User's Manual

CS+ V8.05.00

Integrated Development Environment

User's Manual: GHS CCRH850 Build Tool Operation

Target Device

RH850 Family

Renesas Electronics
www.renesas.com

Rev.1.00 2020.11

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, 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.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

How to Use This Manual

This manual describes the role of the CS+ integrated development environment for developing applications and systems for RH850 family, and provides an outline of its features.

CS+ is an integrated development environment (IDE) for RH850 family, 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 CS+ and design software and hardware application systems.
Purpose	This manual is intended to give users an understanding of the functions of the CS+ 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 A.WINDOW 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:	XXX (overscore over pin or signal name)
	Note:	Footnote for item marked with Note in the text
	Caution:	Information requiring particular attention
	Remarks:	Supplementary information
	Numeric representation:	Decimal ... XXXX Hexadecimal ... 0xXXXX

TABLE OF CONTENTS

1. GENERAL

This chapter explains the overview of the GHS CCRH850 build tool plug-in.

1.1 Overview

The build tool plug-in can be used to set build options for creating load modules or user libraries.

1.2 Features

The features of the build tool plug-in are shown below.

- Build option setting
  Frequently used build options can be set via the graphical user interface (GUI).

2. FUNCTIONS

This chapter describes the build procedure using CS+ 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.

Remark See "CS+ Integrated Development Environment User's Manual: Project Operation" for details about (1), (3), (4), (8), and (9).

(1) Create or load a project
Create a new project, or load an existing one.
When you create a new project, specify [Empty Application(GHS CCRH850)] as the project type.

(2) Confirm the path of tools
Confirm the setting of the [Path of Tools] category from the [Common Options] tab on the Property panel, and change it if this is required.

(3) Set a build target project
Set a build target project.

(4) Set build target files
Add or remove build target files and update the dependencies.

(5) Set the type of the output file
Select the type of the load module to be generated (see "2.2Set the Type of the Output File").

(6) Set build options
Set the options for the compiler, assembler, linker, and the like (see "2.3 Set Compile Options", "2.4 Set Assemble Options", "2.5 Set Link Options", and the like).

Remark Setting of build-tool properties is supported for version 2020.5.5, 2020.1.5, 2019.5.5, 2019.1.5, 2018.5.5, 2018.1.5, 2017.5.5, 2017.1.5, 2016.5.5, 2015.1.7, 2015.1.5, 2014.1.7, 2013.5.5, 2013.1.5, and 2012.5.5 of the GHS compiler.
Even if build options of the GHS compiler are supported by the build tool, some of them may not be supported by the CS+ debug tool. For details, refer to section "Advanced downloading" under "Download/Upload Programs" in chapter "FUNCTIONS" of "CS+ Integrated Development Environment User's Manual: RH850 Debug Tool".

(7) Set the update method of the I/O header file
Update the I/O header file in accordance with the update of the device file (see "2.8 Automatically Update the I/O Header File").

(8) Run a build
Run a build.

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] tab (for a C source file) and [Individual Assemble Options] tab (for an assembly source file).

(9) Save the project
Save the setting contents of the project to the project file.

2.1.2 Create a user library

The procedure for creating a user library is shown below.

Remark See "CS+ Integrated Development Environment User's Manual: Project Operation" for details about (1), (3), (4), (6), and (7).

(1) Create or load a project
Create a new project, or load an existing one.
When you create a new project, specify [Library(GHS CCRH850)] as the project type.

(2) Confirm the path of tools
Confirm the setting of the [Path of Tools] category from the [Common Options] tab on the Property panel, and change it if this is required.

(3) Set a build target project
Set a build target project.

(4) Set build target files
Add or remove build target files and update the dependencies.

(5) Set build options
Set the options for the compiler, assembler, librarian, and the like (see "2.3 Set Compile Options", "2.4 Set Assemble Options", "2.6 Set Create Library Options").

Remark Setting of build-tool properties is supported for version 2020.5.5, 2020.1.5, 2019.5.5, 2019.1.5, 2018.5.5, 2018.1.5, 2017.5.5, 2017.1.5, 2016.5.5, 2015.1.7, 2015.1.5, 2014.1.7, 2013.5.5, 2013.1.5, and 2012.5.5 of the GHS compiler.
Even if build options of the GHS compiler are supported by the build tool, some of them may not be supported by the CS+ debug tool. For details, refer to section "Advanced downloading" under "Download/Upload Programs" in chapter "FUNCTIONS" of "CS+ Integrated Development Environment User's Manual: RH850 Debug Tool".

(6) Run a build
Run a build.

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] tab (for a C source file) and [Individual Assemble Options] tab (for an assembly source file).

(7) Save the project
Save the setting contents of the project to the project file.

2.1.3 Use an existing GHS project file

This section describes how to use an existing GHS project file to create a new CS+ project.

- You can use CS+ to debug load module files built by the GHS compiler, in the same manner as a debug-dedicated project.
- Building based on existing GHS project files is easy.

Remark For notes on the use of GHS download-module files, refer to section "Advanced downloading" under "Download/Upload Programs" in chapter "FUNCTIONS" of "CS+ Integrated Development Environment User's Manual: RH850 Debug Tool".

The procedure for using an existing GHS project file to create a new CS+ project is shown below.

Remark See "CS+ Integrated Development Environment User's Manual: Project Operation" for details about (1), (4), (5), and (6).

(1) Create or load a project
Create a new CS+ project, or load an existing one created using an existing GHS project file.
When you create a new project, specify [Using Existing GHS Project File(GHS CCRH850)] as the project type.

(2) Confirm the path of tools
Confirm the setting of the [Path of Tools] category from the [Build Options] tab on the Property panel, and change it if this is required.

(3) Set a GHS project file to be used
Set a GHS project file to be used in the [Build] category from the [Build Options] tab on the Property panel.

Caution If you need to set build options, set them in the MULTI IDE from Green Hills Software, LLC in advance.

(4) Set a download file
Drag a load module file from Explorer or the like, and drop it onto the Download files node on the project tree.

Remark For details, refer to section "Add a download file" under "Add a File to a Project" in chapter "USING AN EXTERNAL BUILD TOOL" of "CS+ Integrated Development Environment User's Manual: Project Operation". In this case, read the statement "debug-dedicated project(s)" as "project(s) based on an existing GHS project file".

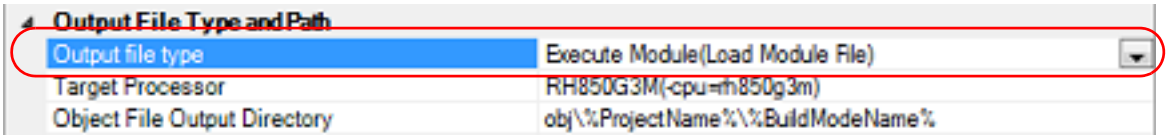
(5) Run a build
Run a build.

(6) Save the project
Save the setting contents of the project to the project file.

2.2 Set the Type of the Output File

Set the type of the file to be output as the product of the build.
Select the build tool node on the project tree and select the [Common Options] tab on the Property panel.
Select the file type in the [Output file type] property in the [Output File Type and Path] category.

Figure 2.1 [Output file type] Property



(1) When [Execute Module(Load Module File)] is selected (Default)
The load module file will be the debug target.

(2) When [Execute Module(Hex File)] is selected
The hex file will be the debug target.

Caution For the library project, this property is always [Library] and cannot be changed.

2.2.1 Change the output file name

The names of the load module file, hex file, and library file output by the build tool are set as follows by default.

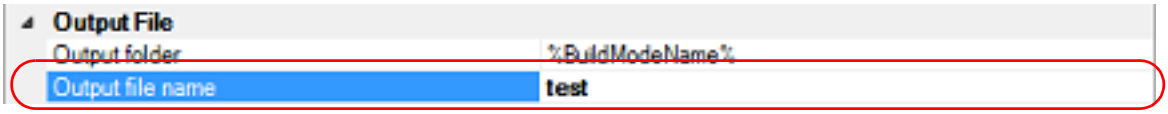
Load module file name: %ProjectName%
Hex file name: %ProjectName%.run
Library file name: lib%ProjectName%.a

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 in the [Output file name] property in the [Output File] category.

Figure 2.2 [Output file name] Property

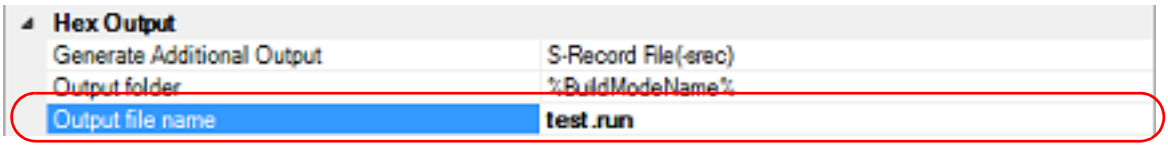


This property supports the following placeholders.
%ActiveProjectName%: Replaces with the active project name.
%MainProjectName%: Replaces with the main project name.
%ProjectName%: Replaces with the project name.

Remark You can also change the option in the same way with the [Output file name] property in the [Frequently Used Options(for Link)] category on the [Common Options] tab.

(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 hex file name to be changed to in the [Output file name] property in the [Hex Output] category.

Figure 2.3 [Output file name] Property

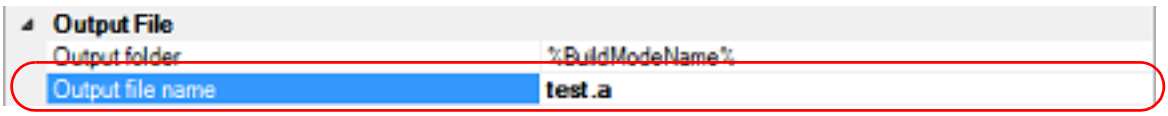


This property supports the following placeholders.
%ActiveProjectName%: Replaces with the active project name.
%MainProjectName%: Replaces with the main project name.

%ProjectName%: Replaces with the project name.

(3) When changing the library file name
Select the build tool node on the project tree and select the [Create Library Options] tab on the Property panel.
Enter the library file name to be changed to on the [Output file name] property in the [Output File] category.

