When **denormalize=off** is specified, denormalized numbers are handled as zero.
- When **denormalize=on** is specified, denormalized numbers are handled as they are.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Handling of denormalized numbers in floating-point constants\]](#) in the [CPU] category

[Remarks]

- This option does not affect the handling of denormalized numbers in floating-point operations during program execution.
- This option is not automatically enabled by the selection of the **fpu** and **nofpu** options.

-dbl_size

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

`-dbl_size={ 4 | 8 }`

- [Default]

The default for this option is **dbl_size=4**.

[Description]

- This option specifies the precision of the **double** type and **long double** type.
- When **dbl_size=4** is specified, the **double** type and **long double** type are handled as the single-precision floating type (4 bytes).
- When **dbl_size=8** is specified, the **double** type and **long double** type are handled as the double-precision floating type (8 bytes).
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Precision of the double type and long double type\]](#) in the [CPU] category

[Remarks]

- When **dbl_size=4** is selected, among the standard functions, the **mathf.h** and **math.h** functions having the same specifications as each other (e.g., **sqrtrf** and **sqrt**) are integrated to configure a standard library. Because of this, phenomena, such as the following example will occur when **dbl_size=4** is selected. When the RX simulator or emulator traces (single-step execution) the calling of **sqrtrf** which is a **mathf.h** header function, it appears as if not **sqrtrf** but **sqrt**, which is a **math.h** header function with the same specifications, has been called.

-int_to_short

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-int_to_short
```

- [Default]

Before compilation, the **int** type is not replaced with the **short** type and the **unsigned int** type is not replaced with the **unsigned short** type in the source file.

[Description]

- Before compilation, the **int** type is replaced with the **short** type and the **unsigned int** type is replaced with the **unsigned short** type in the source file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Replaces the int type with the short type\]](#) in the [CPU] category

[Remarks]

- **INT_MAX**, **INT_MIN**, and **UINT_MAX** of **limits.h** are not converted by this option.
- This option is invalid during C++ and EC++ program compilation. If an external name of a C program may be referred to by a C++, EC++ program, message W0523041 will be output for the external name.
- When the **int_to_short** option is specified and a file including a C standard header is compiled as C++ or EC++, the compiler may show the W0523041 message. In this case, simply ignore the message because it does not indicate a problem.
- Data that are shared between C and C++ (EC++) programs must be declared as the **long** or **short** type rather than as the **int** type.
- When an input function having a format such as that of **scanf** in the standard library is called while this option is enabled, be sure to pass the addresses of the variables of the long and unsigned long types as parameters for use in **%d** and **%u** conversion. If the address of the int-type or unsigned-type variables not declared as long is passed, the program might not handle related operations correctly.

-signed_char

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

-signed_char

- [Default]

The default for this option is **unsigned_char**.

[Description]

- When **signed_char** is specified, the value is handled as the **signed char** type.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Sign of the char type\]](#) in the [CPU] category

[Remarks]

- The bit-field members of the **char** type are not controlled by this option; control them using the **signed_bitfield** and **unsigned_bitfield** options.

-unsigned_char

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

`-unsigned_char`

- [Default]

The default for this option is **unsigned_char**.

[Description]

- When **unsigned_char** is specified, the value is handled as the **unsigned char** type.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Sign of the char type\]](#) in the [CPU] category

[Remarks]

- The bit-field members of the **char** type are not controlled by this option; control them using the **signed_bitfield** and **unsigned_bitfield** options.

-signed_bitfield

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

-signed_bitfield

- [Default]

When **signed_bitfield** is omitted, the value is handled as **unsigned**.

[Description]

- When **signed_bitfield** is specified, the value is handled as **signed**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Sign of the bit-field type\]](#) in the [CPU] category

-unsigned_bitfield

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

-unsigned_bitfield

- [Default]

When **unsigned_bitfield** is omitted, the value is handled as **unsigned**.

[Description]

- When **unsigned_bitfield** is specified, the value is handled as **unsigned**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Sign of the bit-field type\]](#) in the [CPU] category

-auto_enum

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-auto_enum
```

- [Default]

The default for this option is to process the enumeration type size as the **signed long** type.

[Description]

- This option processes the enumerated data qualified by **enum** as the minimum data type with which the enumeration value can fit in.
- The possible enumeration values correspond to the data types as shown in the following table.

Table B.6 Correspondences between Possible Enumeration Values and Data Types

Enumerator		Data Type
Minimum Value	Maximum Value	
-128	127	signed char
0	255	unsigned char
-32768	32767	signed short
0	65535	unsigned short
Other than above	Other than above	signed long

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Selects the enumeration type size automatically\]](#) in the [CPU] category

-bit_order

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-bit_order = { left | right }
```

- [Default]

The default for this option is **bit_order=right**.

[Description]

- This option specifies the order of bit-field members.
- When **bit_order=left** is specified, members are allocated from the upper bit.
- When **bit_order=right** is specified, members are allocated from the lower bit.
- The order of bit-field members can also be specified by the **#pragma bit_order** extension. If both this option and a **#pragma** extension are specified, the **#pragma** specification takes priority.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Order of bit-field members\]](#) in the [CPU] category

-pack

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

-pack

- [Default]

The boundary alignment value for structures and classes equals the maximum boundary alignment value for members.

[Description]

- This option specifies the boundary alignment value for structure members and class members.
- The boundary alignment value for structure members can also be specified by the **#pragma pack** extension. If both this option and a **#pragma** extension are specified, the **#pragma** specification takes priority. The boundary alignment value for structures and classes equals the maximum boundary alignment value for members.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Assumes the boundary alignment value for structure members is 1\]](#) in the [CPU] category

[Remarks]

- The boundary alignment values for structure members and class members when these options are specified are shown in the following table.

Table B.7 Boundary Alignment Values for Structure Members and Class Members When the pack Option is Specified

Member Type	pack	Not Specified
(signed) char	1	1
(unsigned) short	1	2
(unsigned) int ^{Note} , (unsigned) long, (unsigned) long long, floating type, and pointer type	1	4

Note Becomes the same as **short** when the **int_to_short** option is specified.

-unpack

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

-unpack

- [Default]

The boundary alignment value for structures and classes equals the maximum boundary alignment value for members.

[Description]

- This option specifies the boundary alignment value for structure members and class members.
- The boundary alignment value for structures and classes equals the maximum boundary alignment value for members.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Assumes the boundary alignment value for structure members is 1\]](#) in the [CPU] category

[Remarks]

- The boundary alignment values for structure members and class members when these options are specified are shown in the following table.

Table B.8 Boundary Alignment Values for Structure Members and Class Members When the unpack Option is Specified

Member Type	unpack	Not Specified
(signed) char	1	1
(unsigned) short	2	2
(unsigned) int ^{Note} , (unsigned) long, (unsigned) long long, floating type, and pointer type	4	4

Note Becomes the same as **short** when the **int_to_short** option is specified.

-exception

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-exception
```

- [Default]
The C++ exceptional handling function (**try**, **catch**, **throw**) is disabled.

[Description]

- The C++ exceptional handling function (**try**, **catch**, **throw**) is enabled.
- The code performance may be lowered.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Enables C++ exceptional handling function \(try, catch and throw\)\]](#) in the [CPU] category

[Remarks]

- In order to use the C++ exceptional handling function among files, perform the following:
 - Specify **rtti=on**.
 - Do not specify the **noprelink** option in the optimizing linkage editor.
- The **exception** option can be specified only at C++ compilation. The **exception** option is ignored when **lang=cpp** has not been specified and the input file extension is **.c** or **.p**.

-noexception

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-noexception
```

- [Default]
The C++ exceptional handling function (**try**, **catch**, **throw**) is disabled.

[Description]

- The C++ exceptional handling function (**try**, **catch**, **throw**) is disabled.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Enables C++ exceptional handling function \(try, catch and throw\)\]](#) in the [CPU] category

[Remarks]

- In order to use the C++ exceptional handling function among files, perform the following:
 - Specify **rtti=on**.
 - Do not specify the **noprelink** option in the optimizing linkage editor.
- The **noexception** option can be specified only at C++ compilation. The **noexception** option cannot be specified when **lang=cpp** has not been specified and the input file extension is **.c** or **.p**. If specified, an error will occur.

-rtti

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

`-rtti={ on | off }`

- [Default]

The default for this option is **rtti=off**.

[Description]

- This option enables or disables runtime type information.
- When **rtti=on** is specified, **dynamic_cast** and **typeid** are enabled.
- When **rtti=off** is specified, **dynamic_cast** and **typeid** are disabled.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Enables the C++ exceptional handling function \(dynamic_cast and typeid\)\]](#) in the [CPU] category

[Remarks]

- Do not define relocatable files (**.obj**) that were created by this option in a library, and do not output files in the relocatable format (**.rel**) through the optimizing linkage editor. A symbol double definition error or symbol undefined error may occur.
- **rtti=on** can be specified only at C++ compilation. **rtti=on** is ignored when **lang=cpp** has not been specified and the input file extension is **.c** or **.p**.

-fint_register

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-fint_register = { 0 | 1 | 2 | 3 | 4 }
```

- [Default]

The default for this option is **fint_register=0**.

[Description]

- This option specifies the general registers which are to be used only in fast interrupt functions (functions that have the fast interrupt setting (**fint**) in their interrupt specification defined by **#pragma interrupt**). The specified registers cannot be used in functions other than the fast interrupt functions. Since the general registers specified by this option can be used without being saved or restored in fast interrupt functions, the execution speed of fast interrupt functions will most likely be improved. Then again, since the number of usable general registers in other functions is reduced, the efficiency of register allocation in the entire program is degraded.
- The options correspond to the registers as shown in the following table.

Table B.9 Correspondences between Options and Registers

Option	Registers for Fast Interrupts Only
fint_register=0	None
fint_register=1	R13
fint_register=2	R12, R13
fint_register=3	R11, R12, R13
fint_register=4	R10, R11, R12, R13

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[General registers used only in fast interrupt functions\]](#) in the [CPU] category

[Remarks]

- Correct operation is not guaranteed when a register specified by this option is used in a function other than the fast interrupt functions. If a register specified by this option has been specified by the **base** option, an error will occur.

-branch

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

<code>-branch = { 16 24 32 }</code>

- [Default]

The default for this option is **branch=24**.

[Description]

- This option specifies the branch width.
- When **branch=16** is specified, the program is compiled with a branch width within 16 bits.
- When **branch=24** is specified, the program is compiled with a branch width within 24 bits.
- When **branch=32** is specified, the branch width is not specified.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Branch width size\]](#) in the [CPU] category

-base

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-base = { rom=<register>
         | ram=<register>
         | <address value> = <register>}
         <register>:= {R8 to R13}
```

[Description]

- This option specifies the general register used as a fixed base address throughout the program.
- When **base=rom=<register A>** is specified, accesses to **const** variables are performed relative to the specified register A. Note that the total size of the constant area section must be within 64 Kbytes to 256 Kbytes*.
- When **base=ram=<register B>** is specified, accesses to initialized variables and uninitialized variables are performed relative to the specified register B. Note that the total RAM data size must be within 64 Kbytes to 256 Kbytes*.
- When **<address value>=<register C>** is specified, accesses to an area within 64Kbytes to 256 bytes from the address value, among the areas whose addresses are already determined at the time of compilation, are performed relative to the specified register C.

Note This value is in the range from 64 to 256 Kbytes and depends on the total size of variables to be accessed.

- This option is equivalent to the following property in CubeSuite+.
- From the [\[Common Options\] tab](#), [\[Base register for ROM\]](#), [\[Base register for RAM\]](#), [\[Address value of base register that sets the address value\]](#), [\[Register of base register that sets the address value\]](#) in the [CPU] category

[Remarks]

- The same register cannot be specified for different areas.
- Only a single register can be specified for each area. If a register specified by the **fint_register** option is specified by this option, an error will occur.
- When the **pid** option is selected, **base=rom=<register>** cannot be selected. If selected, message W0523039 is output as a warning and the selection of **base=rom=<register>** is disabled.

-patch

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-patch = { rx610 }
```

[Description]

- This option is used to avoid a problem specific to the CPU type.
- When **-patch=rx610** is specified, the **MVTIPL** instruction which causes a problem in the RX610 Group is not used in the generated code. Unless **-patch=rx610** is specified, the code generated in response to the call by the intrinsic function **set_ipi** will contain the **MVTIPL** instruction.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Avoids a problem specific to the CPU type\]](#) in the [CPU] category

-pic

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

-pic

- [Default]

This option does not generate code with the program section as PIC (position independent code).

[Description]

- This option generates code with the program section as PIC (position independent code).
- In PIC, all function calls are performed with BSR or BRA instructions. When acquiring the address of a function, a relative address from the PC should be used. This allows PIC to be located at a desired address after linkage.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Enables the PIC function\]](#) in the [CPU] category

[Example]

- Calling a function (only for **branch=32**)

```
void func()
{
    sub();
}
```

```
[Without -pic]
_func:
    MOV.L    #_sub,R14
    JMP      R14

[With -pic]
_func:
    MOV.L    #_sub-L11,R14
L11:
    BRA      R14
```

- Acquiring a function address

```
void func1(void);
void (*f_ptr)(void);
void func2(void)
{
    f_ptr = func1;
}
```

```

[Without -pic]
_func2:
    MOV.L    #_f_ptr,R4
    MOV.L    #_func1,[R4]
    RTS

[With -pic]
_func2:
    MOV.L    #_f_ptr,R4
L11:
    MVFC     PC,R14
    ADD      #_func1-L11,R14
    MOV.L    R14,[R4]
    RTS

```

[Remarks]

- In C++ or EC++ compilation, the **pic** option cannot be selected. If selected, message W0523039 is output as a warning and the selection of the **pic** option is disabled.
- The address of a function which is PIC should not be used in the initialization expression used for static initialization. If used, error E0523026 will occur.
- <Example of using a PIC address for static initialization>

```

void pic_func1(void), pic_func2(int), pic_func3(int); /* Becomes PIC */
void (*fp_ptr1_for_pic) = pic_func1; /* Uses PIC address in static initialization: Error */
struct PIC_funcs{ int code; void (*fp_ptr)(int); };
struct PIC_funcs pic_funcs[] = {
    { 2, pic_func2 }, /* Uses PIC address in static initialization: Error */
    { 3, pic_func3 }, /* Uses PIC address in static initialization: Error */
};

```

- When creating a code for startup of the application program using the PIC function, refer to Application Startup, instead of Startup.
- For the PIC function, also refer to Usage of PIC/PID Function.

-pid

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

```
-pid[={ 16 | 32 }]
```

- [Default]

The constant area sections **C**, **C_2**, and **C_1**, the literal section **L**, and the **switch** statement branch table sections **W**, **W_2**, and **W_1** are not handled as PID (position independent data).

[Description]

- The constant area sections **C**, **C_2**, and **C_1**, the literal section **L**, and the **switch** statement branch table sections **W**, **W_2**, and **W_1** are handled as PID (position independent data).
- PID can be accessed through a relative address from the PID register. This allows PID to be located at a desired address after linkage.
- A single general register is used to implement the PID function.
- <PID register>
 - Based on the rules in the following table, one register from among R9 to R13 is selected according to the specification of the **fint_register** option. If the **fint_register** option is not specified, R13 is selected.

Table B.10 Correspondences between fint_register Options and PID Registers

fint_register Option	PID Register
No fint_register specification	R13
fint_register = 0	
fint_register = 1	R12
fint_register = 2	R11
fint_register = 3	R10
fint_register = 4	R9

- The PID register can be used only for the purpose of PID access.
- <Parameters>
 - The parameter selects the maximum bit width of the offset when accessing the constant area section from the PID register as 16 bits or 32 bits.
 - The default for this option when the offset width is omitted is **pid=16**. When **pid=16** is specified, the size of the constant area section that can be accessed by the PID register is limited to 64 Kbytes to 256 Kbytes (varies depending on the access width). When **pid=32** is specified, there is no limitation of the size of the constant area section that can be accessed by the PID register, but the size of the code accessing PID is increased.
 - Note that when **pid=32** and the map option with valid external symbol-allocation information are specified at the same time, the allocation information causes code the same as if **pid=16** was specified to be generated if access by the PID register is possible.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\]](#) tab, [\[Enables the PID function\]](#) in the [CPU] category

[Examples]

- Accessing an externally referenced symbol that is **const** qualified

```
extern const int pid;
int work;
void func1()
{
    work = pid;
}
```

```
[Without -pid]
_func1:
    MOV.L    #_pid,R4
    MOV.L    [R4],R5
    MOV.L    #_work,R4
    MOV.L    R5,[R4]
    RTS

[With -pid=16] (only when the PID register is R13)
_func1:
    MOV.L    __pid-__PID_TOP:16[R13],R5
    MOV.L    #_work,R4
    MOV.L    R5,[R4]
    RTS
    .glob    __PID_TOP

[With -pid=32] (only when the PID register is R13)
_func1:
    ADD      #(__pid-__PID_TOP),R13,R6
    MOV.L    [R6],R5
    MOV.L    #_work,R4
    MOV.L    R5,[R4]
    RTS
    .glob    __PID_TOP
```

- Acquiring the address of an externally defined symbol that is **const** qualified

```
extern const int pid = 1000;
const int *ptr;
void func2()
{
    ptr = &pid;
}
```

```
[Without -pid]
_func2:
    MOV.L    #_ptr,R4
    MOV.L    #_pid,[R4]
    RTS

[With -pid] (only when the PID register is R13)
_func2:
    ADD      #(__pid-__PID_TOP),R13,R5
    MOV.L    #_ptr,R4
    MOV.L    R5,[R4]
    RTS
    .glob    __PID_TOP
```

[Remarks]

- The address of an area which is PID should not be used in the initialization expression used for static initialization. If used, error E0523027 will occur.

- <Example of using a PID address for static initialization>

```
extern const int pid_data1;          /* Becomes PID */
const int *ptr1_for_pid = &pid_data1; /* Uses PID address in static initialization:
Error */
const int pid_data4[] = {1,2,3,4};  /* Becomes PID */
const int *ptr2_for_pid = pid_data4; /* Uses PID address in static initialization:
Error */
```

- When creating a code for startup of the application program using the PID function, refer to Application Startup, instead of Startup.
- When the **pid** option is selected, the same external variables in different files all have to be **const** qualified. This is because the **pid** option is used to specify **const** qualified variables as PID. The **pid** option (PID function) should not be used when there may be an external variable that is not **const** qualified.
- If the **map=<file name>** option is enabled while the **pid** option is selected, warning W0530809 may be output when there is an externally referenced variable that is not **const** qualified but used in different files as the same external variable. In the case, the displayed variable is handled as PID.
- In C++ or EC++ compilation, the **pid** option cannot be selected. If selected, message W0523039 is output as a warning and the selection of the **pid** option is disabled.
- When the **pid** option is selected, **base=rom=<register>** cannot be selected. If selected, message W0523039 is output as a warning and the selection of **base=rom=<register>** is disabled.
- If a PID register selected by the **pid** option is also specified by the **base** option, warning W0511149 will occur.
- If the **pid** option and **nouse_pid_register** option are selected simultaneously, error E0511150 will occur.
- For details of the application and PID function, refer to Usage of PIC/PID Function.

-nouse_pid_register

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

-nouse_pid_register

[Description]

- When this option is specified, the generated code does not use the PID register.
- Selection of the PID register according to the settings of the **fint_register** option is based on the same rule as for the **pid** option.
- A master program called by an application program in which the PID function is enabled needs to be compiled with this option. At this time, if the **fint_register** option is selected in the application program, the same parameter **fint_register** should also be selected in the master program.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Uses the PID register for code generation\]](#) in the [CPU] category

[Remarks]

- If the **nouse_pid_register** option and **pid** option are selected simultaneously, error E0511150 will occur.
- A register selected as the PID register also being specified for the **base** option leads to warning W0511149.
- For details of the PID function, refer to Usage of PIC/PID Function.

-save_acc

< [Compile Options](#) / [Microcontroller Options](#) >

[Format]

`-save_acc`

- [Default]

When this option is omitted, it does not generate the saved and restored code of the accumulator (**ACC,ACC0,ACC1**) for interrupt functions.

[Description]

- This option generates the saved and restored code of the accumulator (**ACC,ACC0,ACC1**) for interrupt functions.
- The save and restored code of the **ACC** when the ISA ^{*1} is selected as the RXv1 or the microcomputer type is selected by the CPU ^{*2}.
- The save and restored code of the **ACC0** and **ACC1** when the ISA ^{*1} is selected as the RXv2.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Saves and restores ACC using the interrupt function\]](#) in the [CPU] category

Note

*1) This means a selection by the **-isa** option or the ISA_RX environment variable.

*2) This means a selection by the **-cpu** option or the CPU_RX environment variable.

[Remarks]

- The generated saved and restored code is the same code generated when **acc** is selected in **#pragma interrupt**. For the actual saved and restored code, refer to the description of **acc** and **no_acc** in **#pragma interrupt** of **#pragma Extension Specifiers and Keywords**.

Assemble and Linkage Options

< [Compile Options](#) / [Assemble and Linkage Options](#) >

The following assemble and linkage options are available.

- [-asmcmd](#)
- [-lnkcmd](#)
- [-asmopt](#)
- [-lnkopt](#)

-asmcmd

< [Compile Options](#) / [Assemble and Linkage Options](#) >

[Format]

```
-asmcmd=<file name>
```

[Description]

- This option specifies the assembler options to pass to **asrx** with a subcommand file.

[Example]

```
ccrx -isa=rxv1 -asmcmd=file.sub sample.c
```

- The above description has the same meaning as the following two command lines:

```
ccrx -isa=rxv1 -output=src sample.c  
asrx -isa=rxv1 -subcommand=file.sub sample.src
```

[Remarks]

- If this option is specified for more than one time, all specified subcommand files are valid.

-lnkcmd

< [Compile Options](#) / [Assemble and Linkage Options](#) >

[Format]

```
-lnkcmd=<file name>
```

[Description]

- This option specifies the linkage options to pass to **rlink** with a subcommand file.

[Example]

```
ccrx -isa=rxv1 -output=abs=tp.abs -lnkcmd=file.sub tp1.c tp2.c
```

- The above description has the same meaning as the following three command lines:

```
ccrx -isa=rxv1 -output=src tp1.c tp2.c]
asrx -isa=rxv1 tp1.src tp2.src
rlink -subcommand=file.sub -form=abs -output=tp tp1.obj tp2.obj
```

[Remarks]

- If this option is specified for more than one time, all specified subcommand files are valid.
- Refer to the **-subcommand** option of the optimizing linkage editor for the contents of the subcommand file passed to the **-lnkcmd** option.

-asmopt

< [Compile Options](#) / [Assemble and Linkage Options](#) >

[Format]

```
-asmopt=["]<assembler option>["]
```

[Description]

- This option specifies the assembler options to pass to **asrx** with a string.
- Multiple options can be specified by enclosing them with double-quote marks (").

[Example]

```
ccrx -isa=rxv1 -asmopt="-chkpm" sample.c
```

- The above description has the same meaning as the following two command lines:

```
ccrx -isa=rxv1 -output=src sample.c  
asrx -isa=rxv1 -chkpm sample.src
```

[Remarks]

- If this option is specified for more than one time, all specified assembler options are valid.

-lnkopt

< [Compile Options](#) / [Assemble and Linkage Options](#) >

[Format]

```
-lnkopt=[" "<linkage option>[" "]
```

[Description]

- This option specifies the linkage options to pass to **rlink** with a string.
- Multiple options can be specified by enclosing them with double-quote marks (").

[Example]

```
ccrx -isa=rxv1 -output=abs=tp.abs -lnkopt="-start=P,C,D/100,B/8000" tp1.c tp2.c
```

- The above description has the same meaning as the following three command lines:

```
ccrx -isa=rxv1 -output=src tp1.c tp2.c  
asrx -isa=rxv1 tp1.src tp2.src  
rlink -start=P,C,D/100,B/8000 -form=abs -output=tp tp1.obj tp2.obj
```

[Remarks]

- If this option is specified for more than one time, all specified linkage options are valid.
- A single **-lnkopt** option can only take a single linkage option. To pass multiple linkage options, specify **-lnkopt** options as many times as the number of linkage options you require.

Other Options

< [Compile Options](#) / [Other Options](#) >

The following other options are available.

- [-logo](#)
- [-nologo](#)
- [-euc](#)
- [-sjis](#)
- [-latin1](#)
- [-utf8](#)
- [-big5](#)
- [-gb2312](#)
- [-outcode](#)
- [-subcommand](#)

-logo

< [Compile Options](#) / [Other Options](#) >

[Format]

-logo

- [Default]

The copyright notice is output.

[Description]

- The copyright notice is output.

-nologo

< [Compile Options](#) / [Other Options](#) >

[Format]

-nologo

- [Default]

The copyright notice is output.

[Description]

- When the **nologo** option is specified, output of the copyright notice is disabled.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Outputs the copyright\]](#) in the [\[Others\]](#) category

-euc

< [Compile Options](#) / [Other Options](#) >

[Format]

-euc

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **SJIS** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **euc** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Character code of an input program\]](#) in the [\[Source\]](#) category

-sjis

< [Compile Options](#) / [Other Options](#) >

[Format]

-sjis

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **SJIS** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **SJIS** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Character code of an input program\]](#) in the [\[Source\]](#) category

-latin1

< [Compile Options](#) / [Other Options](#) >

[Format]

-latin1

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **SJIS** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **latin1** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Character code of an input program\]](#) in the [Source] category

-utf8

< [Compile Options](#) / [Other Options](#) >

[Format]

-utf8

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **SJIS** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **utf8** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Character code of an input program\]](#) in the [Source] category

[Remarks]

- The **utf8** option is valid only when the **lang=c99** option has been specified.

-big5

< [Compile Options](#) / [Other Options](#) >

[Format]

-big5

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **BIG5** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **big5** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Character code of an input program\]](#) in the [Source] category

[Remarks]

- When **big5** is specified, the same character coding must be selected for the **outcode** option.

-gb2312

< [Compile Options](#) / [Other Options](#) >

[Format]

-gb2312

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **GB2312** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **gb2312** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\] tab](#), [\[Character code of an input program\]](#) in the [Source] category

[Remarks]

- When **gb2312** is specified, the same character coding must be selected for the **outcode** option.

-outcode

< [Compile Options](#) / [Other Options](#) >

[Format]

```
-outcode = { euc | sjis | latin1 | utf8 | big5 | gb2312 }
```

- [Default]

The default for this option is **outcode=sjis**.

[Description]

- This option specifies the character code to output characters in strings and character constants.
- The options correspond to the character codes as shown in the following table.

Table B.11 Correspondences between Options and Character Codes (outcode)

Option	Character Code
euc	EUC code
sjis	SJIS code
latin1	ISO-Latin1 code
utf8	UTF-8 code
big5	Big5 code
gb2312	GB2312 code

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Compile Options\]](#) tab, [\[Character code of an output assembly-language file\]](#) in the [Object] category

[Remarks]

- The **utf8** option is valid only when the **lang=c99** option has been specified.
- When **outcode=big5** or **outcode=gb2312**, the **big5** or **gb2312** option must also be specified.

-subcommand< [Compile Options](#) / [Other Options](#) >**[Format]**`-subcommand=<subcommand file name>`**[Description]**

- When the **subcommand** option is specified, the compiler options specified in a subcommand file are used at compiler startup. Specify options in a subcommand file in the same format as in the command line.

[Remarks]

- If this option is specified for more than one time, all specified subcommand files are valid.

B.1.3.2 Assembler Command Options

Classification	Option	Description
Source Options	-include	Specifies the names of folders that hold include files.
	-define	Specifies macro definitions.
	-chkpm	Checks for a privileged instruction.
	-chkfpu	Checks for a floating-point operation instruction.
	-chkdsp	Checks for a DSP instruction.
Object Options	-output	Specifies the relocatable file name.
	-debug	Debugging information is output to the object files.
	-nodebug	Debugging information is not output to the object files.
	-goptimize	Outputs additional information for inter-module optimization.
	-fpu	Generates a relocatable file which is capable of containing FPU instructions.
	-nofpu	Generates a relocatable file which is not capable of containing FPU instructions.
List Options	-listfile	An assembler list file is output.
	-nolistfile	An assembler list file is not output.
	-show	Specifies the contents of the source list file.

Classification	Option	Description
Microcontroller Options	-isa	Selects the instruction-set architecture.
	-cpu	Selects the microcontroller type.
	-endian	Selects the endian type.
	-fint_register	Selects a general register for exclusive use with the fast interrupt function.
	-base	Specifies the base registers for ROM and RAM.
	-patch	Selects avoidance or non-avoidance of a problem specific to the CPU type.
	-pic	Enables the PIC function.
	-pid	Enables the PID function.
	-nouse_pid_register	The PID register is not used in code generation.
Other Options	-logo	Selects the output of copyright information.
	-nologo	Selects the non-output of copyright information.
	-subcommand	Specifies a file for including command options.
	-euc	The character codes of input programs are interpreted as EUC codes.
	-sjis	The character codes of input programs are interpreted as SJIS codes.
	-latin1	The character codes of input programs are interpreted as ISO-Latin1 codes.
	-big5	The character codes of input programs are interpreted as BIG5 codes.
	-gb2312	The character codes of input programs are interpreted as GB2312 codes.

Source Options

< [Assembler Command Options](#) / [Source Options](#) >

The following source options are available.

- [-include](#)
- [-define](#)
- [-chkpm](#)
- [-chkfpu](#)
- [-chkdsp](#)

-include

< [Assembler Command Options](#) / [Source Options](#) >

[Format]

```
-include=<path name>[,...]
```

- [Default]

The include file is searched for in the order of the current folder and the folders specified by environment variable **INC_RXA**.

[Description]

- This option specifies the name of the path to the folder that stores the include file.
- Multiple path names can be specified by separating them with a comma (,).
- The include file is searched for in the order of the current folder, the folders specified by the **include** option, and the folders specified by environment variable **INC_RXA**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Additional include paths\]](#) and [\[System include paths\]](#) in the [\[Source\]](#) category

[Example]

- Folders **c:\usr\inc** and **c:\usr\rxc** are searched for the include file.

```
asrx -include=c:\usr\inc,c:\usr\rxc test.src
```

-define

< [Assembler Command Options](#) / [Source Options](#) >

[Format]

```
-define=<sub>[,...]  
    <sub>: <macro name> = <string>
```

[Description]

- This option replaces the macro name with the specified string.
(This provides the same function as writing the **.DEFINE** directive at the beginning of the source file.)
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Macro definition\]](#) in the [\[Source\]](#) category

[Remarks]

- **.DEFINE** takes priority over the **define** option if both are specified.