Figure 2.4 [Output file name] Property



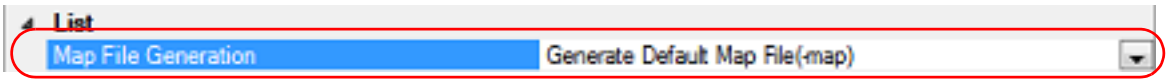
This property supports the following placeholders.
%ActiveProjectName%: Replaces with the active project name.
%MainProjectName%: Replaces with the main project name.
%ProjectName%: Replaces with the project name.

Remark You can also change the option in the same way with the [Output file name] property in the [Frequently Used Options(for Create Library)] category on the [Common Options] tab.

2.2.2 Output map information

The map information (the information of the link result) is output to the map file.
Select the build tool node on the project tree and select the [Link Options] tab (for the application project) or [Create Library Options] tab (for the library project) on the Property panel.
To output the map file, select [Generate Default Map File(-map)] in the [Map File Generation] property in the [List] category.

Figure 2.5 [Map File Generation] Property



The map file is output to the folder specified in the [Output folder] property.
The file name will be the output file name of the linker with the extension replaced by ".map".

2.3 Set Compile Options

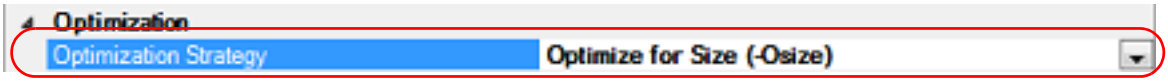
To set options for the compile phase, select the Build tool node on the project tree and select the [Compile Options] tab on the Property panel.
You can set the various compile options by setting the necessary properties in this tab.

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

2.3.1 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 [Optimize for Size (-Osize)] in the [Optimization Strategy] property in the [Optimization] category.

Figure 2.6 [Level of optimization] Property (Code Size Precedence)

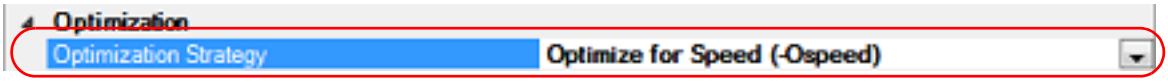


Remark You can also set the option in the same way with the [Optimization Level] property in the [Frequently Used Options(for Compile)] category on the [Common Options] tab.

2.3.2 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 [Optimize for Speed (-Ospeed)] in the [Optimization Strategy] property in the [Optimization] category.

Figure 2.7 [Level of optimization] Property (Execution Speed Precedence)

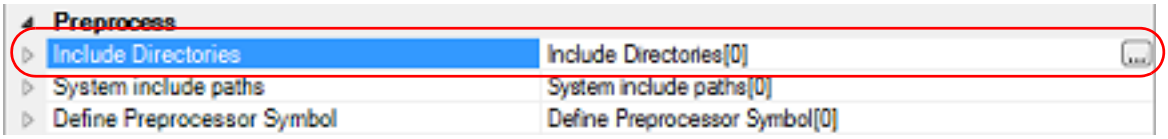


Remark You can also set the option in the same way with the [Optimization Level] property in the [Frequently Used Options(for Compile)] category on the [Common Options] tab.

2.3.3 Add an include path

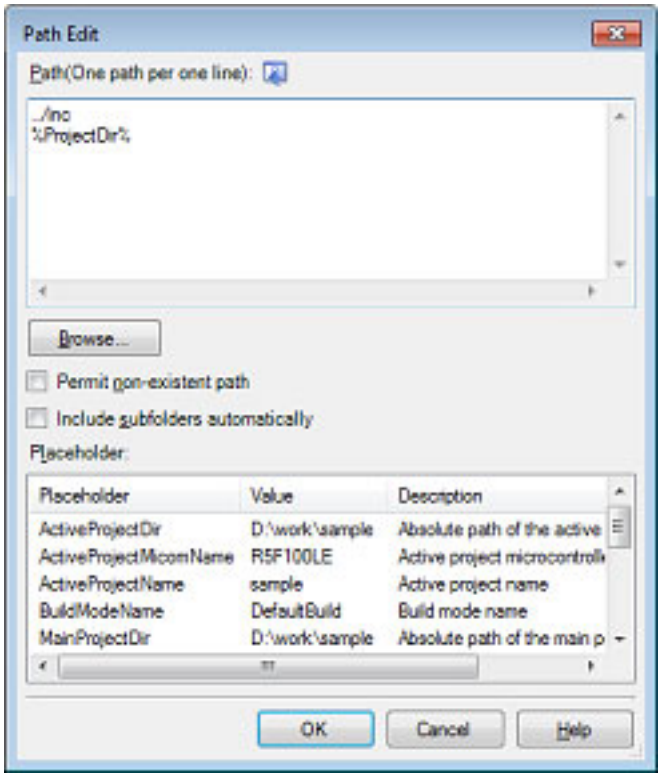
Select the build tool node on the project tree and select the [Compile Options] tab on the Property panel.
The include path setting is made with the [Include Directories] property in the [Preprocess] category.

Figure 2.8 [Include Directories] Property



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

Figure 2.9 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, up to 256 lines.

Remark 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)].

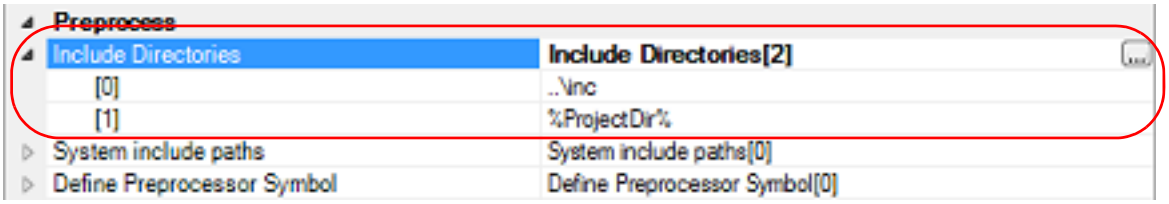
Remark 2. You can also specify the include path by one of the following procedures.

- 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].

Remark 3. Select the [Include subfolders automatically] 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 2.10 [Include Directories] Property (After Adding Include Paths)



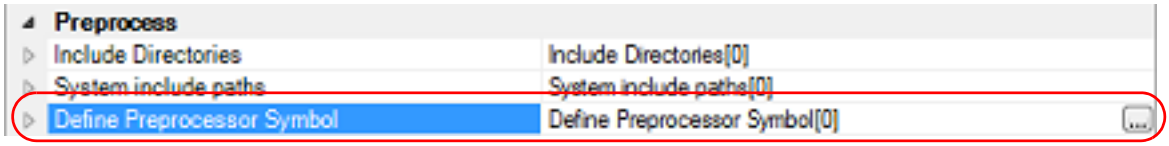
To change the include paths, you can use the [...] button or enter the path directly in the text box of the subproperty.
When the include path is added to the project tree, the path is added to the top of the subproperties automatically.

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

2.3.4 Set a macro definition

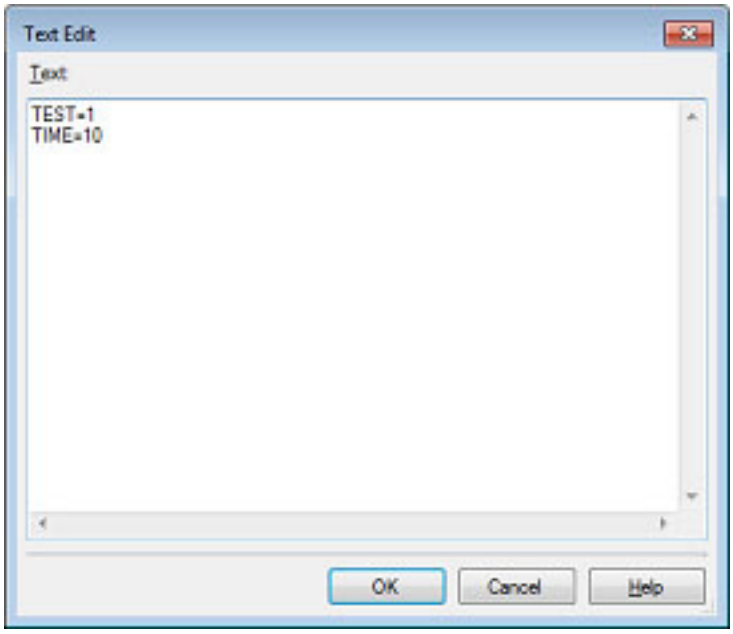
Select the build tool node on the project tree and select the [Compile Options] tab on the Property panel.
The macro definition setting is made with the [Define Preprocessor Symbol] property in the [Preprocess] category.

Figure 2.11 [Define Preprocessor Symbol] Property



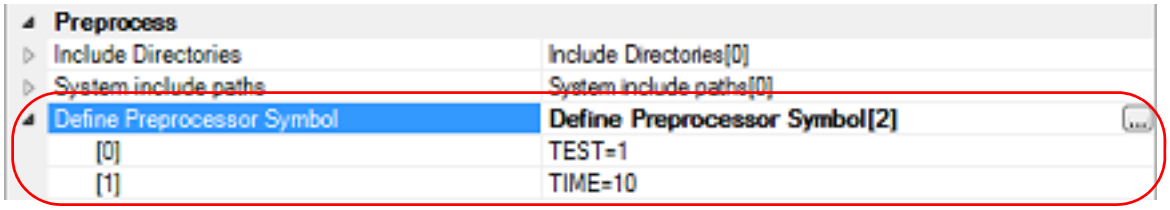
If you click the [...] button, the Text Edit dialog box will open.

Figure 2.12 Text Edit Dialog Box



Enter the macro definition in [Text] in the format of "*macro name=defined value*", with one macro name per line.
You can specify up to 256 characters per line, up to 256 lines.
The "*=defined value*" part can be omitted, and in this case, "1" is used as the defined value.
If you click the [OK] button, the entered macro definitions are displayed as subproperties.

Figure 2.13 [Define Preprocessor Symbol] 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 [Define Preprocessor Symbol] property in the [Frequently Used Options(for Compile)] category on the [Common Options] tab.