-chkpm

< [Assembler Command Options](#) / [Source Options](#) >

[Format]

-chkpm

[Description]

- This option outputs warning W0551011 when a privileged instruction is used in the source file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\]](#) tab, [\[Checks for a privileged instruction\]](#) in the [\[Others\]](#) category

[Remarks]

- For details of the privileged instructions, refer to the RX Family Software Manual.

-chkfpu

< [Assembler Command Options](#) / [Source Options](#) >

[Format]

-chkfpu

[Description]

- This option outputs warning W0551012 when a floating-point operation instruction is used in the source file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\]](#) tab, [\[Checks for a floating-point operation instruction\]](#) in the [\[Others\]](#) category

[Remarks]

- For details of the floating-point operation instructions, refer to the RX Family Software Manual.

-chkdsp

< [Assembler Command Options](#) / [Source Options](#) >

[Format]

-chkdsp

[Description]

- This option outputs warning W0551013 when a DSP instruction is used in the source file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\]](#) tab, [\[Checks for a DSP instruction\]](#) in the [\[Others\]](#) category

[Remarks]

- For details of the DSP instructions, refer to the RX Family Software Manual.

Object Options

< [Assembler Command Options](#) / [Object Options](#) >

The following object options are available.

- [-output](#)
- [-debug](#)
- [-nodebug](#)
- [-goptimize](#)
- [-fpu](#)
- [-nofpu](#)

-output

< [Assembler Command Options](#) / [Object Options](#) >

[Format]

`-output=<output file name>`

- [Default]

This option outputs a relocatable file having the same name as that of the source file with extension **.obj**.

[Description]

- When the specified output file name does not have an extension, the file name appended with extension **.obj** is used for the output relocatable file name. When it has an extension, the extension is replaced with **.obj**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\]](#) tab, [\[Path of the output folder\]](#) in the [\[Object\]](#) category

-debug

< [Assembler Command Options](#) / [Object Options](#) >

[Format]

-debug

- [Default]

If this option is not specified, no debugging information is output to the relocatable file.

[Description]

- When the **debug** option is specified, debugging information is output to the relocatable file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Outputs debugging information\]](#) in the [Object] category

-nodebug

< [Assembler Command Options](#) / [Object Options](#) >

[Format]

-nodebug

- [Default]

If this option is not specified, no debugging information is output to the relocatable file.

[Description]

- When the **nodebug** option is specified, no debugging information is output to the relocatable file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Outputs debugging information\]](#) in the [Object] category

-goptimize

< [Assembler Command Options](#) / [Object Options](#) >

[Format]

-goptimize

- [Default]

If this option is not specified, additional information for the inter-module optimization is not output.

[Description]

- This option outputs the additional information for the inter-module optimization.
- At linkage, inter-module optimization is applied to the file specified with this option.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Output additional information for inter-module optimization\]](#) in the [\[Object\]](#) category

-fpu

< [Assembler Command Options](#) / [Object Options](#) >

[Format]

-fpu

- [Default]

The default for this option is **fpu** when the Instruction-code set as the ISA ^{*1}.

The default for this option is **nofpu** (when RX200 is selected as the target CPU ^{*2}) or **fpu** (in other cases).

Note

*1) This means a selection by the **-isa** option or the ISA_RX environment variable.

*2) This means a selection by the **-cpu** option or the CPU_RX environment variable.

[Description]

- This option is used to generate a relocatable file which is capable of containing FPU instructions.
- This option is equivalent to the following property in CubeSuite+.
- From the , [[Common Options](#)] tab], [[Uses floating-point operation instructions](#)] in the [CPU] category

[Remarks]

- Specifying fpu will lead to an error when the RX200 is selected as the target CPU.

-nofpu

< [Assembler Command Options](#) / [Object Options](#) >

[Format]

-nofpu

- [Default]

The default for this option is **fpu** when the Instruction-code set as the ISA ^{*1}.

The default for this option is **nofpu** (when RX200 is selected as the target CPU ^{*2}) or **fpu** (in other cases).

Note

*1) This means a selection by the **-isa** option or the ISA_RX environment variable.

*2) This means a selection by the **-cpu** option or the CPU_RX environment variable.

[Description]

- This option is used to generate a relocatable file which is not capable of containing FPU instructions.
- This option is equivalent to the following property in CubeSuite+.
 - From the , [[Common Options](#) tab], [[Uses floating-point operation instructions](#)] in the [CPU] category

[Remarks]

- Specifying fpu will lead to an error when the RX200 is selected as the target CPU.

List Options

< [Assembler Command Options](#) / [List Options](#) >

The following list options are available.

- [-listfile](#)
- [-nolistfile](#)
- [-show](#)

-listfile

< [Assembler Command Options](#) / [List Options](#) >

[Format]

`-listfile[=<file name>]`

- [Default]

If this option is not specified, no assemble list file is output.

[Description]

- When the **listfile** option is specified, an assemble list file is output. The name of the file can also be specified.
- <file name> should be specified according to the rules described in the Naming Files section.
- If <file name> is not specified in the **listfile** option, the source file name with the extension replaced with **.lst** is used as the source list file name.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assembler Options\] tab](#), [\[Outputs a assemble list file\]](#) in the [List] category

-nolistfile

< [Assembler Command Options](#) / [List Options](#) >

[Format]

<code>-nolistfile</code>

- [Default]

If this option is not specified, no assemble list file is output.

[Description]

- When the **nolistfile** option is specified, no assemble list file is output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Outputs a assemble list file\]](#) in the [\[List\]](#) category

-show

< [Assembler Command Options](#) / [List Options](#) >

[Format]

```
-show=<sub>[,...]  
    <sub>: { conditionals | definitions | expansions }
```

[Description]

- This option specifies the contents of the list file to be output by the assembler. The following output types can be specified as <sub>.

Table B.12 Output Types Specifiable for show Option

Output Type	Description
conditionals	The statements for which the specified condition is not satisfied in conditional assembly are also output to a source list file.
definitions	The information before replacement specified by .DEFINE is output to a source list file.
expansions	The macro expansion statements are output to a source list file.

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\]](#) tab, [\[Outputs the statements unsatisfied in conditional assembly\]](#), [\[Outputs the information before .DEFINE replacement\]](#), [\[Outputs the assembler macro expansion statements\]](#) in the [\[List\]](#) category

Microcontroller Options

< [Assembler Command Options](#) / [Microcontroller Options](#) >

The following microcontroller options are available.

- [-isa](#)
- [-cpu](#)
- [-endian](#)
- [-fint_register](#)
- [-base](#)
- [-patch](#)
- [-pic](#)
- [-pid](#)
- [-nouse_pid_register](#)

-isa

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

`-isa={ rxv1 | rxv2 }`

- [Default]

The default for this option is determined based on the environment variable **ISA_RX**.

[Description]

- This option is used to select an instruction-set architecture (RXv1 or RXv2) for use in generating instruction codes.
- When **-isa=rxv1** is specified, an instruction code for the RXv1 instruction-set architecture is generated.
- When **-isa=rxv2** is specified, an instruction code for the RXv2 instruction-set architecture is generated.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Instruction-set architecture\]](#) in the [CPU] category

[Remarks]

- When neither the **-nofpu** nor **-fpu** option has been selected, specifying the **-isa** option automatically selects the **-fpu** option.
- Omitting the **-isa** option will lead to an error if neither the **-cpu** option nor one of the environment variables (CPU_RX or ISA_RX) is specified.
- The **-isa** and **-cpu** options cannot be specified at the same time.

-cpu

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

```
-cpu={ rx600 | rx200 }
```

- [Default]

The default for this option is determined based on the environment variable **CPU_RX**.

[Description]

- This option specifies the CPU type for the instruction code to be generated.
- When **-cpu=rx600** is specified, a relocatable file for the RX600 Series is generated.
- When **-cpu=rx200** is specified, a relocatable file for the RX200 Series is generated.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Microcontroller type\]](#) in the [CPU] category

[Remarks]

- This option is for compatibility with earlier products.
- For upcoming RX-family MCUs, the isa option will be used instead of the cpu option to select an instruction-set architecture. In developing new applications, use the isa option where possible.
- The cpu option can be replaced by the -isa, -fpu and -nofpu options as follows.
 - -cpu=rx600 ==> -isa=rxv1 -fpu
 - -cpu=rx200 ==> -isa=rxv1 -nofpu
- Suboptions will be added depending on the microcontroller products developed in the future.
- When **-cpu=rx200** is specified, the **-nofpu** option is automatically selected, and writing floating-point operation instructions which are not supported by the RX200 Series or writing **FPSW** in control registers will cause an error.
- **-cpu=rx200** and the **-fpu** option cannot be specified at the same time.
- When **-cpu=rx600** is specified while neither the **-nofpu** option nor the **-fpu** option has been specified, the **-fpu** option is automatically selected.
- Omitting the cpu option will lead to an error if neither the -isa option nor one of the environment variables (CPU_RX or ISA_RX) is specified.
- The -cpu and -isa options cannot be specified at the same time.

-endian

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

`-endian={ big | little }`

- [Default]

The default for this option is **endian=little**.

[Description]

- When **endian=big** is specified, data bytes are arranged in big endian.
When **endian=little** is specified, data bytes are arranged in little endian.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Endian type for data\]](#) in the [CPU] category

-fint_register

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

```
-fint_register = { 0 | 1 | 2 | 3 | 4 }
```

- [Default]

The default for this option is **fint_register=0**.

[Description]

- This option outputs to the relocatable file the information about the general registers that are specified to be used only for fast interrupts through the same-name option in the compiler.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[General registers used only in fast interrupt functions\]](#) in the [CPU] category

[Remarks]

- Be sure to set this option to the same value for all assembly processes in the project. If a different setting is made, correct operation is not guaranteed.
- Do not use a general register dedicated to fast interrupts for other purposes in assembly-language files. If such a register is used for any other purpose, correct operation is not guaranteed.
- If a register specified by this option is also specified by the **base** option, an error will be output.

-base

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

```
-base = {    rom = <register>
           |    ram = <register>
           |    <address> = <register>}
           <register> = {R8 to R13}
```

[Description]

- This option outputs to the relocatable file the information about the general register that is specified to be used only as a base address register through the same-name option in the compiler.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Base register for ROM\]](#), [\[Base register for RAM\]](#), [\[Address value of base register that sets the address value\]](#), [\[Register of base register that sets the address value\]](#) in the [CPU] category

[Remarks]

- Be sure to set this option to the same value for all assembly processes in the project. If a different setting is made, correct operation is not guaranteed.
- Do not use a general register specified by this option for other purposes than a base address register. If such a register is used for any other purpose, correct operation is not guaranteed.
- If a single general register is specified for different areas, an error will be output.
- If a general register specified by the **fint_register** option is also specified by this option, an error will be output.

-patch

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

```
-patch = { rx610 }
```

[Description]

- This option is used to avoid a problem specific to the CPU type.
- When **-patch=rx610** is specified, the **MVTIPL** instruction which causes a problem in the RX610 Group is handled as an undefined instruction. The **MVTIPL** instruction will not be recognized as an instruction and the error message E0552113 will be output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Avoids a problem specific to the CPU type\]](#) in the [CPU] category

-pic

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

-pic

- [Default]

This option generates a relocatable object indicating that code was generated with the PIC function disabled.

[Description]

- This option generates a relocatable object indicating that code was generated with the PIC function enabled.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Enables the PIC function\]](#) in the [CPU] category

[Remarks]

- Even if code conflicting with this option is written in the assembly code, it will not be checked.
- A relocatable object with the PIC function enabled cannot be linked with a relocatable object with the PIC function disabled.
- For the PIC function, also refer to Usage of PIC/PID Function.

-pid

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

```
-pid[={ 16 | 32 }]
```

- [Default]

This option generates a relocatable object indicating that code was generated with the PID function disabled.

[Description]

- This option generates a relocatable object indicating that code was generated with the PID function enabled.
- <PID register>
 - Based on the rules in the following table, one register from among R9 to R13 is selected according to the specification of the **fint_register** option. If the **fint_register** option is not specified, R13 is selected.

Table B.13 Correspondences between fint_register Options and PID Registers

fint_register Option	PID Register
No fint_register specification	R13
fint_register = 0	
fint_register = 1	R12
fint_register = 2	R11
fint_register = 3	R10
fint_register = 4	R9

- The PID register can be used only for the purpose of PID access.
- <Parameters>
 - The meaning of a parameter is the same as that for the compiler option with the same name.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Enables the PID function\]](#) in the [CPU] category

[Remarks]

- Even if code conflicting with PID is written in the assembly code, it will not be checked.
- A relocatable object with the PID function enabled cannot be linked with a relocatable object with the PID function disabled.
- If a PID register specified by the **pid** option is also specified by the **base** option, error F0553111 will be output.
- If the **pid** option and **nouse_pid_register** option are selected simultaneously, error F0553103 will be output.
- For the PID function, also refer to Usage of PIC/PID Function.

-nouse_pid_register

< [Assembler Command Options](#) / [Microcontroller Options](#) >

[Format]

`-nouse_pid_register`

[Description]

- This option generates a relocatable object that was generated without using the PID register.
- If the PID register is used in the assembly-language source file, error message E0552058 will be output. Specifying this option, however, does not lead to an error if a substitute register defined in the assembler specifications is used as the PID register.
- A master program called by an application program in which the PID function is enabled needs to be assembled with this option. At this time, if the **fint_register** option is selected in the application program, the same parameter **fint_register** should also be selected in the master program.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Uses the PID register for code generation\]](#) in the [CPU] category

[Remarks]

- If the **nouse_pid_register** option and **pid** option are selected simultaneously, error F0553103 will be output.
- If a register specified by the **nouse_pid_register** option is also specified by the **base** option, error F0553112 will be output.
- For the PID function, also refer to Usage of PIC/PID Function.

Other Options

< [Assembler Command Options](#) / [Other Options](#) >

The following other options are available.

- [-logo](#)
- [-nologo](#)
- [-subcommand](#)
- [-euc](#)
- [-sjis](#)
- [-latin1](#)
- [-big5](#)
- [-gb2312](#)

-logo

< [Assembler Command Options](#) / [Other Options](#) >

[Format]

-logo

- [Default]

The copyright notice is output.

[Description]

- The copyright notice is output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\]](#) tab, [\[Outputs the copyright\]](#) in the [\[Others\]](#) category

-nologo

< [Assembler Command Options](#) / [Other Options](#) >

[Format]

-nologo

- [Default]

The copyright notice is output.

[Description]

- When the **nologo** option is specified, output of the copyright notice is disabled.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Outputs the copyright\]](#) in the [\[Others\]](#) category

-subcommand

< [Assembler Command Options](#) / [Other Options](#) >

[Format]

```
-subcommand=<subcommand file name>
```

[Description]

- When the **subcommand** option is specified, the assembler options specified in a subcommand file are used at assembler startup. Specify options in a subcommand file in the same format as in the command line.

[Example]

- Contents of subcommand file **opt.sub**:

```
-listfile  
-debug
```

- Command line specifications:
- When options are specified in the command line as shown (1) below, the assembler interprets them as shown in (2).

```
(1) asrx -endian=big -subcommand=opt.sub test.src  
(2) asrx -endian=big -listfile -debug test.src
```

-euc

< [Assembler Command Options](#) / [Other Options](#) >

[Format]

-euc

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **sjis** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **euc** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Character code of an input program\]](#) in the [\[Source\]](#) category

-sjis

< [Assembler Command Options](#) / [Other Options](#) >

[Format]

-sjis

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **sjis** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **sjis** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Character code of an input program\]](#) in the [\[Source\]](#) category

-latin1

< [Assembler Command Options](#) / [Other Options](#) >

[Format]

-latin1

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **sjis** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **latin1** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Character code of an input program\]](#) in the [\[Source\]](#) category

-big5

< [Assembler Command Options](#) / [Other Options](#) >

[Format]

-big5

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **BIG5** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **big5** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Character code of an input program\]](#) in the [\[Source\]](#) category

-gb2312

< [Assembler Command Options](#) / [Other Options](#) >

[Format]

-gb2312

- [Default]

This option specifies the character code to handle the characters in strings, character constants, and comments in **GB2312** code.

[Description]

- This option specifies the character code to handle the characters in strings, character constants, and comments in **gb2312** code.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Assemble Options\] tab](#), [\[Character code of an input program\]](#) in the [\[Source\]](#) category

B.1.3.3 Optimizing Linkage Editor (rlink) Options

Classification	Option	Description
Input Options	-Input	Specifies relocatable files.
	-library	Specifies library files.
	-binary	Specifies binary files.
	-define	Specifies symbol definitions.
	-entry	Specifies an entry symbol or entry address.
	-noprelink	Selects non-initiation of the prelinker.
Output Options	-form	Selects the output file format.
	-debug	Debugging information is output to load module files.
	-sdebug	Debugging information is output to the .dbg file.
	-nodebug	Debugging information is not output.
	-record	Selects the record size.
	-rom	Specifies the section mapping from ROM to RAM.
	-output	Specifies the names of files to be output.
	-map	Outputs an external symbol-allocation information file.
	-space	Data are output to fill unused ranges of memory.
	-message	Information-level messages are output.
	-nomessage	The output of messages is disabled.
	-msg_unused	Messages are output to indicate the presence of externally defined symbols to which there is no reference.
	-byte_count	Specifies the number of bytes in a data record.
	-crc	Specifies the format for output of the CRC code.
	-padding	Padding data are included at the end of each section.
	-vectn	Assigns an address to the specified vector number in the variable vector table (for the RX Family and M16C Family).
	-vect	Assigns an address to an unused area in the variable vector table (for the RX Family and M16C Family).
	-jump_entries_for_pic	Outputs a jump table file (for the PIC function of the RX Family).
List Options	-list	A linkage list file is output.
	-show	Selects the contents to be output in the linkage list file.

Classification	Option	Description
Optimize Options	-optimize	Selects the items to be optimized at linkage.
	-nooptimize	Selects no optimization at linkage.
	-samesize	Specifies the minimum size for unification of the same codes.
	-symbol_forbid	Specifies symbols for which unreferenced symbol deletion is disabled.
	-samecode_forbid	Specifies symbols for which same code unification is disabled.
	-section_forbid	Specifies a section where optimization is disabled.
	-absolute_forbid	Specifies an address range where optimization is disabled.
Section Options	-start	Specifies a section start address.
	-fsymbol	Specifies the section where an external defined symbol will be placed in the output file.
	-aligned_section	Specifies the section alignment value as 16 bytes.
Verify Options	-cpu	Checks addresses for consistency.
	-contiguous_section	Specifies sections that will not be divided.
Other Options	-s9	Selects the output of an s9 record at the end of the file.
	-stack	Selects the output of a stack-usage information file.
	-compress	Debugging information are compressed.
	-nocompress	Debugging information are not compressed.
	-memory	Selects the amount of memory to be used in linkage.
	-rename	Specifies symbol names and section names to be modified.
	-delete	Specifies symbol names and module names to be deleted.
	-replace	Specifies library modules to be replaced.
	-extract	Specifies modules to be extracted from library files.
	-strip	Debugging information is deleted from absolute files and library files.
	-change_message	Specifies changes to the levels of messages (information, warning, and error).
	-hide	Name information on local symbols is deleted.
	-total_size	The total sizes of sections after linkage are sent to standard output.
Subcommand File Option	-subcommand	Specifies a file from which to include command options.
Options Other Than Above	-logo	Selects the output of copyright information.
	-nologo	Selects the non-output of copyright information.
	-end	Selects the execution of option strings specified before END .
	-exit	Specifies the end of option specification.

Input Options

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Input Options](#) >

The following input options are available.

- [-Input](#)
- [-library](#)
- [-binary](#)
- [-define](#)
- [-entry](#)
- [-noprelink](#)

-Input

< [Optimizing Linkage Editor \(rlink\) Options / Input Options](#) >

[Format]

```
-Input = <suboption>[{ , | Δ}...]
        <suboption>: <file name>[ (<module name>[,...] ) ]
```

[Description]

- Specifies an input file. Two or more files can be specified by separating them with a comma (,) or space.
- Wildcards (* or ?) can also be used for the specification. String literals specified with wildcards are expanded in alphabetical order. Expansion of numerical values precedes that of alphabetical letters. Uppercase letters are expanded before lowercase letters.
- Specifiable files are object files output from the compiler or the assembler, and relocatable or absolute files output from the optimizing linkage editor. A module in a library can be specified as an input file using the format of <library name>(<module name>). The module name is specified without an extension.
- If an extension is omitted from the input file specification, **obj** is assumed when a module name is not specified and **lib** is assumed when a module name is specified.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Input object module file\]](#) in the [Input] category
 - From the [\[Librarian Options\] tab](#), [\[Input object module file\]](#) in the [Input] category

[Examples]

```
input=a.obj lib1(e)      ; Inputs a.obj and module e in lib1.lib.
input=c*.obj             ; Inputs all .obj files beginning with c.
```

[Remarks]

- When **form=object** or **extract** is specified, this option is unavailable.
- When an input file is specified on the command line, **input** should be omitted.

-library

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Input Options](#) >

[Format]

```
-library = <file name>[,...]
```

[Description]

- Specifies an input library file. Two or more files can be specified by separating them with a comma (,).
- Wildcards (* or ?) can also be used for the specification. String literals specified with wildcards are expanded in the alphabetical order. Expansion of numerical values precedes that of alphabetical letters. Uppercase letters are expanded before lowercase letters.
- If an extension is omitted from the input file specification, **lib** is assumed.
- If **form=library** or **extract** is specified, the library file is input as the target library to be edited.
- Otherwise, after the linkage processing between files specified for the input files are executed, undefined symbols are searched in the library file.
- The symbol search in the library file is executed in the following order: user library files with the library option specification (in the specified order), the system library files with the library option specification (in the specified order), and then the default library (environment variable **HLNK_LIBRARY1,2,3**).
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Using libraries\]](#) and [\[System library file\]](#) in the [\[Input\]](#) category
 - From the [\[Librarian Options\] tab](#), [\[Using libraries\]](#) and [\[System library file\]](#) in the [\[Input\]](#) category

[Examples]

```
library=a.lib,b      ; Inputs a.lib and b.lib.  
library=c*.lib       ; Inputs all files beginning with c with the extension .lib.
```

-binary

< [Optimizing Linkage Editor \(rlink\) Options / Input Options](#) >

[Format]

```
-binary = <suboption>[,...]  
    <suboption>: <file name>(<section name>  
                [[:<boundary alignment>]]/<section attribute>][,<symbol name>])  
    <section attribute>: CODE | DATA  
    <boundary alignment>: 1 | 2 | 4 | 8 | 16 | 32 (default: 1)
```

[Description]

- Specifies an input binary file. Two or more files can be specified by separating them with a comma (,).
- If an extension is omitted for the file name specification, **bin** is assumed.
- Input binary data is allocated as the specified section data. The section address is specified with the **start** option. That section cannot be omitted.
- When a symbol is specified, the file can be linked as a defined symbol. For a variable name referenced by a C/C++ program, add an underscore (_) at the head of the reference name in the program.
- The section specified with this option can have its section attribute and boundary alignment specified.
- CODE or DATA can be specified for the section attribute.
- When section attribute specification is omitted, the write, read, and execute attributes are all enabled by default.
- A boundary alignment value can be specified for the section specified by this option. A power of 2 can be specified for the boundary alignment; no other values should be specified.
- When the boundary alignment specification is omitted, 1 is used as the default.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Input binary data file\]](#) in the [\[Input\]](#) category
 - From the [\[Librarian Options\] tab](#), [\[Input binary data file\]](#) in the [\[Input\]](#) category

[Examples]

```
input=a.obj  
start=P,D*/200  
binary=b.bin(D1bin),c.bin(D2bin:4,_datab)  
form=absolute
```

- Allocates **b.bin** from 0x200 as the **D1bin** section.
- Allocates **c.bin** after **D1bin** as the **D2bin** section (with boundary alignment = 4).
- Links **c.bin** data as the defined symbol **_datab**.

[Remarks]

- When **form={object | library}** or **strip** is specified, this option is unavailable.
- If no input object file is specified, this option cannot be specified.

-define

< [Optimizing Linkage Editor \(rlink\) Options / Input Options](#) >

[Format]

```
-define = <suboption>[,...]  
        <suboption>: <symbol name>={<symbol name> | <numerical value>}
```

[Description]

- Defines an undefined symbol forcedly as an externally defined symbol or a numerical value.
- The numerical value is specified in the hexadecimal notation. If the specified value starts with a letter from A to F, symbols are searched first, and if no corresponding symbol is found, the value is interpreted as a numerical value. Values starting with 0 are always interpreted as numerical values.
- If the specified symbol name is a C/C++ variable name, add an underscore (`_`) at the head of the definition name in the program. If the symbol name is a C++ function name (except for the **main** function), enclose the definition name with the double-quotes including parameter strings. If the parameter is **void**, specify as "<function name>()".
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Symbol definition\]](#) in the [\[Input\]](#) category

[Examples]

```
define=_sym1=data ; Defines _sym1 as the same value as the externally defined symbol data.  
define=_sym2=4000 ; Defines _sym2 as 0x4000.
```

[Remarks]

- When **form={object | relocate | library}** is specified, this option is unavailable.

-entry

< [Optimizing Linkage Editor \(rlink\) Options / Input Options](#) >

[Format]

```
-entry = {<symbol name> | <address>}
```

[Description]

- Specifies the execution start address with an externally defined symbol or address.
- The address is specified in hexadecimal notation. If the specified value starts with a letter from A to F, symbols are searched first, and if no corresponding symbol is found, the value is interpreted as an address. Values starting with 0 are always interpreted as addresses.
- For a C function name, add an underscore (_) at the head of the definition name in the program. For a C++ function name (except for the **main** function), enclose the definition name with double-quotes in the program including parameter strings. If the parameter is **void**, specify as "<function name>()".
- If the **entry** symbol is specified at compilation or assembly, this option precedes the entry symbol.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Specifies execution start address\]](#) and [\[Execution start address\]](#) in the [\[Input\]](#) category

[Examples]

```
entry=_main      ; Specifies main function in C/C++ as the execution start address.
entry="init()"    ; Specifies init function in C++ as the execution start address.
entry=100        ; Specifies 0x100 as the execution start address.
```

[Remarks]

- When **form={object | relocate | library}** or **strip** is specified, this option is unavailable.
- When optimization with undefined symbol deletion (**optimize=symbol_delete**) is specified, the execution start address should be specified. If it is not specified, the specification of the optimization with undefined symbol deletion is unavailable. Optimization with undefined symbol deletion is not available when an address is specified with this option.

-noprelink

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Input Options](#) >

[Format]

`-noprelink`

- [Default]

If this option is not specified, the prelinker is initiated.

[Description]

- Disables the prelinker initiation.
- The prelinker supports the functions to generate the C++ template instance automatically and to check types at run time. When the C++ template function and the run-time type test function are not used, specify the **noprelink** option to reduce the link time.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Initiates the prelinker\]](#) in the [Input] category
 - From the [\[Librarian Options\] tab](#), [\[Initiates the prelinker\]](#) in the [Input] category

[Remarks]

- When **extract** or **strip** is specified, this option is unavailable.
- If **form=lib** or **form=rel** is specified while the C++ template function and run-time type test are used, do not specify **noprelink**.

Output Options

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

The following output options are available.

- [-form](#)
- [-debug](#)
- [-sdebug](#)
- [-nodebug](#)
- [-record](#)
- [-rom](#)
- [-output](#)
- [-map](#)
- [-space](#)
- [-message](#)
- [-nomessage](#)
- [-msg_unused](#)
- [-byte_count](#)
- [-crc](#)
- [-padding](#)
- [-vectn](#)
- [-vect](#)
- [-jump_entries_for_pic](#)

-form

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

```
-form = {Absolute | Relocate | Object | Library[={S | U}]} | Hexadecimal | Stype | Binary}
```

- [Default]

When this option is omitted, the default is **form=absolute**.

[Description]

- Specifies the output format.
- Table B-14 lists the suboptions.

Table B.14 Suboptions of form Option

Suboption	Description
absolute	Outputs an absolute file
relocate	Outputs a relocatable file
object	Outputs an object file. This is specified when a module is extracted as an object file from a library with the extract option.
library	Outputs a library file. When library=s is specified, a system library is output. When library=u is specified, a user library is output. Default is library=u .
hexadecimal	Outputs a HEX file. For details of the HEX format, refer to HEX File Format.
stype	Outputs an S -type file. For details of the S -type format, refer to S-Type File Format.
binary	Outputs a binary file.

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\] tab](#), [\[Output file type\]](#) in the [Output File Type and Path] category
 - From the [\[Hex Output Options\] tab](#), [\[Hex file format\]](#) in the [Hex Format] category

[Remarks]

Table B-15 shows relations between output formats and input files or other options.

Table B.15 Relations Between Output Format and Input File or Other Options

Output Format	Specified Option	Enabled File Format	Specifiable Option ^{Note1}
Absolute	strip specified	Absolute file	input, output
	Other than above	Object file Relocatable file Binary file Library file	input, library, binary, debug/nodebug, sdebug, cpu, start, rom, entry, output, map, hide, optimize/nooptimize, samesize, symbol_forbid, samecode_forbid, section_forbid, absolute_forbid, compress, rename, delete, define, fsymbol, stack, noprelink, memory, msg_unused, show=symbol, reference, xreference, jump_entries_for_pic, aligned_section
Relocate	extract specified	Library file	library, output
	Other than above	Object file Relocatable file Binary file Library file	input, library, debug/nodebug, output, hide, rename, delete, noprelink, msg_unused, show=symbol, xreference
Object	extract specified	Library file	library, output
Hexadecimal Stype Binary		Object file Relocatable file Binary file Library file	input, library, binary, cpu, start, rom, entry, output, map, space, optimize/nooptimize, samesize, symbol_forbid, samecode_forbid, section_forbid, absolute_forbid, rename, delete, define, fsymbol, stack, noprelink, record, s9 ^{Note2} , byte_count ^{Note3} , memory, msg_unused, show=symbol, reference, xreference, jump_entries_for_pic, aligned_section
		Absolute file	input, output, record, s9 ^{Note2} , byte_count ^{Note3} , show=symbol, reference, xreference
Library	strip specified	Library file	library, output, memory ^{Note4} , show=symbol, section
	extract specified	Library file	library, output
	Other than above	Object file Relocatable file	input, library, output, hide, rename, delete, replace, noprelink, memory ^{Note4} , show=symbol, section

Notes 1. **message/nomessage**, **change_message**, **logo/nologo**, **form**, **list**, and **subcommand** can always be specified.