2.4 Set Assemble Options

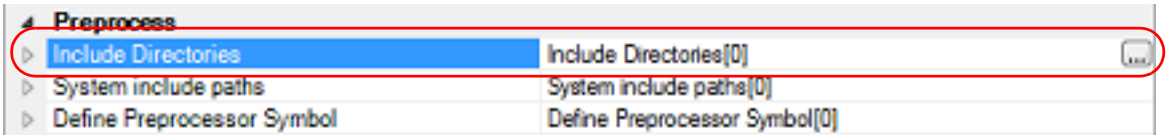
To set options for the assemble phase, select the Build tool node on the project tree and select the [Assemble Options] tab on the Property panel.
You can set the various assemble options by setting the necessary properties in this tab.

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

2.4.1 Add an include path

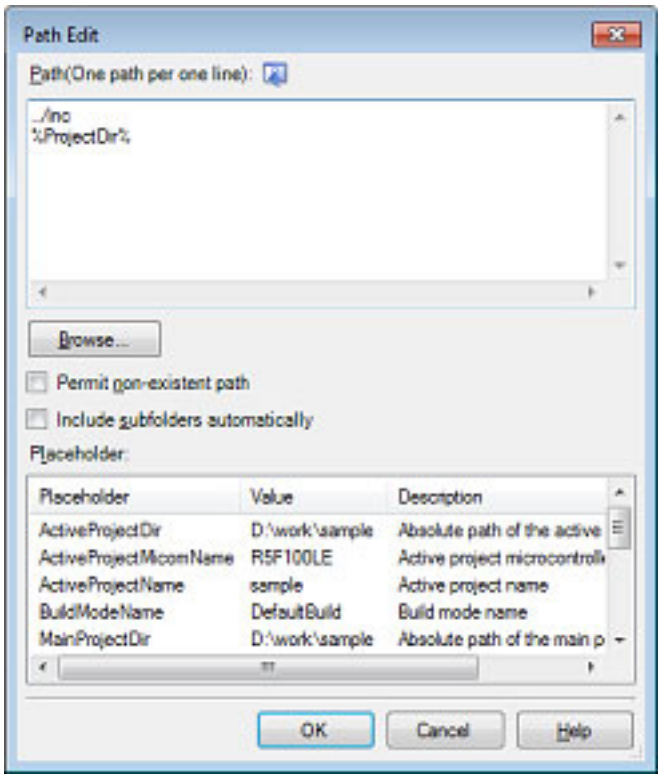
Select the build tool node on the project tree and select the [Assemble Options] tab on the Property panel.
The include path setting is made with the [Include Directories] property in the [Preprocess] category.

Figure 2.14 [Include Directories] Property



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

Figure 2.15 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, up to 256 lines.

Remark 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)].

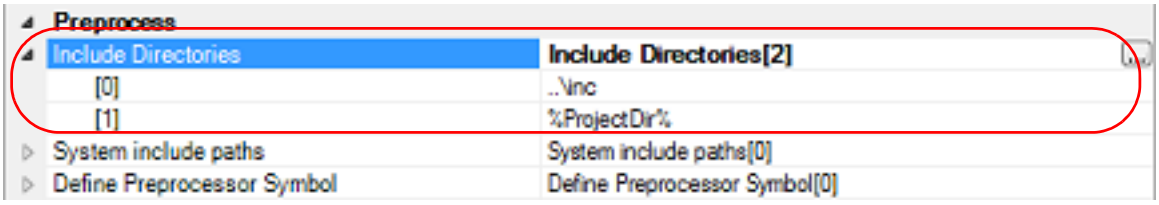
Remark 2. You can also specify the include path by one of the following procedures.

- 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].

Remark 3. Select the [Include subfolders automatically] 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 2.16 [Include Directories] Property (After Adding Include Paths)



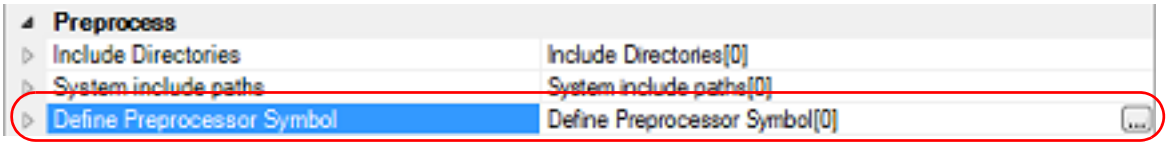
To change the include paths, you can use the [...] button or enter the path directly in the text box of the subproperty.
When the include path is added to the project tree, the path is added to the top of the subproperties automatically.

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

2.4.2 Set a macro definition

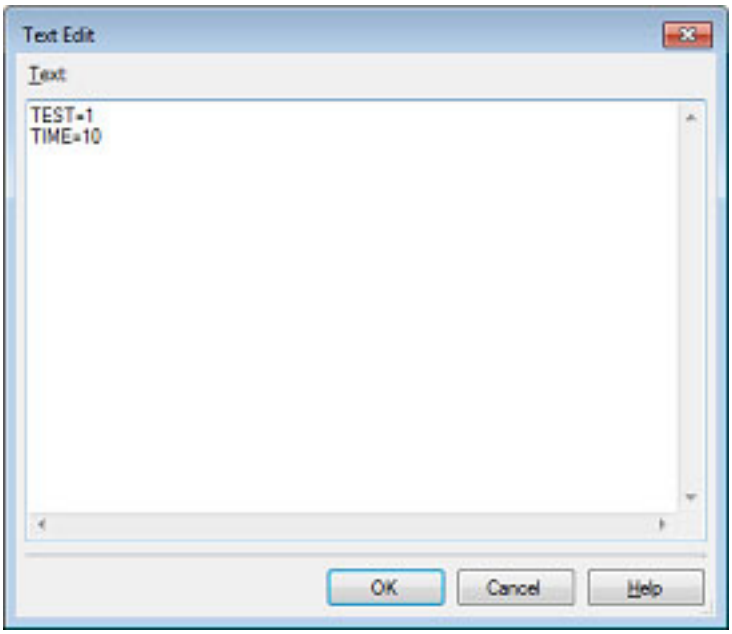
Select the build tool node on the project tree and select the [Assemble Options] tab on the Property panel.
The macro definition setting is made with the [Define Preprocessor Symbol] property in the [Preprocess] category.

Figure 2.17 [Define Preprocessor Symbol] Property



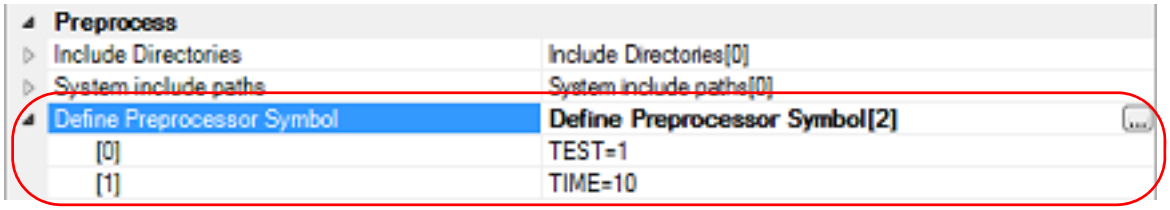
If you click the [...] button, the Text Edit dialog box will open.

Figure 2.18 Text Edit Dialog Box



Enter the macro definition in [Text] in the format of "*macro name=defined value*", with one macro name per line.
You can specify up to 256 characters per line, up to 256 lines.
The "=*defined value*" part can be omitted, and in this case, "1" is used as the defined value.
If you click the [OK] button, the entered macro definitions are displayed as subproperties.

Figure 2.19 [Define Preprocessor Symbol] 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 [Define Preprocessor Symbol] property in the [Frequently Used Options(for Assemble)] category on the [Common Options] tab.

2.5 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.

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

2.5.1 Add a user library

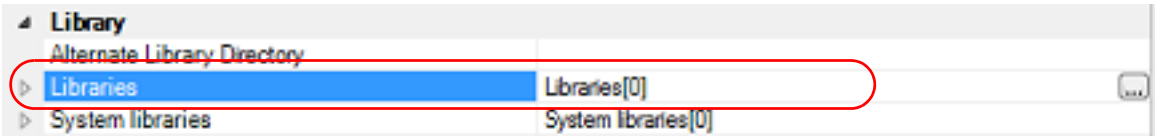
Adding a user library is made with the property or on the project tree.

(1) Addition using the property

Select the build tool node on the project tree and select the [Link Options] tab on the Property panel.

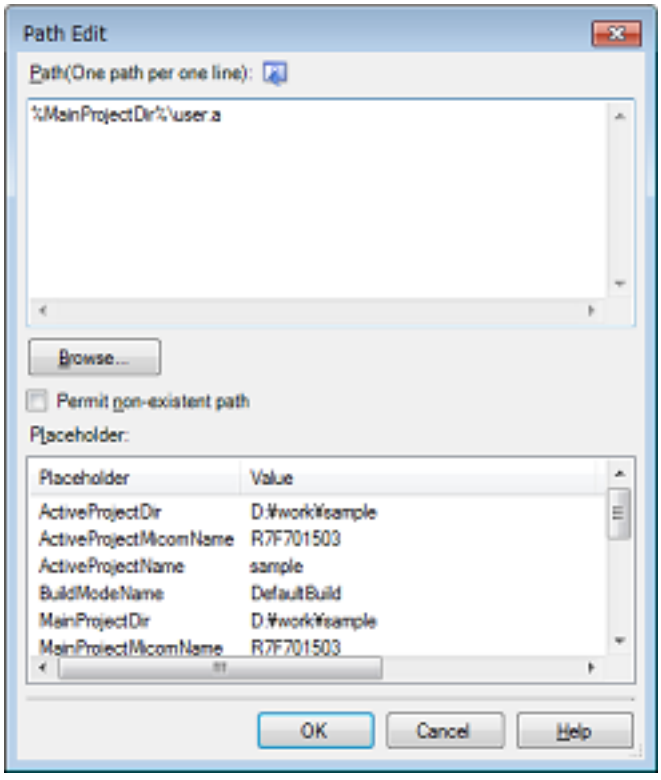
Adding a user library is made with the [Libraries] property in the [Library] category.

Figure 2.20 [Libraries] Property



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

Figure 2.21 Path Edit Dialog Box



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

You can specify up to 259 characters per line, up to 65536 lines.