Notes 2. **s9** can be used only when **form=stype** is specified for the output format.

Notes 3. **byte_count** can be used only when **form=hexadecimal** is specified for the output format.

Notes 4. **memory** cannot be used when **hide** is specified.

-debug

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

-debug

- [Default]

When this option is omitted, debugging information is output to the output file.

[Description]

- When **debug** is specified, debugging information is output to the output file.
- If **debug** is specified and if two or more files are specified to be output with **output**, they are interpreted as **sdebug** and debugging information is output to **<first output file name>.dbg**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Outputs debugging information\]](#) in the [Output] category
 - From the [\[Librarian Options\] tab](#), [\[Outputs debugging information\]](#) in the [Output] category

[Remarks]

- When **form={object | library | hexadecimal | stype | binary}**, **strip** or **extract** is specified, this option is unavailable.

-sdebug

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

-sdebug

- [Default]

When this option is omitted, debugging information is output to the output file.

[Description]

- When **sdebug** is specified, debugging information is output to **<output file name>.dbg** file.
- If **sdebug** and **form=relocate** are specified, **sdebug** is interpreted as **debug**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Outputs debugging information\]](#) in the [Output] category

[Remarks]

- When **form={object | library | hexadecimal | stype | binary}**, **strip** or **extract** is specified, this option is unavailable.

-nodebug

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

-nodebug

- [Default]

When this option is omitted, debugging information is output to the output file.

[Description]

- When **nodebug** is specified, debugging information is not output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Outputs debugging information\]](#) in the [Output] category
 - From the [\[Librarian Options\] tab](#), [\[Outputs debugging information\]](#) in the [Output] category

[Remarks]

- When **form={object | library | hexadecimal | stype | binary}**, **strip** or **extract** is specified, this option is unavailable.

-record

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

<code>-record = { H16 H20 H32 S1 S2 S3 }</code>

- [Default]

When this option is omitted, various data records are output according to each address.

[Description]

- Outputs data with the specified data record regardless of the address range.
- If there is an address that is larger than the specified data record, the appropriate data record is selected for the address.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Hex Output Options\] tab](#), [\[Unifies the record size\]](#) in the [\[Convert Load Module File\]](#) category

[Remarks]

- This option is available only when **form=hexadecimal** or **stype** is specified.

-rom

< [Optimizing Linkage Editor \(rlink\) Options / Output Options](#) >

[Format]

```
-rom = <suboption>[,...]  
      <suboption>: <ROM section name>=<RAM section name>
```

[Description]

- Reserves ROM and RAM areas in the initialized data area and relocates a defined symbol in the ROM section with the specified address in the RAM section.
- Specifies a relocatable section including the initial value for the ROM section.
- Specifies a nonexistent section or relocatable section whose size is 0 for the RAM section.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[ROM to RAM mapped section\]](#) in the [\[Output\]](#) category

[Examples]

```
rom=D=R  
start=D/100,R/8000
```

- Reserves **R** section with the same size as **D** section and relocates defined symbols in **D** section with the **R** section addresses.

[Remarks]

- When **form={object | relocate | library}** or **strip** is specified, this option is unavailable.

-output

< [Optimizing Linkage Editor \(rlink\) Options / Output Options](#) >

[Format]

```
-output = <suboption>[,...]
        <suboption>:  <file name>[=<output range>]
        <output range>: {<start address>-<end address> | <section name>[:...]}
```

- [Default]

When this option is omitted, the default is <first input file name>.<default extension>.

The default extensions are as follows:

form=absolute: abs, form=relocate: rel, form=object: obj, form=library: lib, form=hexadecimal: hex, form=stype: mot, form=binary: bin

[Description]

- Specifies an output file name. When **form=absolute**, **hexadecimal**, **stype**, or **binary** is specified, two or more files can be specified. An address is specified in the hexadecimal notation. If the specified data starts with a letter from A to F, sections are searched first, and if no corresponding section is found, the data is interpreted as an address. Data starting with 0 are always interpreted as addresses.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Common Options\]](#) tab, [\[Output file type\]](#) in the [\[Output File Type and Path\]](#) category, [\[Output hex file\]](#), [\[Load module file convert format\]](#), [\[Output folder\]](#), [\[Output file name\]](#), [\[Division output file\]](#) in the [\[Frequency Used Options \(for Hex Output\)\]](#) category
 - From the [\[Link Options\]](#) tab, [\[Path of the output folder\]](#), [\[Output file name\]](#) in the [\[Output\]](#) category
 - From the [\[Hex Output Options\]](#) tab, [\[Output folder\]](#), [\[Output file name\]](#), [\[Division output file\]](#) in the [\[Output\]](#) category
 - From the [\[Librarian Options\]](#) tab, [\[Path of the output folder\]](#), [\[Output file name\]](#) in the [\[Output\]](#) category

[Examples]

```
output=file1.abs=0-ffff,file2.abs=10000-1ffff
```

- Outputs the range from 0 to 0xffff to **file1.abs** and the range from 0x10000 to 0x1ffff to **file2.abs**.

```
output=file1.abs=sec1:sec2,file2.abs=sec3
```

- Outputs the **sec1** and **sec2** sections to **file1.abs** and the **sec3** section to **file2.abs**.

[Remarks]

- When a file is output in section units while the CPU type is RX Family in big endian, the section size should be a multiple of 4.

-map

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

```
-map [= <file name>]
```

[Description]

- Outputs the external-symbol-allocation information file that is used by the compiler in optimizing access to external variables.
- When <file name> is not specified, the file has the name specified by the **output** option or the name of the first input file, and the extension **bls**.
- If the order of the declaration of variables in the external-symbol-allocation information file is not the same as the order of the declaration of variables found when the object was read after compilations, an error will be output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Outputs the external symbol-allocation information file\]](#) in the [Output] category

[Remarks]

- This option is valid only when **form={absolute | hexadecimal | stype | binary}** is specified.

-space

[< Optimizing Linkage Editor \(rlink\) Options / Output Options >](#)

[Format]

`-space [= {<numerical value> | Random}]`

[Description]

- Fills the unused areas in the output ranges with random values or a user-specified hexadecimal value.
- The following unused areas are filled with the value according to the output range specification in the **output** option:
 - When section names are specified for the output range:
 - The specified value is output to unused areas between the specified sections.
 - When an address range is specified for the output range:
 - The specified value is output to unused areas within the specified address range.
- A 1-, 2-, or 4-byte value can be specified. The hexadecimal value specified to the **space** option determines the output data size. If a 3-byte value is specified, the upper digit is extended with 0 to use it as a 4-byte value. If an odd number of digits are specified, the upper digits are extended with 0 to use it as an even number of digits.
- If the size of an unused area is not a multiple of the size of the specified value, the value is output as many times as possible, then a warning message is output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Hex Output Options\] tab](#), [\[Fills the unused areas in the output ranges with the value\]](#) and [\[Output padding data\]](#) in the [\[Convert Load Module File\]](#) category

[Remarks]

- When no suboption is specified by this option, unused areas are not filled with values.
- This option is available only when **form={binary | stype | hexadecimal}** is specified.
- When no output range is specified by the **output** option, this option is unavailable.

-message

< [Optimizing Linkage Editor \(rlink\) Options / Output Options](#) >

[Format]

-message

[Description]

- When **message** is specified, information-level messages are output.
- When this option is omitted, the output of information-level messages is disabled.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Enables information-level message output\]](#) in the [Output] category
 - From the [\[Librarian Options\] tab](#), [\[Enables information-level message output\]](#) in the [Output] category

-nomessage

< [Optimizing Linkage Editor \(rlink\) Options / Output Options](#) >

[Format]

```
-nomessage [=<suboption>[,...]]  
      <suboption>:  <error number>[-<error number>]
```

- [Default]

When this option is omitted, the output of information-level messages is disabled.

[Description]

- When **nomessage** is specified, the output of information-level messages is disabled. If an error number is specified, the output of the error message with the specified error number is disabled. A range of error message numbers to be disabled can be specified using a hyphen (-).
- Each error number consists of a component number (05), phase (6), and a four-digit value (e.g. 0004 in the case of M0560004). If the four-digit section has leading zeroes, e.g. before the 4 in the case of M0560004, these can be omitted.
- If a warning or error level message number is specified, the message output is disabled assuming that **change_message** has changed the specified message to the information level.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Enables information-level message output\]](#) and [\[Suppresses the number of information-level messages\]](#) in the [\[Output\]](#) category
 - From the [\[Librarian Options\] tab](#), [\[Enables information-level message output\]](#) and [\[Suppresses the number of information-level messages\]](#) in the [\[Output\]](#) category

[Examples]

- Messages of L0004, L0200 to L0203, and L1300 are disabled to be output.

```
nomessage=4,200-203,1300
```

-msg_unused

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

```
-msg_unused
```

[Description]

- Notifies the user of the externally defined symbol which is not referenced during linkage through an output message.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\]](#) tab, [\[Enables information-level message output\]](#) in the [\[Output\]](#) category
 - From the [\[Librarian Options\]](#) tab, [\[Enables information-level message output\]](#) in the [\[Output\]](#) category

[Examples]

```
rlink -msg_unused a.obj
```

[Remarks]

- When an absolute file is input, this option is invalid.
- To output a message, the **message** option must also be specified.
- The linkage editor may output a message for the function that was inline-expanded at compilation. To avoid this, add a **static** declaration for the function definition.
- In any of the following cases, references are not correctly analyzed so that information shown by output messages will be incorrect.
 - There are references to constant symbols within the same file.
 - There are branches to immediate subordinate functions when optimization is specified at compilation.

-byte_count

< [Optimizing Linkage Editor \(rlink\) Options / Output Options](#) >

[Format]

```
-byte_count=<numerical value>
```

[Description]

- Specifies the maximum byte count for a data record when a file is to be created in the **Intel-Hex** format. Specify a one-byte hexadecimal value (01 to FF) for the byte count. When this option is not specified, the linkage editor assumes FF as the maximum byte count when creating an **Intel-Hex** file.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Hex Output Options\] tab](#), [\[Specifies byte count for data record\]](#) and [\[Maximum byte count for a data record\]](#) in the [\[Hex Format\]](#) category

[Examples]

```
byte_count=10
```

[Remarks]

- This option is invalid when the file to be created is not an **Intel-Hex**-type (**form=hex**) file.

-crc

[< Optimizing Linkage Editor \(rlink\) Options / Output Options >](#)

[Format]

```
-CRc = <suboption>
      <suboption>: <address where the result is output>=<target range>
                   [ /<polynomial expression> ][ :<endian> ]
                   <address where the result is output>: <address>
                   <target range>: <start address>-<end address>[,...]
                   <polynomial expression>: { CCITT | 16 }
                   <endian>: { BIG | LITTLE }
```

[Description]

- This option is used for cyclic redundancy checking (CRC) of values from the lowest to the highest address of each target range and outputs the calculation result to the specified address.
- <endian> can be specified only when the CPU type is RX Family. When <endian> is specified, the calculation result is output to the specified address in the specified endian. When <endian> is not specified, the result is output to the specified address in the endian used in the absolute file.
- **CRC-CCITT** or **CRC-16** is selectable as a polynomial expression (default: **CRC-CCITT**).
- Polynomial expression:

```
CRC-CCITT
  X^16+X^12+X^5+1
  In bit expression: (100010000000100001)
CRC-16
  X^16+X^15+X^2+1
  In bit expression: (110000000000000101)
```

- This option is equivalent to the following property in CubeSuite+.
- From the [\[Hex Output Options\] tab](#), [\[Outputs the calculation result of CRC\]](#), [\[Output address\]](#), and [\[Target range\]](#) in the [\[Hex Format\]](#) category

[Example]

- `rlink *.obj -form=stype -start=P1,P2/1000,P3/2000 -crc=2FFE=1000-2FFD -output=out.mot=1000-2FFF`
- **crc** option: `-crc=2FFE=1000-2FFD`
 - In this example, CRC will be calculated for the range from 0x1000 to 0x2FFD and the result will be output to address 0x2FFE.
 - When the **space** option has not been specified, **space=0xFF** is assumed for calculation of free areas within the target range.
- **output** option: `-output=out.mot=1000-2FFF`
 - Since the **space** option has not been specified, the free areas are not output to the **out.mot** file. 0xFF is used in CRC for calculation of the free areas, but will not be filled into these areas.

Notes 1. The address where the result of CRC will be output cannot be included in the target range.

Notes 2. The address where the result of CRC will be output must be included in the output range specified with the **output** option.

- `rlink *.obj -form=stype -start=P1/1000,P2/1800,P3/2000`
`-space=7F -crc=2FFE=1000-17FF,2000-27FF`
`-output=out.mot=1000-2FFF`

 - **crc** option: `-crc=2FFE=1000-2FFD,2000-27FF`
 - In this example, CRC will be calculated for the two ranges, 0x1000 to 0x17FF and 0x2000 to 0x27FF, and the result will be output to address 0x2FFE.
Two or more non-contiguous address ranges can be selected as the target range for CRC.
 - **space** option: `-space=7F`
 - The value of the **space** option (0x7F) is used for CRC in free areas within the target range.
 - **output** option: `-output=out.mot=1000-2FFF`
 - Since the **space** option has been specified, the free areas are output to the **out.mot** file. 0x7F will be filled into the free areas.
- Notes 1. The order that CRC is calculated for the specified address ranges is not the order that the ranges have been specified. CRC proceeds from the lowest to the highest address.
- Notes 2. Even if you wish to use the **crc** and **space** options at the same time, the **space** option cannot be set as **random** or a value of 2 bytes or more. Only 1-byte values are valid.
-
- `rlink *.obj -form=stype -start=P1,P2/1000,P3/2000`
`-crc=1FFE=1000-1FFD,2000-2FFF`
`-output=flmem.mot=1000-1FFF`

 - **crc** option: `-crc=1FFE=1000-1FFD,2000-2FFF`
 - In this example, CRC will be calculated for the two ranges, 0x1000 to 0x1FFD and 0x2000 to 0x2FFF, and the result will be output to address 0x1FFE.
When the **space** option has not been specified, **space=0xFF** is assumed for calculation of free areas within the target range.
 - **output** option: `-output=flmem.mot=1000-1FFF`
 - Since the **space** option has not been specified, the free areas are not output to the **flmem.mot** file. 0xFF is used in CRC for calculation of the free areas, but will not be filled into these areas.

[Remarks]

- This option is invalid when two or more absolute files have been selected.
 - This option is valid only when **form={hexadecimal | stype}**.
 - When the **space** option has not been specified and the target range includes free areas that will not be output, the linkage editor assumes in CRC that 0xFF has been set in the free areas.
 - An error occurs if the target range includes an overlay area.
- Sample Code: The sample code shown below is provided to check the result of CRC figured out by the **crc** option. The sample code program should match the result of CRC by `rlink`.
- When the selected polynomial expression is **CRC-CCITT**:

```

typedef unsigned char    uint8_t;
typedef unsigned short   uint16_t;
typedef unsigned long    uint32_t;

uint16_t CRC_CCITT(uint8_t *pData, uint32_t iSize)
{
    uint32_t    ui32_i;
    uint8_t     *pui8_Data;
    uint16_t     ui16_CRC = 0xFFFFu;

    pui8_Data = (uint8_t *)pData;

    for(ui32_i = 0; ui32_i < iSize; ui32_i++)
    {
        ui16_CRC = (uint16_t)((ui16_CRC >> 8u) |
                               ((uint16_t)((uint32_t)ui16_CRC << 8u)));
        ui16_CRC ^= pui8_Data[ui32_i];
        ui16_CRC ^= (uint16_t)((ui16_CRC & 0xFFu) >> 4u);
        ui16_CRC ^= (uint16_t)((ui16_CRC << 8u) << 4u);
        ui16_CRC ^= (uint16_t)(((ui16_CRC & 0xFFu) << 4u) << 1u);
    }
    ui16_CRC = (uint16_t)( 0x0000FFFFu &
                           ((uint32_t)~(uint32_t)ui16_CRC) );

    return ui16_CRC;
}

```

- When the selected polynomial expression is **CRC-16**:

```

#define POLYNOMIAL 0xa001 // Generated polynomial expression CRC-16

typedef unsigned char    uint8_t;
typedef unsigned short   uint16_t;
typedef unsigned long    uint32_t;

uint16_t CRC16(uint8_t *pData, uint32_t iSize)
{
    uint16_t crcdData = (uint16_t)0;
    uint32_t data = 0;
    uint32_t i, cycLoop;

    for(i=0; i<iSize; i++){
        data = (uint32_t)pData[i];
        crcdData = crcdData ^ data;
        for (cycLoop = 0; cycLoop < 8; cycLoop++) {
            if (crcdData & 1) {
                crcdData = (crcdData >> 1) ^ POLYNOMIAL;
            } else {
                crcdData = crcdData >> 1;
            }
        }
    }
    return crcdData;
}

```

-padding

< [Optimizing Linkage Editor \(rlink\) Options / Output Options](#) >

[Format]

```
-padding
```

[Description]

- Fills in padding data at the end of a section so that the section size is a multiple of the boundary alignment of the section.
- The file name is <output file>.jmp.
 - From the [\[Link Options\]](#) tab, [\[Fills in padding data at the end of a section\]](#) in the [\[Output\]](#) category

[Examples]

```
-start=P,C/0 -padding
```

- When the boundary alignment of section **P** is 4 bytes, the size of section **P** is 0x06 bytes, the boundary alignment of section **C** is 1 byte, and the size of section **C** is 0x03 bytes, two bytes of padding data is filled in section **P** to make its size become 0x08 bytes and then linkage is performed.

```
-start=P/0,C/7 -padding
```

- When the boundary alignment of section **P** is 4 bytes, the size of section **P** is 0x06 bytes, the boundary alignment of section **C** is 1 byte, and the size of section **C** is 0x03 bytes, if two bytes of padding data is filled in section **P** to make its size become 0x08 bytes and then linkage is performed, error L2321 will be output because section **P** overlaps with section **C**.

[Remarks]

- The value of the created padding data is 0x00.
- Since padding is not performed to an absolute address section, the size of an absolute address section should be adjusted by the user.

-vectn

< [Optimizing Linkage Editor \(rlink\) Options / Output Options](#) >

[Format]

```
-vectn = <suboption>[,...]  
      <suboption>: <vector number> = {<symbol> | <address>}
```

[Description]

- Assigns the specified address to the specified vector number in the variable vector table section.
- When this option is specified, a variable vector table section is created and the specified address is set in the table even if there is no interrupt function in the source code.
- Specify a decimal value from 0 to 255 for <vector number>.
- Specify the external name of the target function for <symbol>.
- Specify the desired hexadecimal address for <address>.
- The file name is <output file>.jmp.
 - From the [\[Link Options\] tab](#), [\[Address setting for specified vector number\]](#) in the [\[Output\]](#) category

[Examples]

```
-vectn=30=_f1,31=0000F100 ;Specifies the _f1 address for vector  
                        ;number 30 and 0x0f100 for vector number 31
```

[Remarks]

- This option is ignored when the user creates a variable vector table section in the source program because the variable vector table is not automatically created in this case.

-vect

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

`-vect={ <symbol> | <address> }`

[Description]

- Assigns the specified address to the vector number to which no address has been assigned in the variable vector table section.
- When this option is specified, a variable vector table section is created by the linkage editor and the specified address is set in the table even if there is no interrupt function in the source code.
- Specify the external name of the target function for <symbol>.
- Specify the desired hexadecimal address for <address>.
- The file name is <output file>.jmp.
 - From the [\[Link Options\] tab](#), [\[Address setting for unused vector area\]](#) in the [\[Output\]](#) category

[Remarks]

- This option is ignored when the user creates a variable vector table section in the source program because the variable vector table is not automatically created in this case.
- When the **{<symbol>|<address>}** specification is started with 0, the whole specification is assumed as an address.

-jump_entries_for_pic

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Output Options](#) >

[Format]

```
-jump_entries_for_pic=<section name>[, ...]
```

[Description]

- Outputs an assembly-language source for a jump table to branch to external definition symbols in the specified section.
- The file name is <output file>.jmp.
 - From the [\[Link Options\]](#) tab, [\[Outputs the jump table\]](#) in the [\[Output\]](#) category

[Examples]

- A jump table for branching to external definition symbols in the sections **sct2** and **sct3** is output to **test.jmp**.

```
jump_entries_for_pic=sct2,sct3  
output=test.abs
```

- [Example of a file output to **test.jmp**]

```
;OPTIMIZING LINKAGE EDITOR GENERATED FILE 2009.07.19  
    .glob _func01  
    .glob _func02  
    .SECTION P, CODE  
_func01:  
    MOV.L #1000H, R14  
    JMP   R14  
_func02:  
    MOV.L #2000H, R14  
    JMP   R14  
    .END
```

[Remarks]

- This option is invalid when **form={object | relocate| library}** or **strip** is specified.
- The generated jump table is output to the **P** section.
- Only the program section can be specified for the type of section in the section name.

List Options

< [Optimizing Linkage Editor \(rlink\) Options](#) / [List Options](#) >

The following list options are available.

- [-list](#)
- [-show](#)

-list

< [Optimizing Linkage Editor \(rlink\) Options / List Options](#) >

[Format]

```
-list [=<file name>]
```

[Description]

- Specifies list file output and a list file name.
- If no list file name is specified, a list file with the same name as the output file (or first output file) is created, with the extension **lbp** when **form=library** or **extract** is specified, or **map** in other cases.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Outputs the linkage list file\]](#) in the [List] category
 - From the [\[Librarian Options\] tab](#), [\[Outputs the linkage list file\]](#) in the [List] category

-show

< [Optimizing Linkage Editor \(rlink\) Options / List Options](#) >

[Format]

```
-show [=<sub>[,...]]
      <sub>:{ symbol | reference | section | xreference | total_size | vector | all}
```

[Description]

- Specifies output contents of a list.
- Table B-16 lists the suboptions.
- For details of list examples, refer to Linkage List, and Library List in the user's manual.

Table B.16 Suboptions of show Option

Output Format	Suboption Name	Description
form=library or extract is specified.	symbol	Outputs a symbol name list in a module (when extract is specified).
	reference	Not specifiable.
	section	Outputs a section list in a module.
	xreference	Not specifiable.
	total_size	Not specifiable.
	vector	Not specifiable.
	all	Not specifiable (when extract is specified). Outputs a symbol name list and a section list in a module (when form=library).
Other than form=library and extract is not specified.	symbol	Outputs symbol address, size, type, and optimization contents.
	reference	Outputs the number of symbol references.
	section	Not specifiable.
	xreference	Outputs the cross-reference information.
	total_size	Shows the total sizes of sections allocated to the ROM and RAM areas.
	vector	Outputs vector information.
	all	If form=rel the linkage editor outputs the same information as when show=symbol,xreference,total_size is specified. If form=rel,data_stuff have been specified, the linkage editor outputs the same information as when show=symbol,total_size is specified. If form=abs the linkage editor outputs the same information as when show=symbol,reference,xreference,total_size is specified. If form=hex/stype/bin the linkage editor outputs the same information as when show=symbol,reference,xreference,total_size is specified. If form=obj , all is not specifiable.

- This option is equivalent to the following property in CubeSuite+.

- From the [Link Options] tab, [Outputs the linkage list file], [Outputs a symbol name list in a module], [Outputs the number of symbol references], [Outputs the cross-reference information], [Shows the total sizes of sections], [Outputs vector information] in the [List] category
- From the [Librarian Options] tab, [Outputs the linkage list file], [Outputs a symbol name list in a module], [Outputs a section list in a module], [Outputs the cross-reference information], [Shows the total sizes of sections], [Outputs vector information] in the [List] category

[Remarks]

- The following table shows whether suboptions will be valid or invalid by all possible combinations of options **form**, **show**, and/or **show=all**.

		Symbol	Reference	Section	Xreference	Vector	Total_size
form=abs	show	Valid	Valid	Invalid	Invalid	Invalid	Invalid
	show=all	Valid	Valid	Invalid	Valid	Valid	Valid
form=lib	show	Valid	Invalid	Valid	Invalid	Invalid	Invalid
	show=all	Valid	Invalid	Valid	Invalid	Invalid	Invalid
form=rel	show	Valid	Invalid	Invalid	Invalid	Invalid	Invalid
	show=all	Valid	Invalid	Invalid	Valid ^{Note}	Invalid	Valid
form=obj	show	Valid	Valid	Invalid	Invalid	Invalid	Invalid
	show=all	Valid	Invalid	Invalid	Invalid	Invalid	Invalid
form=hex/bin/sty	show	Valid	Valid	Invalid	Invalid	Invalid	Invalid
	show=all	Valid	Valid	Invalid	Valid	Valid ^{Note}	Valid ^{Note}

Note The option is invalid if an absolute-format file is input.

- Note the following limitations on output of the cross-reference information.
 - When an absolute-format file is input, the referrer address information is not output.
 - Information about references to constant symbols within the same file is not output.
 - When optimization is specified at compilation, information about branches to immediate subordinate functions is not output.
 - When optimization of access to external variables is specified, information about references to variables other than base symbols is not output.
 - Both **show=total_size** and **total_size** output the same information.
 - When show=reference is valid, the number of references of the variable specified by #pragma address is output as 0.

Optimize Options

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Optimize Options](#) >

The following optimize options are available.

- [-optimize](#)
- [-nooptimize](#)
- [-samesize](#)
- [-symbol_forbid](#)
- [-samecode_forbid](#)
- [-section_forbid](#)
- [-absolute_forbid](#)

-optimize

< [Optimizing Linkage Editor \(rlink\) Options / Optimize Options](#) >

[Format]

```
-optimize [= <suboption>[,...]]
<suboption>: { SYmbol_delete | SAME_code | SHort_format | Branch | SPeed | SAFe }
```

- [Default]

When this option is omitted, the default is **optimize**.

[Description]

- When **optimize** is specified, optimization is performed for the file specified with the **goptimize** option at compilation or assembly.
- -optimize (no suboptions) executes all optimization. It has the same meaning as -optimize=symbol_delete,same_code,short_format,branch.
- -optimize=speed executes optimizations other than those reducing object speed. It has the same meaning as -optimize=symbol_delete,short_format,branch
- -optimize=safe executes optimization other than those limited by variable or function attributes. It has the same meaning as -optimize=short_format,branch
- Other suboptions mean optimization as the following table.

Table B.17 Suboptions of optimize Option

Suboption	Description	Program to be Optimized ^{Note1}	
		RXC	RXA
symbol_delete	Deletes variables/functions that are not referenced. Always be sure to specify #pragma entry at compilation or the entry option in the optimizing linkage editor.	O	x
same_code	Creates a subroutine for the same instruction sequence.	O	x
short_format	Replaces an instruction having a displacement or an immediate value with a smaller-size instruction when the code size of the displacement or immediate value can be reduced.	O	O
branch	Optimizes branch instruction size according to program allocation information. Even if this option is not specified, it is performed when any other optimization is executed.	O	O

Notes 1. RXC: C/C++ program for RX Family,
 RXA: Assembly program for RX Family

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Optimization type\]](#), [\[Deletes variables/functions that are not referenced\]](#), [\[Creates a subroutine for the same instruction sequence\]](#), [\[Replaces an instruction with a smaller-size instruction\]](#), [\[Optimizes branch instruction size\]](#) in the [\[Optimization\]](#) category

[Remarks]

- When **form={object | relocate | library}** or **strip** is specified, this option is unavailable.
- When a start function with **#pragma entry** or **entry** is not specified, **optimize=symbol_delete** is invalid.

-nooptimize

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Optimize Options](#) >

[Format]

-nooptimize

- [Default]

When this option is omitted, the default is **optimize**.

- This option is equivalent to the following property in CubeSuite+.

- From the [\[Link Options\] tab](#), [\[Optimization type\]](#) in the [Optimization] category

[Description]

- When **pnooptimize** is specified, optimization is not performed at linkage.

-samesize

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Optimize Options](#) >

[Format]

<code>-samesize = <size></code>

- [Default]

When this option is omitted, the default is **samesize=1E**.

[Description]

- Specifies the minimum code size for the optimization with the same-code unification (**optimize=same_code**). Specify a hexadecimal value from 8 to 7FFF.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Minimum code size\]](#) in the [\[Optimization\]](#) category

[Remarks]

- When **optimize=same_code** is not specified, this option is unavailable.

-symbol_forbid

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Optimize Options](#) >

[Format]

`-symbol_forbid = <symbol name> [, ...]`

[Description]

- Disables optimization regarding unreferenced symbol deletion. For a C/C++ variable or C function name, add an underscore (_) at the head of the definition name in the program. For a C++ function, enclose the definition name in the program with double-quotes including the parameter strings. When the parameter is **void**, specify as "<function name>()".
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Unreferenced symbol that disables deletion by optimization\]](#) in the [\[Optimization\]](#) category

[Remarks]

- If optimization is not applied at linkage, this option is ignored.

-samecode_forbid

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Optimize Options](#) >

[Format]

`-samecode_forbid = <function name> [,...]`

[Description]

- Disables optimization regarding same-code unification. For a C/C++ variable or C function name, add an underscore (`_`) at the head of the definition name in the program. For a C++ function, enclose the definition name in the program with double-quotes including the parameter strings. When the parameter is **void**, specify as "`<function name>()`".
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Same-code that disables unification regarding optimization\]](#) in the [\[Optimization\]](#) category

[Remarks]

- If optimization is not applied at linkage, this option is ignored.

-section_forbid

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Optimize Options](#) >

[Format]

```
-section_forbid = <sub>[,...]  
                 <sub>: [<file name>|<module name>](<section name>[,...])
```

[Description]

- Disables optimization for the specified section. If an input file name or library module name is also specified, the optimization can be disabled for a specific file, not only the entire section.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Section to disable optimization\]](#) in the [\[Optimization\]](#) category

[Remarks]

- If optimization is not applied at linkage, this option is ignored.
- To disable optimization for an input file with its path name, type the path with the file name when specifying **section_forbid**.