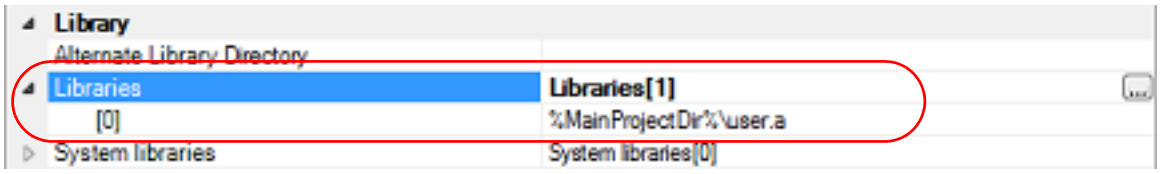
Remark 1. This property supports placeholders.
If a line is double clicked in [Placeholder], the placeholder will be reflected in [Path(One path per one line)].

Remark 2. You can also specify the library file 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].

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

Figure 2.22 [Libraries] 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.

Remark You can also set the option in the same way with the [Libraries] property in the [Frequently Used Options(for Link)] category on the [Common Options] tab.

(2) Addition from the project tree
Adding a library file to the project tree is performed from the Add Existing File dialog box.
Dropping a library file in the project tree is also possible.
When a library file is added from the project tree, it is subject to timestamp comparison with the load module at build, and the link processing is executed when the added library file is updated.

2.6 Set Create Library Options

To set options for the librarian, select the Build tool node on the project tree and select the [Create Library Options] tab on the Property panel.
You can set the various create library options by setting the necessary properties in this tab.

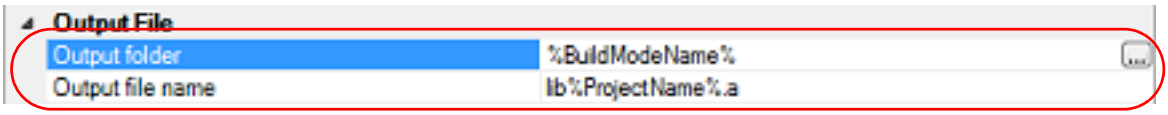
Caution This tab is displayed for the library project.

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

2.6.1 Set the output of a library file

Select the build tool node on the project tree and select the [Create Library Options] tab on the Property panel.
The setting to output a library file is made with the [Output File] category.

Figure 2.23 [Output File] Category



(1) Set the output folder
Setting the output folder is made with 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.

%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.

"%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.
If the extension is omitted, ".a" is automatically added.

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.

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

2.7 Set Build Options Separately

Build options are set at the project or file level.

Project level: See "2.7.1Set build options at the project level"
File level: See "2.7.2Set build options at the file level"

2.7.1 Set build options at the project level

To set options for build options for the project (main project or subproject), select the Build tool node on the project tree to display the Property panel.
Select the phase tab and set build options by setting the necessary properties.

Compile phase: [Compile Options] tab
Assemble phase: [Assemble Options] tab
Link phase: [Link Options] tab
Create library phase: [Create Library Options] tab
I/O header file generation tool: [I/O Header File Generation Options] tab

2.7.2 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 the C source file on the project tree and select the [Build Settings] tab on the Property panel.
Select [Yes] in the [Set individual compile option] property in the [Build] category. The Message Dialog Box will open.

Figure 2.24 [Set individual compile option] Property

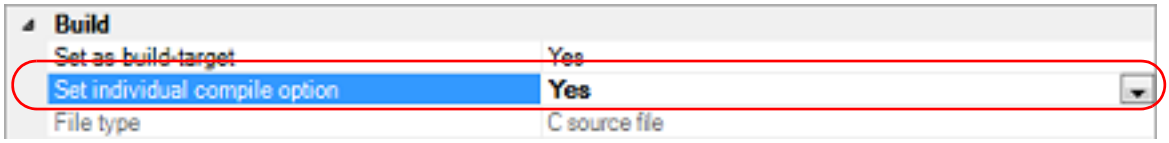
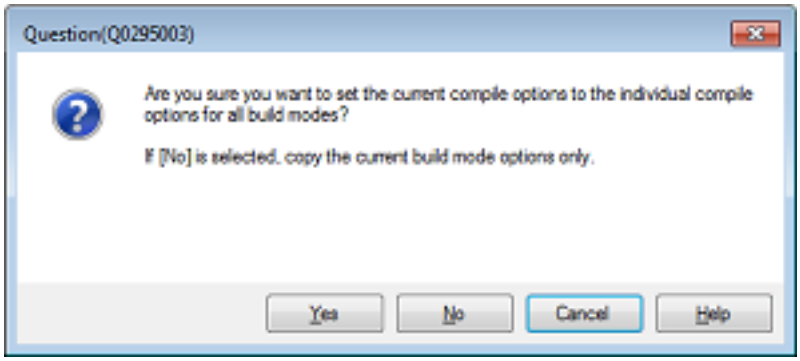


Figure 2.25 Message Dialog Box



Click [Yes] in the dialog box. The [Individual Compile Options] tab will be displayed.
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 [Common Options] tab and [Compile Options] tab by default except the properties shown below.

- [Include Directories] and [Use whole include paths specified for build tool] in the [Preprocess] category
- [Object module file name] in the [Output File] category

(2) When setting assemble options for an assembly source file
Select the assembly source file on the project tree and select the [Build Settings] tab on the Property panel.
Select [Yes] in the [Set individual assemble option] property in the [Build] category. The Message Dialog Box will open.

Figure 2.26 [Set individual assemble option] Property

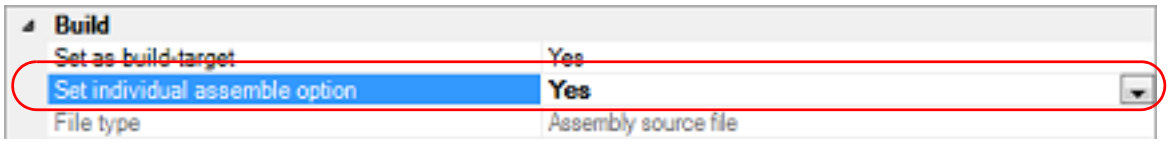
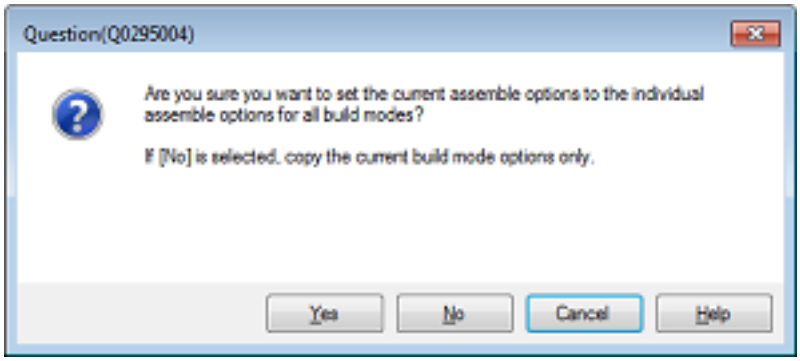


Figure 2.27 Message Dialog Box



Click [Yes] in the dialog box. The [Individual Assemble Options] tab will be displayed.
You can set assemble options for the assembly source file by setting the necessary properties in this tab.

Note that this tab takes over the settings of the [Common Options] tab and [Assemble Options] tab by default except the properties shown below.

- [Include Directories] and [Use whole include paths specified for build tool] in the [Preprocess] category
- [Object module file name] in the [Output File] category

2.8 Automatically Update the I/O Header File

When an empty application project is newly created, an I/O header file corresponding to the selected device is automatically generated.

If the I/O header file needs to be automatically updated in response to the update of the device file, use the following update method.

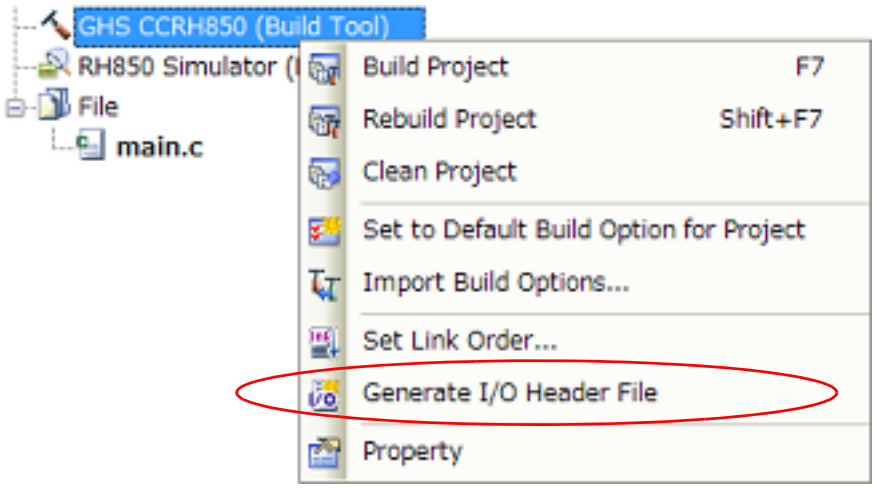
The I/O header file is automatically generated as "iodefine.h" when an empty application project is newly created.

Remark The I/O header file is generated in the same folder containing the project file. If a file with the same name already exists, the existing file is renamed as "iodefine.bak" as a backup.

The timing to update the I/O header file and the update method are shown below.

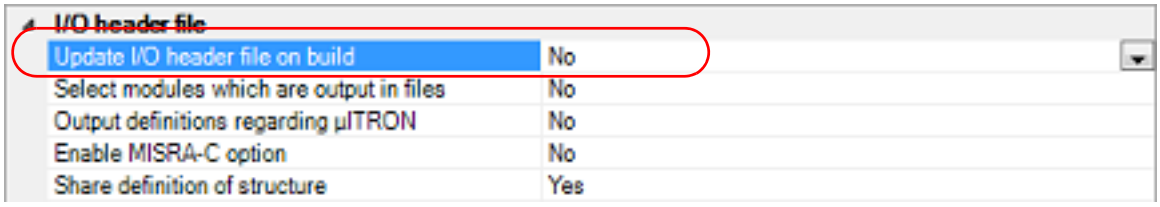
- At opening of the project
  CS+ checks the version of the device file when a project is opened.
  If the device file has been updated and there is a possibility that the I/O header file needs to be updated, a message is displayed on the Output panel. Update the I/O header file with the method below as required.
  - On the Project Tree panel, select the Build tool node, and then select [Generate I/O Header File] from the context menu

Figure 2.28 [Generate I/O Header File] Item



- At build
  The I/O header file can be updated automatically when the build process is performed and immediately before build.
  Set the [Update I/O header file on build] property of the [I/O Header File Generation Options] tab in the Property panel. The update conditions can also be changed in the property of the same category.