-absolute_forbid

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Optimize Options](#) >

[Format]

<code>-absolute_forbid = <address> [+<size>] [, ...]</code>

[Description]

- Disables optimization regarding address + size specification.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\]](#) tab, [\[Address range to disable optimization\]](#) in the [\[Optimization\]](#) category

[Remarks]

- If optimization is not applied at linkage, this option is ignored.

Section Options

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Section Options](#) >

The following section options are available.

- [-start](#)
- [-fsymbol](#)
- [-aligned_section](#)

-start

< [Optimizing Linkage Editor \(rlink\) Options / Section Options](#) >

[Format]

```
-start = <sub> [,...]
          <sub>: [{ <section name> [{ : | , } <section name> [...] ] []] [...] [ /
<address>]
```

- [Default]

The section is allocated at 0.

[Description]

- Specifies the start address of the section. Specify an address as the hexadecimal.
- The section name can be specified with wildcards "***". Sections specified with wildcards are expanded according to the input order.
- Two or more sections can be allocated to the same address (i.e., sections are overlaid) by separating them with a colon ":".
- Sections specified at a single address are allocated in the specification order.
- Sections to be overlaid can be changed by enclosing them by parentheses "()".
- Objects in a single section are allocated in the specification order of the input file or the input library.
- If no address is specified, the section is allocated at 0.
- A section which is not specified with the **start** option is allocated after the last allocation address.
- This option is equivalent to the following property in CubeSuite+.
- From the [\[Link Options\] tab](#), [\[Section start address\]](#) in the [\[Section\]](#) category

[Examples]

This example shows how sections are allocated when the objects are input in the following order (names enclosed by parentheses are sections in the objects).

- tp1.obj(A,D1,E) -> tp2.obj(B,D3,F)) -> tp3.obj(C,D2,E,G) -> lib.lib(E)

-

- -start=A,B,E/400,C,D*:F:G/8000

0x400

A	B	E (tp1) ; E (tp3) ; E (lib)
---	---	-----------------------------

0x8000

C	D1	D3	D2
F			
G			

- Sections C, F, and G separated by colons are allocated to the same address.
- Sections specified with wildcards "*" (in this example, the sections whose names start with D) are allocated in the input order.
- Objects in the sections having the same name (E in this example) are allocated in the input order.
- An input library's section having the same name (E in this example) as those of input objects is allocated after the input objects.

- -start=A,B,C,D1:D2,D3,E,F:G/400

0x400

A	B	C	D1			
D2	D3	E			F	
G						

- The sections that come immediately after the colons (**A**, **D2**, and **G** in this example) are selected as the start and allocated to the same address.

- -start=A,B,C,(D1:D2,D3),E,(F:G)/400

0x400

A	B	C	D1		E		F	
			D2	D3			G	

- When the sections to be allocated to the same address are enclosed by parentheses, the sections within parentheses are allocated to the address immediately after the sections that come before the parentheses (C and E in this example).
- The section that comes after the parentheses (E in this example) is allocated after the last of the sections enclosed by the parentheses.

[Remarks]

- When **form={object | relocate | library}** or **strip** is specified, this option is unavailable.
- Parentheses cannot be nested.
- One or more colons must be written within parentheses. Parentheses cannot be written without a colon.
- Colons cannot be written outside of parentheses.
- When this option is specified with parentheses, optimization with the linkage editor is disabled.

-fsymbol

< [Optimizing Linkage Editor \(rlink\) Options / Section Options](#) >

[Format]

```
-fsymbol = <section name> [,...]
```

[Description]

- Outputs externally defined symbols in the specified section to a file in the assembler directive format.
- The file name is **<output file>.fsy**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[The specified section that outputs externally defined symbols to the file\]](#) in the [\[Section\]](#) category

[Examples]

- Outputs externally defined symbols in sections **sct2** and **sct3** to **test.fsy**.

```
fsymbol = sct2, sct3  
output=test.abs
```

- [Output example of **test.fsy**]

```
;RENESAS OPTIMIZING LINKER GENERATED FILE 2012.07.19  
;fsymbol = sct2, sct3  
;SECTION NAME = sct2  
  .glb_f  
_f: .equ 00000000h  
  .glb_g  
_g: .equ 00000016h  
;SECTION NAME = sct3  
  .glb_main  
_main: .equ 00000020h  
  .end
```

[Remarks]

- When **form={object | relocate | library}** or **strip** is specified, this option is unavailable.

-aligned_section

< [Optimizing Linkage Editor \(rlink\) Options / Section Options](#) >

[Format]

```
-aligned_section = <section name>[,...]
```

[Description]

- Changes the alignment value for the specified section to 16 bytes.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Section alignment\]](#) in the [\[Section\]](#) category

[Remarks]

- When **form={object | relocate | library}**, **extract**, or **strip** is specified, this option is unavailable.

Verify Options

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Verify Options](#) >

The following verify options are available.

- [-cpu](#)
- [-contiguous_section](#)

-cpu

< [Optimizing Linkage Editor \(rlink\) Options / Verify Options](#) >

[Format]

```
-cpu={ <memory type> = <address range> [,...] | STRIDE}
      <memory type>: { ROM | RAM | FIX }
      <address range>: <start address> - <end address>
```

[Description]

- When **cpu=stride** is not specified, a section larger than the specified range of addresses leads to an error.
- When **cpu=stride** is specified, a section larger than the specified range of addresses is allocated to the next area of the same memory type or the section is divided.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Checks the section larger than the specified range of addresses\]](#), [\[Address range of the memory type\]](#), [\[Allocates to the next area of the same memory type or the section is divided\]](#) in the [\[Verify\]](#) category

[Examples]

- When the **stride** suboption is not specified:

```
start=D1,D2/100
cpu=ROM=100-1FF, RAM=200-2FF
```

- The result is normal when **D1** and **D2** are respectively allocated within the ranges from 100 to 1FF and from 200 to 2FF. If they are not allocated within the ranges, an error will be output.

- When the **stride** suboption is specified:

```
start=D1,D2/100
cpu=ROM=100-1FF, RAM=200-2FF, ROM=300-3FF
cpu=stride
```

- The result is normal when **D1** and **D2** are allocated within the ROM area (regardless of whether the section is divided). A linkage error occurs when they are not allocated within the ROM area even though the section is divided.
- Specify an address range in which a section can be allocated in hexadecimal notation. The memory type attribute is used for the inter-module optimization.
- **FIX** for <memory type> is used to specify a memory area where the addresses are fixed (e.g. I/O area).
- If the address range of <start>-<end> specified for **FIX** overlaps with that specified for another memory type, the setting for **FIX** is valid.
- When <memory type> is **ROM** or **RAM** and the section size is larger than the specified memory range, suboption **STRIDE** can be used to divide a section and allocate them to another area of the same memory type. Sections are divided in module units.

```
cpu=ROM=0-FFFF, RAM=10000-1FFFF
```

- Checks that section addresses are allocated within the range from 0 to FFFF or from 10000 to 1FFFF.
- Object movement is not provided between different attributes with the inter-module optimization.

```
cpu=ROM=100-1FF,ROM=400-4FF, RAM=500-5FF  
cpu=stride
```

- When section addresses are not allocated within the range from 100 to 1FF, the linkage editor divides the sections in module units and allocates them to the range from 400 to 4FF.

[Remarks]

- When **form={object | relocate | library}** or **strip** is specified, this option is unavailable.
- When **cpu=stride** and **memory=low** are specified, this option is unavailable.
- When section **B** is divided by **cpu=stride**, the size of section **C\$BSEC** increases by 8 bytes × number of divisions because this amount of information is required for initialization.

-contiguous_section

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Verify Options](#) >

[Format]

```
-contiguous_section=<section name>[,...]
```

[Description]

- Allocates the specified section to another available area of the same memory type without dividing the section when **cpu=stride** is valid.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\]](#) tab, [\[Not divide the specified section\]](#) in the [\[Verify\]](#) category

[Examples]

```
start=P,PA,PB/100  
cpu=ROM=100-1FF,ROM=300-3FF,ROM=500-5FF  
cpu=stride  
contiguous_section=PA
```

- Section **P** is allocated to address 100.
- If section **PA** which is specified as **contiguous_section** is over address 1FF, section **PA** is allocated to address 300 without being divided.
- If section **PB** which is not specified as **contiguous_section** is over address 3FF, section **PB** is divided and allocated to address 500.

[Remarks]

- When **cpu=stride** is invalid, this option is unavailable.

Other Options

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

The following other options are available.

- [-s9](#)
- [-stack](#)
- [-compress](#)
- [-nocompress](#)
- [-memory](#)
- [-rename](#)
- [-delete](#)
- [-replace](#)
- [-extract](#)
- [-strip](#)
- [-change_message](#)
- [-hide](#)
- [-total_size](#)

-s9

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

-s9

[Description]

- Outputs the **S9** record at the end even if the entry address exceeds 0x10000.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Hex Output Options\] tab](#), [\[Outputs the S9 record at the end\]](#) in the [\[Hex Format\]](#) category

[Remarks]

- When **form=stype** is not specified, this option is unavailable.

-stack

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

-stack

[Description]

- Outputs a stack consumption information file.
- The file name is **<output file name>.sni**.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Outputs a stack use information file\]](#) in the [\[Others\]](#) category

[Remarks]

- When **form={object | relocate | library}** or **strip** is specified, this option is unavailable.

-compress

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

<code>-compress</code>

- [Default]

If this option is omitted, the debugging information is not compressed.

[Description]

- The debugging information is compressed.
- By compressing the debugging information, the debugger loading speed is improved.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Compresses the debugging information\]](#) in the [\[Others\]](#) category

[Remarks]

- When **form={object | relocate | library | hexadecimal | stype | binary}** or **strip** is specified, this option is unavailable.

-nocompress

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

<code>-nocompress</code>

- [Default]

If this option is omitted, the debugging information is not compressed.

- This option is equivalent to the following property in CubeSuite+.

- From the [\[Link Options\] tab](#), [\[Compresses the debugging information\]](#) in the [\[Others\]](#) category

[Description]

- The debugging information is not compressed.
- If the **nocompress** option is specified, the link time is reduced.

-memory

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

```
-memory = [ High | Low ]
```

- [Default]

The default for this option is **memory = high**.

[Description]

- Specifies the memory size occupied for linkage.
- When **memory = high** is specified, the processing is the same as usual.
- When **memory = low** is specified, the linkage editor loads the information necessary for linkage in smaller units to reduce the memory occupancy. This increases file accesses and processing becomes slower when the occupied memory size is less than the available memory capacity.
- **memory = low** is effective when processing is slow because a large project is linked and the memory size occupied by the linkage editor exceeds the available memory in the machine used.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\]](#) tab, [\[Reduces the memory occupancy\]](#) in the [\[Others\]](#) category
 - From the [\[Librarian Options\]](#) tab, [\[Reduces the memory occupancy\]](#) in the [\[Others\]](#) category

[Remarks]

- When one of the following options is specified, the **memory=low** option is unavailable:
- When **form=absolute**, **hexadecimal**, **stype**, or **binary** is specified:
compress, **delete**, **rename**, **map**, **stack**, **cpu=stride**, or
list and **show[={reference | xreference}]** are specified in combination.
- When **form=library** is specified:
delete, **rename**, **extract**, **hide**, or **replace**
- When **form=object** or **relocate** is specified:
extract
- Some combinations of this option and the input or output file format are unavailable.

-rename

[< Optimizing Linkage Editor \(rlink\) Options / Other Options >](#)

[Format]

```
-rename = <suboption> [,...]
        <suboption>: { [<file>] (<name> = <name> [,...])
                      | [<module>] (<name> = <name> [,...] ) }
```

[Description]

- Modifies an external symbol name or a section name.
- Symbol names or section names in a specific file or library in a module can be modified.
- For a C/C++ variable name, add an underscore (_) at the head of the definition name in the program.
- When a function name is modified, the operation is not guaranteed.
- If the specified name matches both section and symbol names, the symbol name is modified.
- If there are several files or modules of the same name, the priority depends on the input order.

[Examples]

```
rename=(_sym1=data)      ; Modifies _sym1 to data.
rename=lib1(P=P1)        ; Modifies the section P to P1
                        ; in the library module lib1.
```

[Remarks]

- When **extract** or **strip** is specified, this option is unavailable.
- When **form=absolute** is specified, the section name of the input library cannot be modified.
- Operation is not guaranteed if this option is used in combination with compile option **-merge_files**.

-delete

< [Optimizing Linkage Editor \(rlink\) Options / Other Options](#) >

[Format]

```
-delete = <suboption> [,...]  
        <suboption>: {[<file>] (<name>[,...]) | <module>}
```

[Description]

- Deletes an external symbol name or library module.
- Symbol names or modules in the specified file can be deleted.
- For a C/C++ variable name or C function name, add an underscore (_) at the head of the definition name in the program. For a C++ function name, enclose the definition name in the program with double-quotes including the parameter strings. If the parameter is **void**, specify as "<function name>()". If there are several files or modules of the same name, the file that is input first is applied.
- When a symbol is deleted using this option, the object is not deleted but the attribute is changed to the internal symbol.

[Examples]

```
delete=(_sym1)           ; Deletes the symbol _sym1 in all files.  
delete=file1.obj(_sym2)  ; Deletes the symbol _sym2 in the file file1.obj.
```

[Remarks]

- When **extract** or **strip** is specified, this option is unavailable.
- When **form=library** has been specified, this option deletes modules.
- When **form={absolute|relocate|hexadecimal|stype|binary}** has been specified, this option deletes external symbols.
- Operation is not guaranteed if this option is used in combination with compile option **-merge_files**.

-replace

[< Optimizing Linkage Editor \(rlink\) Options / Other Options >](#)

[Format]

```
-replace = <suboption> [,...]
          <suboption>: <file name> [ ( <module name> [,...] ) ] }
```

[Description]

- Replaces library modules.
- Replaces the specified file or library module with the module of the same name in the library specified with the **library** option.

[Examples]

```
replace=file1.obj          ; Replaces the module file1 with the module file1.obj.
replace=lib1.lib(md11)     ; Replaces the module md11 with the module md11
                           ; in the input library file lib1.lib.
```

[Remarks]

- When **form={object | relocate | absolute | hexadecimal | stype | binary}**, **extract**, or **strip** is specified, this option is unavailable.
- Operation is not guaranteed if this option is used in combination with compile option **-merge_files**.

-extract

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

```
-extract = <module name> [,...]
```

[Description]

- Extracts library modules.
- Extracts the specified library module from the library file specified using the **library** option.

[Examples]

```
extract=file1 ; Extracts the module file1.
```

[Remarks]

- When **form={absolute | hexadecimal | stype | binary}** or **strip** is specified, this option is unavailable.
- When **form=library** has been specified, this option deletes modules.
- When **form={absolute|relocate|hexadecimal|stype|binary}** has been specified, this option deletes external symbols.

-strip

[< Optimizing Linkage Editor \(rlink\) Options / Other Options >](#)

[Format]

```
-strip
```

[Description]

- Deletes debugging information in an absolute file or library file.
- When the **strip** option is specified, one input file should correspond to one output file.

[Examples]

```
input=file1.abs file2.abs file3.abs  
strip
```

- Deletes debugging information of **file1.abs**, **file2.abs**, and **file3.abs**, and outputs this information to **file1.abs**, **file2.abs**, and **file3.abs**, respectively. Files before debugging information is deleted are backed up in **file1.abk**, **file2.abk**, and **file3.abk**.

[Remarks]

- When **form={object | relocate | hexadecimal | stype | binary}** is specified, this option is unavailable.

-change_message

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

```
-change_message = <suboption> [,...]
                  <suboption>: <error level> [= <error number> [-<error number>] [,...]
]
                  <error level>: {Information | Warning | Error}
```

[Description]

- Modifies the level of information, warning, and error messages.
- Specifies the execution continuation or abort at the message output.
- When a message number is specified, the error level of the message with the specified error number changes to the given level.
- A range of error message numbers can be specified by using a hyphen (-).
- Each error number must consist of a component number (05), phase (6), and a four-digit value (e.g. 2310 in the case of E0562310).
- If no error number is specified, all messages will be changed to the specified level.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\]](#) tab, [\[Changes the warning-level messages to information-level messages\]](#), [\[Error number of warning-level message\]](#), [\[Changes the information-level messages to warning-level messages\]](#), [\[Error number of information-level message\]](#), [\[Changes the information-level and warning-level messages to error-level messages\]](#), [\[Error number of information-level and warning-level message\]](#) in the [\[Others\]](#) category
 - From the [\[Librarian Options\]](#) tab, [\[Changes the warning-level messages to information-level messages\]](#), [\[Error number of warning-level message\]](#), [\[Changes the information-level messages to warning-level messages\]](#), [\[Error number of information-level message\]](#), [\[Changes the information-level and warning-level messages to error-level messages\]](#), [\[Error number of information-level and warning-level message\]](#) in the [\[Others\]](#) category

[Examples]

```
change_message=warning=2310
```

- This changes E0562310 to a warning-level message so that linkage proceeds even if E0562310 is output.

```
change_message=error
```

- This changes all information and warning messages to error level messages.
When a message is output, the execution is aborted.

-hide

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

```
-hide
```

[Description]

- Deletes local symbol name information from the output file. Since all the name information regarding local symbols is deleted, local symbol names cannot be checked even if the file is opened with a binary editor. This option does not affect the operation of the generated file.
- Use this option to keep the local symbol names secret.
- The following types of symbol names are hidden:
 - C source: Variable or function names specified with the **static** qualifiers
 - C source: Label names for the **goto** statements
 - Assembly source: Symbol names of which external definition (reference) symbols are not declared

Note The entry function name is not hidden.

- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\]](#) tab, [\[Deletes local symbol name information\]](#) in the [\[Others\]](#) category
 - From the [\[Librarian Options\]](#) tab, [\[Deletes local symbol name information\]](#) in the [\[Others\]](#) category

[Examples]

- The following is a C source example in which this option is valid:

```
int g1;
int g2=1;
const int g3=3;
static int s1;           //<- The static variable name will be hidden.
static int s2=1;         //<- The static variable name will be hidden.
static const int s3=2;    //<- The static variable name will be hidden.

static int sub1()         //<- The static function name will be hidden.
{
    static int s1;        //<- The static variable name will be hidden.
    int l1;
    s1 = l1; l1 = s1;
    return(l1);
}

int main()
{
    sub1();
    if (g1==1)
        goto L1;
    g2=2;
L1:                      //<- The label name of the goto statement will be hidden.
    return(0);
}
```

[Remarks]

- This option is available only when the output file format is specified as **absolute**, **relocate**, or **library**.
- When the input file was compiled or assembled with the **goptimize** option specified, this option is unavailable if the output file format is specified as **relocate** or **library**.
- To use this option with the external variable access optimization, do not use this option for the first linkage, and use it only for the second linkage.
- The symbol names in the debugging information are not deleted by this option.

-total_size

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Other Options](#) >

[Format]

`-total_size`

[Description]

- Sends total sizes of sections after linkage to standard output. The sections are categorized as follows, with the overall size of each being output.
- Executable program sections
- Non-program sections allocated to the ROM area
- Sections allocated to the RAM area
- This option makes it easy to see the total sizes of sections allocated to the ROM and RAM areas.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Displays the total sizes of sections\]](#) in the [\[Others\]](#) category
 - From the [\[Librarian Options\] tab](#), [\[Displays the total sizes of sections\]](#) in the [\[Others\]](#) category

[Remarks]

- The **show=total_size** option must be used if total sizes of sections are to be output in the linkage listing.
- When the ROM-support function (**rom** option) has been specified for a section, the section will be used by both the source (ROM) and destination (RAM) of the transfer. The sizes of sections of this type will be added to the total sizes of sections in both ROM and RAM.

Subcommand File Option

< [Optimizing Linkage Editor \(rlink\) Options / Subcommand File Option](#) >

The following subcommand file option is available.

- [-subcommand](#)

-subcommand

< [Optimizing Linkage Editor \(rlink\) Options / Subcommand File Option](#) >

[Format]

```
-subcommand = <file name>
```

[Description]

- Specifies options with a subcommand file.
- The format of the subcommand file is as follows:
<option> { = | Δ } [<suboption> [...]] [Δ&] [;<comment>]
- The option and suboption are separated by an "=" sign or a space.
- For the **input** option, suboptions are separated by a space.
- One option is specified per line in the subcommand file.
- If a subcommand description exceeds one line, the description can be allowed to overflow to the next line by using an ampersand (&).
- The **subcommand** option cannot be specified in the subcommand file.

[Examples]

- Command line specification:

```
rlink file1.obj -sub=test.sub file4.obj
```

- Subcommand specification:

```
input    file2.obj file3.obj      ; This is a comment.  
library lib1.lib, &              ; Specifies line continued.  
lib2.lib
```

- Option contents specified with a subcommand file are expanded to the location at which the subcommand is specified on the command line and are executed.
- The order of file input is **file1.obj**, **file2.obj**, **file3.obj**, and **file4.obj**.

Options Other Than Above

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Options Other Than Above](#) >

The following options other than above are available.

- [-logo](#)
- [-nologo](#)
- [-end](#)
- [-exit](#)

-logo

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Options Other Than Above](#) >

[Format]

-logo

- \[Default]

When this option is omitted, the copyright notice is output.

[Description]

- The copyright notice is output.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Displays the copyright information\]](#) in the [\[Others\]](#) category
 - From the [\[Librarian Options\] tab](#), [\[Displays the copyright information\]](#) in the [\[Others\]](#) category

-nologo

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Options Other Than Above](#) >

[Format]

-nologo

- [Default]

When this option is omitted, the copyright notice is output.

[Description]

- Output of the copyright notice is disabled.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Link Options\] tab](#), [\[Displays the copyright information\]](#) in the [\[Others\]](#) category
 - From the [\[Librarian Options\] tab](#), [\[Displays the copyright information\]](#) in the [\[Others\]](#) category

-end

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Options Other Than Above](#) >

[Format]

```
-end
```

[Description]

- Executes option strings specified before **END**. After the linkage processing is terminated, option strings that are specified after **END** are input and the linkage processing is continued.
- This option cannot be specified on the command line.

[Examples]

```
input=a.obj,b.obj      ; Processing (1)
start=P,C,D/100,B/8000 ; Processing (2)
output=a.abs           ; Processing (3)
end
input=a.abs            ; Processing (4)
form=stype             ; Processing (5)
output=a.mot           ; Processing (6)
```

- Executes the processing from (1) to (3) and outputs **a.abs**. Then executes the processing from (4) to (6) and outputs **a.mot**.

-exit

< [Optimizing Linkage Editor \(rlink\) Options](#) / [Options Other Than Above](#) >

[Format]

```
-exit
```

[Description]

- Specifies the end of the option specifications.
- This option cannot be specified on the command line.

[Examples]

- Command line specification:

```
rlink -sub=test.sub -nodebug
```

- test.sub:

```
input=a.obj,b.obj          ; Processing (1)
start=P,C,D/100,B/8000     ; Processing (2)
output=a.abs               ; Processing (3)
exit
```

- Executes the processing from (1) to (3) and outputs **a.abs**.
- The **nodebug** option specified on the command line after **exit** is executed is ignored.

B.1.3.4 Library Generator Options

Classification	Option	Description
Library Options	-head	Specifies a configuration library.
	-output	Specifies an output library file name.
	-nofloat	Creates a simple I/O function.
	-lang	Selects the set of functions available from the C standard library.
	-simple_stdio	Creates a functionally cut down version of the set of I/O functions.
	-logo -nologo	Outputs the copyright. Disables output of the copyright.

Library Options

< [Library Generator Options](#) / [Library Options](#) >

The following library options are available.

- [-head](#)
- [-output](#)
- [-nofloat](#)
- [-lang](#)
- [-simple_stdio](#)
- [-logo](#)
- [-nologo](#)

-head

< [Library Generator Options](#) / [Library Options](#) >

[Format]

```
-head=<sub>[,...]
<sub>:{ all | runtime | ctype | math | mathf | stdarg | stdio | stdlib | string | ios |
      new | complex | cppstring | c99_complex | fenv | inttypes | wchar | wctype}
```

- [Default]

The default for this option is **head=all**.

[Description]

- This option specifies a configuration file with a header file name.
- When **head=all** is specified, all header file names will be configured.
- The runtime library is always configured.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Library Generate Options\] tab](#), [\[Configuration library\]](#), [\[Enables runtime library\]](#), [\[Enables ctype.h\(C89/C99\)\]](#), [\[Enables math.h\(C89/C99\)\]](#), [\[Enables mathf.h\(C89/C99\)\]](#), [\[Enables stdarg.h\(C89/C99\)\]](#), [\[Enables stdio.h\(C89/C99\)\]](#), [\[Enables stdlib.h\(C89/C99\)\]](#), [\[Enables string.h\(C89/C99\)\]](#), [\[Enables ios\(EC++\)\]](#), [\[Enables new\(EC++\)\]](#), [\[Enables complex\(EC++\)\]](#), [\[Enables string\(EC++\)\]](#), [\[Enables complex.h\(C99\)\]](#), [\[Enables fenv.h\(C99\)\]](#), [\[Enables inttypes.h\(C99\)\]](#), [\[Enables wchar.h\(C99\)\]](#), and [\[Enables wctype.h\(C99\)\]](#) in the [\[Standard Library\]](#) category

-output

< [Library Generator Options](#) / [Library Options](#) >

[Format]

`-output=<file name>`

- [Default]

The default for this option is **output=stdlib.lib**.

[Description]

- This option specifies an output file name.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Library Generate Options\] tab](#), [\[Path of the output folder\]](#) and [\[Output file name\]](#) in the [\[Object\] category](#)

-nofloat

< [Library Generator Options](#) / [Library Options](#) >

[Format]

-nofloat

[Description]

- This option creates simple I/O functions that do not support the conversion of floating-point numbers (%f, %e, %E, %g, %G).
- When inputting or outputting files that do not require the conversion of floating-point numbers, ROM can be saved. Target functions: fprintf, fscanf, printf, scanf, sprintf, sscanf, vfprintf, vprintf, and vsprintf
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Library Generate Options\] tab](#), [\[Generation mode of the standard library\]](#) in the [Object] category

[Remarks]

- In a library created with this option specified, correct operation cannot be guaranteed when floating-point numbers are input to or output from the target functions.

-lang

< [Library Generator Options](#) / [Library Options](#) >

[Format]

`-lang = { c | c99 }`

- [Default]

The default for this option is **lang=c**.

[Description]

- This option selects which functions are to be usable in the C standard library.
- When **lang=c** is specified, only the functions conforming to the **C89** standard are included in the C standard library, and the extended functions of the **C99** standard are not included. When **lang=c99** is specified, the functions conforming to the **C89** standard and the functions conforming to the **C99** standard are included in the C standard library.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Library Generate Options\] tab](#), [\[Library configuration\]](#) in the [Standard Library] category

[Remarks]

- There are no changes in the functions included in the C++ and EC++ standard libraries.
- When **lang=c99** is specified, all functions including those specified by the **C99** standard can be used. Since the number of available functions is greater than when **lang=c** is specified, however, generating a library may take a long time.

-simple_stdio

< [Library Generator Options](#) / [Library Options](#) >

[Format]

`-simple_stdio`

[Description]

- This option creates a functional cutdown version of I/O functions.
- The functional cutdown version does not include the conversion of floating-point numbers (same as the function not supported with the **nofloat** option), the conversion of **long long** type, and the conversion of 2-byte code. When inputting or outputting files that do not require these functions, ROM can be saved.
Target functions: `fprintf`, `fscanf`, `printf`, `scanf`, `sprintf`, `sscanf`, `vfprintf`, `vprintf`, and `vsprintf`
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Library Generate Options\] tab](#), [\[Generation mode of the standard library\]](#) in the [\[Object\]](#) category

[Remarks]

- In a library created with this option specified, correct operation cannot be guaranteed when a cutdown function is used in the target functions.
- This function is disabled during C++ and EC++ program compilation.

-logo

< [Library Generator Options](#) / [Library Options](#) >

[Format]

-logo

- [Default]

When this option is omitted, the copyright notice is output.

[Description]

- The copyright notice is output.

-nologo

< [Library Generator Options](#) / [Library Options](#) >

[Format]

-nologo

- [Default]

When this option is omitted, the copyright notice is output.

[Description]

- Output of the copyright notice is disabled.
- This option is equivalent to the following property in CubeSuite+.
 - From the [\[Library Generate Options\] tab](#), [\[Outputs the copyright\]](#) in the [\[Others\]](#) category

Compiler Options That Become Invalid

In addition to the options in [B.1.3.4 Library Generator Options](#), the C/C++ compiler options can be specified in the library generator as options used for library compilation. However, the options listed below are invalid; they are not selected at library compilation.

Table B.18 Invalid Options

No.	Options that Become Invalid	Conditions for Invalidation	Library Configuration When Made Invalid
1	lang	Always invalid	None
2	include	Always invalid	None
3	define	Always invalid	None
4	undefined	Always invalid	None
5	message nomessage	Always invalid	nomessage
6	change_message	Always invalid	None
7	file_inline_path	Always invalid	None
8	comment	Always invalid	None
9	check	Always invalid	None
10	output	Always invalid	output=obj
11	noline	Always invalid	None
12	debug nodebug	Always invalid	nodebug
13	listfile nolistfile show	Always invalid	nolistfile
14	file_inline	Always invalid	None
15	asmcmd	Always invalid	None
16	Inkcmd	Always invalid	None
17	asmopt	Always invalid	None
18	Inkopt	Always invalid	None
19	logo nologo	Always invalid	nologo
20	euc sjis latin1 utf8	Always invalid	None
21	outcode	Always invalid	None
22	subcommand	Always invalid	None
23	alias	Always invalid	alias=noansi
24	pic pid	lang=c++ or at C++ source compilation ^{Note1}	None

No.	Options that Become Invalid	Conditions for Invalidation	Library Configuration When Made Invalid
25	ip_optimize	Always invalid	None
26	merge_files	Always invalid	None
27	whole_program	Always invalid	None
28	big5 gb2312	Always invalid ^{Note2}	None

Notes 1. Warning W0511171 is output.

Notes 2. Error F0593305 is output. (This library cannot be generated.)

Revision Record

Rev.	Date	Description	
		Page	Summary
Rev.1.00	May. 20, 2014	-	First Edition issued

CubeSuite+ V2.02.00 User's Manual:
RX Build

Publication Date: Rev.1.00 May. 20, 2014

Published by: Renesas Electronics Corporation



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

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

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 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

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 906, 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 Korea Co., Ltd.

12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

CubeSuite+ V2.02.00