Figure 2.29 [Update I/O header file on build] Property



A. WINDOW REFERENCE

This appendix explains panels/dialog boxes used in the build tool.

A.1 Description

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

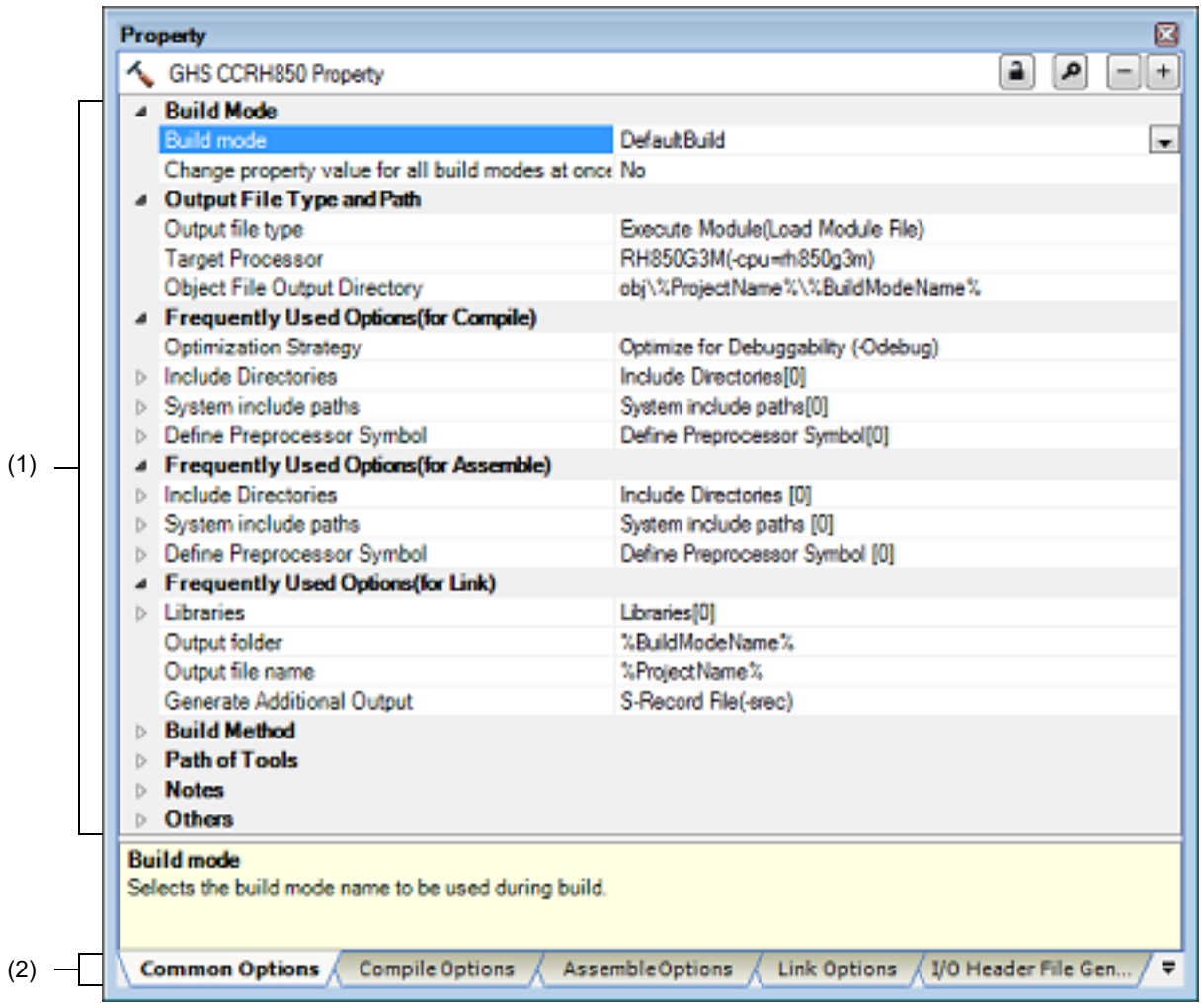
Table A.1 List of Panels/Dialog Boxes

Panel/Dialog Box Name	Function Description
Property panel	This panel is used to display the detailed information on the Build tool node or file that is selected on the Project Tree panel and change the settings of the information.
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.
Select Modules Which Are Output in Files dialog box	This dialog box is used to set modules which are output to the I/O header file.

Property panel

This panel is used to display the detailed information on the Build tool node or file that is selected on the Project Tree panel by every category and change the settings of the information.

Figure A.1 Property Panel



The following items are explained here.

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

[How to open]

- On the Project Tree panel, select the Build tool node or file and then select [Property] from the [View] menu or [Property] from the context menu.

Remark When either one of the Build tool node or file on the Project Tree panel is selected while the Property panel has been opened, the detailed information of the selected item is displayed.

[Description of each area]

(1) Detailed information display/change area
In this area, the detailed information on the Build tool node or file that is selected on the Project Tree panel is displayed by every category in the list. And the settings of the information can be changed directly.
Mark ◢ indicates that all the items in the category are expanded. Mark ▷ indicates that all the items are collapsed. You can expand/collapse the items by clicking these marks or double clicking the category name.
Mark HEX indicates that only a hexadecimal number is allowed to input in the text box.
See the section on each tab for the details of the display/setting in the category and its contents.

(2) Tab selection area
Categories for the display of the detailed information are changed by selecting a tab.
In this panel, the following tabs are contained (see the section on each tab for the details of the display/setting on the tab).

Remark When multiple components are selected on the Project Tree panel, only the tab that is common to all the components is displayed.
If the value of the property is modified, that is taken effect to the selected components all of which are common to all.

(a) When the Build tool node is selected on the Project Tree panel, and when the project type is [Empty Application(GHS CCRH850)] or [Library(GHS CCRH850)]

- [Common Options] tab
- [Compile Options] tab
- [Assemble Options] tab
- [Link Options] tab
- [Create Library Options] tab
- [I/O Header File Generation Options] tab

(b) When the Build tool node is selected on the Project Tree panel, and when the project type is [Using Existing GHS Project File(GHS CCRH850)]

- [Build Options] tab

(c) When a file is selected on the Project Tree panel

- [Build Settings] tab (for C source file, assembly source file, object file, and library file)
- [Individual Compile Options] tab (for C source file)
- [Individual Assemble Options] tab (for assembly source file)
- [File Information] tab[Note]

Note See "CS+ Integrated Development Environment User's Manual: Project Operation" for details about the [File Information] tab.

[[Edit] menu (only available for the Property panel)]

Undo	Cancels the previous edit operation of the value of the property.
Cut	While editing the value of the property, cuts the selected characters and copies them to the clipboard.
Copy	Copies the selected characters of the property to the clipboard.
Paste	While editing the value of the property, inserts the contents of the clipboard.
Delete	While editing the value of the property, deletes the selected characters.
Select All	While editing the value of the property, selects all the characters of the selected property.

[Context menu]

Undo	Cancels the previous edit operation of the value of the property.
Cut	While editing the value of the property, cuts the selected characters and copies them to the clipboard.
Copy	Copies the selected characters of the property to the clipboard.
Paste	While editing the value of the property, inserts the contents of the clipboard.
Delete	While editing the value of the property, deletes the selected characters.
Select All	While editing the value of the property, selects all the characters of the selected property.
Reset to Default	Restores the configuration of the selected item to the default configuration of the project. For the [Individual Compile Options] tab and [Individual Assemble Options] tab, restores to the configuration of the general option.
Reset All to Default	Restores all the configuration of the current tab to the default configuration of the project. For the [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)[Output File Type and Path]
(3)[Frequently Used Options(for Compile)]
(4)[Frequently Used Options(for Assemble)]
(5)[Frequently Used Options(for Link)]
(6)[Frequently Used Options(for Create Library)]
(7)[Build Method]
(8)[Path of Tools]
(9)[Notes]
(10)[Others]

Remark 1. 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 Create Library)] category	[Create Library Options] tab

Remark 2. For details on compiler driver options, refer to "MULTI: Building Applications for Embedded V850 and RH850" released by Green Hills Software, LLC. Please do not contact Green Hills Software, LLC with any inquiries about the CCRH850 build-tool plugin of CS+.

Caution 1. This tab is displayed only when the current project has been created with [Empty Application(GHS CCRH850)] or [Library(GHS CCRH850)] selected as the project type.

Caution 2. If you have upgraded the GHS compiler version, confirm the setting of the [Compiler package folder] property from the [Path of Tools] category.

[Description of each category]

(1) [Build Mode]
The detailed information on the build mode is displayed and the configuration can be changed.

Build mode	Select the build mode to be used during a 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	Runs a build with the default build mode that is set when a new project is created.
		Build mode that is added to the project	Runs a build with the build mode that is added to the project (other than Default-Build).

Change property value for all build modes at once	Select whether to reflect the value newly set to all build modes when a value is set in this property. Be careful since the value set may not be an appropriate value for other build modes.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Reflects the value newly set to all build modes when a value is set in this property.
		No	Does not reflect the value newly set to all build modes when a value is set in this property.

(2) [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	The file type set here will be the debug target for other than the library project. For other than library projects, only [Execute Module(Load Module File)] and [Execute Module(Hex File)] are displayed. For the library project, only [Library] is displayed.		
	Default	- For other than the library project Execute Module(Load Module File) - For the library project Library	
	How to change	Select from the drop-down list.	
	Restriction	Execute Mod-ule(Load Module File)	Generates a load module file during a build. The load module file will be the debug target.
		Execute Module(Hex File)	Generates a hex file during a build. The hex file will be the debug target. This item is displayed only when other than [Not specify(No option specified)] in the [Generate Additional Output] property in the [Hex Output] category from the [Link Options] tab is selected.
		Library	Generates a library file during a build.

Target Processor	Specify the -cpu option of the compiler driver.		
	Default	RH850G3M(-cpu=rh850g3m)	
	How to change	Select from the drop-down list.	
	Restriction	RH850G3M(-cpu=rh850g3m)	Specifies the -cpu=rh850g3m option.
		RH850G3K(-cpu=rh850g3k)	Specifies the -cpu=rh850g3k option.
		RH850G3KH(-cpu=rh850g3kh)	Specifies the -cpu=rh850g3kh option.
		RH850G3MH(-cpu=rh850g3mh)	Specifies the -cpu=rh850g3mh option.
		RH850G4MH(-cpu=rh850g4mh)	Specifies the -cpu=rh850g4mh option.
		Not specify(No option specified)	Does not specify the -cpu option.
Object File Output Directory	Specify the folder which the intermediate file is output. The following placeholder is supported. %ActiveProjectDir%: Replaces with the absolute path of the active project folder. %BuildModeName%: Replaces with the build mode name. %MainProjectDir%: Replaces with the absolute path of the main project folder. %ProjectDir%: Replaces with the absolute path of the project folder. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -object_dir or -o option of the compiler driver.		
	Default	obj\%ProjectName%\%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	

(3) [Frequently Used Options(for Compile)]
The detailed information on frequently used options during compilation is displayed and the configuration can be changed.

<table>
<tr><td rowspan="9">Optimization Strategy</td><td colspan="3">Specify the -O option of the compiler driver.</td></tr>
<tr><td>Default</td><td colspan="2">Optimize for Debuggability (-Odebug)</td></tr>
<tr><td>How to change</td><td colspan="2">Select from the drop-down list.</td></tr>
<tr><td rowspan="6">Restriction</td><td>Optimize for Debug-gability (-Odebug)</td><td>Specifies the -Odebug option.</td></tr>
<tr><td>Optimize for General Use (-O)</td><td>Specifies the -O option.</td></tr>
<tr><td>Optimize for Size (-Osize)</td><td>Specifies the -Osize option.</td></tr>
<tr><td>Optimize for Speed (-Ospeed)</td><td>Specifies the -Ospeed option.</td></tr>
<tr><td>No Optimizations (-Onone)</td><td>Specifies the -Onone option.</td></tr>
<tr><td>Not specify(No option specified)</td><td>Does not specify Optimization Strategy.</td></tr>
<tr><td rowspan="4">Include Directories</td><td colspan="3">Specify the -I option of the compiler driver.
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.
The specified include path is displayed as the subproperty.
Uppercase characters and lowercase characters are not distinguished for the include paths.</td></tr>
<tr><td>Default</td><td colspan="2">Include Directories[number of defined items]</td></tr>
<tr><td>How to change</td><td colspan="2">Edit by the Path Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td colspan="2">Up to 247 characters
Up to 256 items can be specified.</td></tr>
</table>

<table>
<tr><td rowspan="4">System include paths</td><td colspan="2">Change the specified order of the include paths which CS+ sets during compiling.
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 system include path is searched with lower priority than the additional include path.
The reference point of the path is the project folder.
This corresponds to the -I option of the compiler driver.
The include path is displayed as the subproperty.</td></tr>
<tr><td>Default</td><td>System include paths[number of defined items]</td></tr>
<tr><td>How to change</td><td>Edit by the System Include Path Order dialog box which appears when clicking the [...] button.</td></tr>
<tr><td>Restriction</td><td>Changes not allowed (Only the specified order of the include paths can be changed.)</td></tr>
<tr><td rowspan="4">Define Preprocessor Symbol</td><td colspan="2">Specify the -D option of the compiler driver.
Specify in the format of "macro name=defined value", with one macro name per line.
The specified macro is displayed as the subproperty.</td></tr>
<tr><td>Default</td><td>Define Preprocessor Symbol[number of defined items]</td></tr>
<tr><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td>Up to 256 characters
Up to 256 items can be specified.</td></tr>
</table>

(4) [Frequently Used Options(for Assemble)]
The detailed information on frequently used options during assembling is displayed and the configuration can be changed.

<table>
<tr><td>Include Directories</td><td colspan="2">Specify the -I option of the compiler driver.
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.
The specified include path is displayed as the subproperty.
Uppercase characters and lowercase characters are not distinguished for the include paths.</td></tr>
<tr><td></td><td>Default</td><td>Include Directories[number of defined items]</td></tr>
<tr><td></td><td>How to change</td><td>Edit by the Path Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td></td><td>Restriction</td><td>Up to 247 characters
Up to 256 items can be specified.</td></tr>
<tr><td>System include paths</td><td colspan="2">Change the specified order of the include paths which CS+ sets during assembling.
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 system include path is searched with lower priority than the additional include path.
The reference point of the path is the project folder.
This corresponds to the -I option of the compiler driver.
The include path is displayed as the subproperty.</td></tr>
<tr><td></td><td>Default</td><td>System include paths[number of defined items]</td></tr>
<tr><td></td><td>How to change</td><td>Edit by the System Include Path Order dialog box which appears when clicking the [...] button.</td></tr>
<tr><td></td><td>Restriction</td><td>Changes not allowed (Only the specified order of the include paths can be changed.)</td></tr>
</table>

Define Preprocessor Symbol	Specify the -D option of the compiler driver. Specify in the format of "*macro name=defined value*", with one macro name per line. The specified macro is displayed as the subproperty.	
	Default	Define Preprocessor Symbol[*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 256 characters Up to 256 items can be specified.

(5) [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.

Libraries	Specify the -l option of the compiler driver. 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 library file name is displayed as the subproperty.	
	Default	Libraries[*number of defined items*]
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. -> Edit by the Specify Using Library File dialog box which appears when clicking the [Browse...] button. For the subproperty, you can enter directly in the text box.
	Restriction	Up to 259 characters Up to 65536 items can be specified.

Output folder	Specify the output folder. 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. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -o option of the compiler driver.		
	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. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectName%: Replaces with the main project name. %ProjectName%: Replaces with the project name. This corresponds to the -o option of the compiler driver.		
	Default	%ProjectName%	
	How to change	Directly enter in the text box.	
	Restriction	Up to 259 characters	
Generate Additional Output	Specify the -srec or -hex option of the compiler driver.		
	Default	S-Record File(-srec)	
	How to change	Select from the drop-down list.	
	Restriction	S-Record File(-srec)	Specifies the -srec option.
		HEX386 File(-hex)	Specifies the -hex option.
		Not specify(No option specified)	Does not specify Generate Additional Output.

(6) [Frequently Used Options(for Create Library)]
The detailed information on frequently used options during library generation is displayed and the configuration can be changed.
This category is displayed only for the library project.

Output folder	Specify the output folder. 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. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -o option of the compiler driver.	
	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. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectName%: Replaces with the main project name. %ProjectName%: Replaces with the project name. This corresponds to the -o option of the compiler driver.	
	Default	%ProjectName%
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

(7) [Build Method]
The detailed information on the build method is displayed and the configuration can be changed.

Handling the source file includes unfound file	Select whether to recompile/assemble the source file if it includes a file that is not found in the standard include paths and the [Include Directories] property.		
	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 it includes a file that is not found.
		Ignore re-compiling/assembling the source file	Does not recompile/assemble the source file even if it includes a file that is not found.

(8) [Path of Tools]
The detailed information on the path of tools is displayed and the configuration can be changed.

Compiler package folder	Specify the folder of the compiler package to be used. This is common to all the build modes.	
	Default	*Value-according-to-installation-environment*
	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
Executable file name of compiler	Specify the executable file name of the compiler. This is common to all the build modes.	
	Default	ccrh850.exe
	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
Executable file name of assembler	Specify the executable file name of the assembler. This is common to all the build modes.	
	Default	ccrh850.exe
	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
Executable file name of linker	Specify the executable file name of the linker. This is common to all the build modes.	
	Default	ccrh850.exe
	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

(9) [Notes]
The detailed information on notes is displayed and the configuration can be changed.

Memo	Add memos to the build tool. Add one item in one line. This setting is common to all the build modes. The specified memo is displayed as the subproperty.	
	Default	Memo[*number-of-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 256 characters Up to 256 items can be specified.

(10) [Others]
Other detailed information on the build tool is displayed and the configuration can be changed.

<table>
<tr><td rowspan="5">Output message format</td><td colspan="3">Specify the format of the message under build execution.
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.
%Options%: Replaces with the command line option under build execution.
%Program%: Replaces with the program name under execution.
%TargetFiles%: Replaces with the file name being compile/assemble or making link.
If this is blank, "%Program% %Options%" will be set automatically.</td></tr>
<tr><td>Default</td><td colspan="2">%TargetFiles%</td></tr>
<tr><td>How to change</td><td colspan="2">Directly enter in the text box (up to 256 characters) or select from the drop-down list.</td></tr>
<tr><td rowspan="3">Restriction</td><td>%TargetFiles%</td><td>Displays the file name in the output message.</td></tr>
<tr><td>%TargetFiles%: %Options%</td><td>Displays the file name and command line options in the output message.</td></tr>
<tr><td>%Program% %Options%</td><td>Displays the program name and command line options in the output message.</td></tr>
<tr><td rowspan="4">Format of build option list</td><td colspan="3">Specify the display format of the build option list.
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.
%Options%: Replaces with the command line option under build execution.
%Program%: Replaces with the program name under execution.
%TargetFiles%: Replaces with the file name being compile/assemble or making link.
If this is blank, "%TargetFiles% : %Program% %Options%" will be set automatically.</td></tr>
<tr><td>Default</td><td colspan="2">%TargetFiles% : %Program% %Options%</td></tr>
<tr><td>How to change</td><td colspan="2">Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.</td></tr>
<tr><td>Restriction</td><td colspan="2">Up to 256 characters</td></tr>
</table>

<table>
<tr><td>Commands executed before build processing</td><td colspan="2">Specify 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.
%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.
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.
The placeholders can be described in the script.
The specified command is displayed as the subproperty.</td></tr>
<tr><td></td><td>Default</td><td>Commands executed before build processing[number of defined items]</td></tr>
<tr><td></td><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td></td><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
<tr><td>Commands executed after build processing</td><td colspan="2">Specify 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.
%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.
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.
The placeholders can be described in the script.
The specified command is displayed as the subproperty.</td></tr>
<tr><td></td><td>Default</td><td>Commands executed after build processing[number of defined items]</td></tr>
<tr><td></td><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td></td><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
</table>

Other additional options	Input the option to be added additionally. The options set here are added at the end of the options group of the compiler driver.	
	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

[Compile Options] tab

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

Remark For details on compiler driver options, refer to "MULTI: Building Applications for Embedded V850 and RH850" released by Green Hills Software, LLC. Please do not contact Green Hills Software, LLC with any inquiries about the CCRH850 build-tool plugin of CS+.

Caution This tab is displayed only when the current project has been created with [Empty Application(GHS CCRH850)] or [Library(GHS CCRH850)] selected as the project type.

[Description of each category]

(1) [Debug Information]
The detailed information on debug information is displayed and the configuration can be changed.

Debugging Level	Specify the -G option of the compiler driver.		
	Default	MULTI(-G)	
	How to change	Select from the drop-down list.	
	Restriction	MULTI(-G)	Specifies the -G option.
		Not specify(No option specified)	Does not specify the -G option.
Generate MULTI and Native Information	Specify the -dwarf2 option of the compiler driver.		
	Default	On(-dwarf2)	
	How to change	Select from the drop-down list.	
	Restriction	On(-dwarf2)	Specifies the -dwarf2 option.
		Not specify(No option specified)	Does not specify the -dwarf2 option.

(2) [Optimization]
The detailed information on the optimization is displayed and the configuration can be changed.

Optimization Strategy	Specify the -O option of the compiler driver.		
	Default	Optimize for Debuggability (-Odebug)	
	How to change	Select from the drop-down list.	
	Restriction	Optimize for Debug-gability (-Odebug)	Specifies the -Odebug option.
		Optimize for General Use (-O)	Specifies the -O option.
		Optimize for Size (-Osize)	Specifies the -Osize option.
		Optimize for Speed (-Ospeed)	Specifies the -Ospeed option.
		No Optimizations (-Onone)	Specifies the -Onone option.
		Not specify(No option specified)	Does not specify Optimization Strategy.

(3) [Preprocess]
The detailed information on preprocessing is displayed and the configuration can be changed.

Include Directories	Specify the -I option of the compiler driver. 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. The specified include path is displayed as the subproperty. Uppercase characters and lowercase characters are not distinguished for the include paths.	
	Default	Include Directories[*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 enter directly in the text box.
	Restriction	Up to 247 characters Up to 256 items can be specified.

System include paths	Change the specified order of the include paths which CS+ sets during compiling. 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 system include path is searched with lower priority than the additional include path. The reference point of the path is the project folder. This corresponds to the -I option of the compiler driver. The include path is displayed as the subproperty.	
	Default	System include paths[*number of defined items*]
	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.)
Define Preprocessor Symbol	Specify the -D option of the compiler driver. Specify in the format of "*macro name=defined value*", with one macro name per line. The specified macro is displayed as the subproperty.	
	Default	Define Preprocessor Symbol[*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 256 characters Up to 256 items can be specified.

(4) [Output Code]
The detailed information on output codes is displayed and the configuration can be changed.

Special Data Area	Specify the -sda and -zda options of the compiler driver.		
	Default	Not specify(No option specified)	
	How to change	Select from the drop-down list.	
	Restriction	Small Data Area(-sda)	Specifies the -sda option.
		Small Data Area with Threshold(-sda=size)	Specifies the -sda=size option.
		Zero Data Area with Threshold(-zda=size)	Specifies the -zda=size option.
		Not specify(No option specified)	Does not specify Special Data Area.

Threshold Value	Specify Threshold Value of the -sda or -zda option of the compiler driver. This property is displayed when [Small Data Area with Threshold(-sda=size)] or [Zero Data Area with Threshold(-zda=size)] in the [Special Data Area] property is selected.		
	Default	Blank	
	How to change	Directly enter in the text box.	
	Restriction	Decimal number or blank	
23-bit SDA	Specify the -large_sda option of the compiler driver.		
	Default	Not specify(No option specified)	
	How to change	Select from the drop-down list.	
	Restriction	Generate 23-bit SDA relocations for load/store instructions(-large_sda)	Specifies the -large_sda option.
		Not specify(No option specified)	Does not specify the -large_sda option.

(5) [Others]
Other detailed information on compilation is displayed and the configuration can be changed.

Intermediate Forms of Output	Specify the -c option of the compiler driver.		
	Default	Object File(-c)	
	How to change	Select from the drop-down list.	
	Restriction	Object File(-c)	Specifies the -c option.
		Not specify(No option specified)	Does not specify the -c option.

<table>
<tr><td>Commands executed before compile processing</td><td colspan="2">Specify 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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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.</td></tr>
<tr><td></td><td>Default</td><td>Commands executed before compile processing[number of defined items]</td></tr>
<tr><td></td><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td></td><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
</table>

<table>
<tr><td rowspan="4">Commands executed after compile processing</td><td colspan="2">Specify 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.
%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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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 compile processing.
The placeholders can be described in the script.
The specified command is displayed as the subproperty.</td></tr>
<tr><td>Default</td><td>Commands executed after compile processing[number of defined items]</td></tr>
<tr><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
<tr><td rowspan="4">Other additional options</td><td colspan="2">Input the compile option to be added additionally.
The options set here are added at the end of the compile options group.</td></tr>
<tr><td>Default</td><td>Blank</td></tr>
<tr><td>How to change</td><td>Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.</td></tr>
<tr><td>Restriction</td><td>Up to 32767 characters</td></tr>
</table>

[Assemble Options] tab

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

(1)[Debug Information]
(2)[Preprocess]
(3)[Others]

Remark For details on compiler driver options, refer to "MULTI: Building Applications for Embedded V850 and RH850" released by Green Hills Software, LLC. Please do not contact Green Hills Software, LLC with any inquiries about the CCRH850 build-tool plugin of CS+.

Caution This tab is displayed only when the current project has been created with [Empty Application(GHS CCRH850)] or [Library(GHS CCRH850)] selected as the project type.

[Description of each category]

(1) [Debug Information]
The detailed information on debug information is displayed and the configuration can be changed.

Debugging Level	Specify the -G option of the compiler driver.		
	Default	MULTI(-G)	
	How to change	Select from the drop-down list.	
	Restriction	MULTI(-G)	Specifies the -G option.
		Not specify(No option specified)	Does not specify the -G option.

(2) [Preprocess]
The detailed information on preprocessing is displayed and the configuration can be changed.

Include Directories	Specify the -I option of the compiler driver. 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. The specified include path is displayed as the subproperty. Uppercase characters and lowercase characters are not distinguished for the include paths.	
	Default	Include Directories[*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 enter directly in the text box.
	Restriction	Up to 247 characters Up to 256 items can be specified.

System include paths	Change the specified order of the include paths which CS+ sets during assembling. 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 system include path is searched with lower priority than the additional include path. The reference point of the path is the project folder. This corresponds to the -I option of the compiler driver. The include path is displayed as the subproperty.	
	Default	System include paths[*number of defined items*]
	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.)
Define Preprocessor Symbol	Specify the -D option of the compiler driver. Specify in the format of "*macro name=defined value*", with one macro name per line. The specified macro is displayed as the subproperty.	
	Default	Define Preprocessor Symbol[*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 256 characters Up to 256 items can be specified.

(3) [Others]
Other detailed information on assembly is displayed and the configuration can be changed.

Intermediate Forms of Output	Specify the -c option of the compiler driver.		
	Default	Object File(-c)	
	How to change	Select from the drop-down list.	
	Restriction	Object File(-c)	Specifies the -c option.
		Not specify(No option specified)	Does not specify the -c option.

<table>
<tr><td>Commands executed before assemble processing</td><td colspan="2">Specify 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.
%MicomToolPath%: Replaces with the absolute path of the install folder of this product.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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.</td></tr>
<tr><td></td><td>Default</td><td>Commands executed before assemble processing[number of defined items]</td></tr>
<tr><td></td><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td></td><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
</table>

<table>
<tr><td rowspan="4">Commands executed after assemble processing</td><td colspan="2">Specify 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.
%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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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 assemble processing.
The placeholders can be described in the script.
The specified command is displayed as the subproperty.</td></tr>
<tr><td>Default</td><td>Commands executed after assemble processing[number of defined items]</td></tr>
<tr><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
<tr><td rowspan="4">Other additional options</td><td colspan="2">Input the assemble option to be added additionally.
Add -asm= as required.
The options set here are added at the end of the assemble options group.</td></tr>
<tr><td>Default</td><td>Blank</td></tr>
<tr><td>How to change</td><td>Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.</td></tr>
<tr><td>Restriction</td><td>Up to 32767 characters</td></tr>
</table>

[Link Options] tab

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

Remark For details on compiler driver options, refer to "MULTI: Building Applications for Embedded V850 and RH850" released by Green Hills Software, LLC. Please do not contact Green Hills Software, LLC with any inquiries about the CCRH850 build-tool plugin of CS+.

Caution 1. This tab is not displayed for the library project.

Caution 2. This tab is displayed only when the current project has been created with [Empty Application(GHS CCRH850)] or [Library(GHS CCRH850)] selected as the project type.

[Description of each category]

(1) [Debug Information]
The detailed information on debug information is displayed and the configuration can be changed.

Debugging Level	Specify the -G option of the compiler driver.		
	Default	MULTI(-G)	
	How to change	Select from the drop-down list.	
	Restriction	MULTI(-G)	Specifies the -G option.
		Not specify(No option specified)	Does not specify the -G option.

(2) [Output File]
The detailed information on output files is displayed and the configuration can be changed.

Output folder	Specify the output folder. 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. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -o option of the compiler driver.	
	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. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectName%: Replaces with the main project name. %ProjectName%: Replaces with the project name. This corresponds to the -o option of the compiler driver.	
	Default	%ProjectName%
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

(3) [Library]
The detailed information on the library is displayed and the configuration can be changed.

Alternate Library Directory	Specify the -YL option of the compiler driver. 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. Uppercase characters and lowercase characters are not distinguished for the library directory names.	
	Default	Blank
	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

Libraries	Specify the -l option of the compiler driver. 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 library file name is displayed as the subproperty.	
	Default	Libraries[*number of defined items*]
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. -> Edit by the Specify Using Library File dialog box which appears when clicking the [Browse...] 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 libraries	The library files which CS+ sets during linking are displayed. This corresponds to the -l option of the compiler driver. The system library file name is displayed as the subproperty.	
	Default	System libraries[*number of defined items*]
	How to change	Changes not allowed

(4) [Hex Output]
The detailed information on hex output is displayed and the configuration can be changed.

Generate Additional Output	Specify the -srec or -hex option of the compiler driver.		
	Default	S-Record File(-srec)	
	How to change	Select from the drop-down list.	
	Restriction	S-Record File(-srec)	Specifies the -srec option.
		HEX386 File(-hex)	Specifies the -hex option.
		Not specify(No option specified)	Does not specify Generate Additional Output.

Output folder	Specify the folder which the hex file is output. 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. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -srec or -hex option of the compiler driver. This property is displayed only when other than [Not specify(No option specified)] in the [Generate Additional Output] 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
Output file name	Specify the hex file name. If the extension is omitted, “.run” is automatically added. The following placeholders are supported. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectName%: Replaces with the main project name. %ProjectName%: Replaces with the project name. This corresponds to the -srec or -hex option of the compiler driver. This property is displayed only when other than [Not specify(No option specified)] in the [Generate Additional Output] property is selected.	
	Default	%ProjectName%.run
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

(5) [List]
The detailed information on the list is displayed and the configuration can be changed.

Map File Generation	Specify the -map option of the compiler driver.		
	Default	Generate Default Map File(-map)	
	How to change	Select from the drop-down list.	
	Restriction	Generate Default Map File(-map)	Specifies the -map option.
		Not specify(No option specified)	Does not specify the -map option.

(6) [Others]
Other detailed information on linking is displayed and the configuration can be changed.

<table>
<tr><td rowspan="4">Start Files</td><td colspan="3">Specify the -nostartfiles option of the compiler driver.</td></tr>
<tr><td>Default</td><td colspan="2">Not specify(No option specified)</td></tr>
<tr><td>How to change</td><td colspan="2">Select from the drop-down list.</td></tr>
<tr><td rowspan="2">Restriction</td><td>Do Not Use Start Files(-nostartfiles)</td><td>Specifies the -nostartfiles option.</td></tr>
<tr><td>Not specify(No option specified)</td><td>Does not specify the -nostartfiles option.</td></tr>
<tr><td rowspan="4">Commands executed before link processing</td><td colspan="3">Specify 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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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.</td></tr>
<tr><td>Default</td><td colspan="2">Commands executed before link processing[number of defined items]</td></tr>
<tr><td>How to change</td><td colspan="2">Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td colspan="2">Up to 1023 characters
Up to 64 items can be specified.</td></tr>
</table>

Commands executed after link processing	Specify 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. %Options%: Replaces with the command line option under build execution. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the program name under execution. %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.	
	Default	Commands executed after link processing[*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 1023 characters Up to 64 items can be specified.
Other additional options	Input the link option 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 in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.
	Restriction	Up to 32767 characters

[Create Library Options] tab

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

Remark For details on compiler driver options, refer to "MULTI: Building Applications for Embedded V850 and RH850" released by Green Hills Software, LLC. Please do not contact Green Hills Software, LLC with any inquiries about the CCRH850 build-tool plugin of CS+.

Caution 1. This tab is not displayed for the library project.

Caution 2. This tab is displayed only when the current project has been created with [Empty Application(GHS CCRH850)] or [Library(GHS CCRH850)] selected as the project type.

[Description of each category]

(1) [Debug Information]
The detailed information on debug information is displayed and the configuration can be changed.

Debugging Level	Specify the -G option of the compiler driver.		
	Default	MULTI(-G)	
	How to change	Select from the drop-down list.	
	Restriction	MULTI(-G)	Specifies the -G option.
		Not specify(No option specified)	Does not specify the -G option.

(2) [Output File]
The detailed information on output files is displayed and the configuration can be changed.

Output folder	Specify the output folder. 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. If this is blank, it is assumed that the project folder has been specified. This corresponds to the -o option of the compiler driver.	
	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. %ActiveProjectName%: Replaces with the active project name. %BuildModeName%: Replaces with the build mode name. %MainProjectName%: Replaces with the main project name. %ProjectName%: Replaces with the project name. This corresponds to the -o option of the compiler driver.	
	Default	lib%ProjectName%.a
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

(3) [Library]
The detailed information on the library is displayed and the configuration can be changed.

Alternate Library Directory	Specify the -YL option of the compiler driver. 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. Uppercase characters and lowercase characters are not distinguished for the library directory names.	
	Default	Blank
	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

Libraries	Specify the -l option of the compiler driver. 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 library file name is displayed as the subproperty.	
	Default	Libraries[*number of defined items*]
	How to change	Edit by the Path Edit dialog box which appears when clicking the [...] button. -> Edit by the Specify Using Library File dialog box which appears when clicking the [Browse...] 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 libraries	The library files which CS+ sets during linking are displayed. This corresponds to the -l option of the compiler driver. The system library file name is displayed as the subproperty.	
	Default	System libraries[*number of defined items*]
	How to change	Changes not allowed

(4) [List]
The detailed information on the list is displayed and the configuration can be changed.

Map File Generation	Specify the -map option of the compiler driver.		
	Default	Not specify(No option specified)	
	How to change	Select from the drop-down list.	
	Restriction	Generate Default Map File(-map)	Specifies the -map option.
		Not specify(No option specified)	Does not specify the -map option.

(5) [Others]
Other detailed information on creating a library is displayed and the configuration can be changed.

<table>
<tr><td rowspan="4">Commands executed before create library processing</td><td colspan="2">Specify the command to be executed before library generation 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 generation 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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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 generation processing.
The placeholders can be described in the script.
The specified command is displayed as the subproperty.</td></tr>
<tr><td>Default</td><td>Commands executed before library generate processing[number of defined items]</td></tr>
<tr><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
</table>

<table>
<tr><td rowspan="4">Commands executed after create library processing</td><td colspan="2">Specify the command to be executed after library generation 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 generation 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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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 library generation processing.
The placeholders can be described in the script.
The specified command is displayed as the subproperty.</td></tr>
<tr><td>Default</td><td>Commands executed after library generate processing[number of defined items]</td></tr>
<tr><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
<tr><td rowspan="4">Other additional options</td><td colspan="2">Input the create library options to be added additionally.
The options set here are added at the end of the create library options group.</td></tr>
<tr><td>Default</td><td>Blank</td></tr>
<tr><td>How to change</td><td>Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.</td></tr>
<tr><td>Restriction</td><td>Up to 32767 characters</td></tr>
</table>

[I/O Header File Generation Options] tab

This tab shows the detailed information on the I/O header file generation tool categorized by the following and the configuration can be changed.

[Description of each category]

(1) [I/O Header File]
The detailed information on the I/O header file is displayed and the configuration can be changed.

Update I/O header file on build	Select whether to update the I/O header file at build. The I/O header file is updated when the device file is newer than that at generation of the I/O header file or properties related to generation of the I/O header file have been updated. Update is performed by automatic overwriting and a backup file with the bak extension is created. This contents are common to all the build modes.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(Checking the device file)	Updates the I/O header file when the device file has been updated at build.
		Yes(Checking the property)	Updates the I/O header file when the properties have been updated at build.
		Yes(Checking the device file and the property)	Updates the I/O header file when the device file or properties have been updated at build.
		No	Does not update the I/O header file at build.
Device file on generating I/O header file	The file name and version of the device file when the I/O header file was generated are displayed. Note that this property is displayed only when a choice other than [No] was made in the [Update I/O header file on build] property.		
	Default	*The file name and version of the device file when the I/O header file was generated*	
	How to change	Changes not allowed	
Current device file	The file name and version of the device file which is installed in the running CS+ environment are displayed. Note that this property is displayed only when a choice other than [No] was made in the [Update I/O header file on build] property.		
	Default	*Current device file*	
	How to change	Changes not allowed	

Select modules which are output in files	Select whether to select modules which are output to the I/O header file.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Displays the [Modules which are defined in device file] property to select modules which are output to the I/O header file. Only the I/O register in the selected module is output to the I/O header file.
		No	Outputs all modules to the I/O header file.
Modules which are defined in device file	The list of modules which are defined in the device file are displayed. The following items are displayed in the subproperty. Module name: The names of modules which are defined in the device file File name: The names of the I/O header files to which the modules are output Output: Whether to enable or disable output to the I/O header file This property is displayed only when [Yes] in the [Select modules which are output in files] property is selected.		
	Default	[*Total number of modules defined in device file*]	
	How to change	Edit by the Select Modules Which Are Output in Files dialog box which appears when clicking the [...] button. Editing by directly entering the subproperty is not allowed.	
	Restriction	Up to 259 characters	
Output definitions regarding µITRON	Select whether to output definitions regarding µITRON.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-uitron=on)	Outputs definitions regarding µITRON.
		No	Does not output definitions regarding µITRON.
Enable MISRA-C option	Select whether to output an I/O header file compatible with the MISRA-C rules.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-misra_c=on2)	Outputs an I/O header file compatible with the MISRA-C rules. Access to the I/O register allocated to the same address can be made in only the maximum access size because no union is output.
		No	The MISRA-C rules are not considered.

Enable module array option	Select whether definitions which become accessible in arrays are to be output to the header file for modules that have numbers starting with 0.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-modulearray=on)	Enables the module array option.
		No	Does not enable the module array option.
Define blocks in modules	Select whether to define blocks in modules. Note that this property is displayed only when there is an information file for defining blocks.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Defines blocks in a module to generate an array for the blocks and for the I/O registers.
		No	Does not define blocks in modules.
Enable IOR array option	Select whether definitions which become accessible in arrays are to be output to IORs that have numbers starting with 0.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes(-iorarray=on)	Enables the IOR array option.
		No	Does not enable the IOR array option.
Share definition of structure	Select whether to share definitions of structures.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Shares definitions of structures.
		No(-share_structure=off)	Does not share definitions of structures.

(2) [Others]
Other detailed information on the I/O header file is displayed and the configuration can be changed.

Other additional options	Input the I/O header file options to be added additionally. The options set here are added at the end of the I/O header file generation 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

[Build Settings] tab

This tab shows the detailed information on each C source file, assembly source file, link directive file, object file, and library file categorized by the following and the configuration can be changed.

(1)[Build]

[Description of each category]

(1) [Build]
The detailed information on the build is displayed and the configuration can be changed.

Set as build-target	Select whether to run a build of the selected file.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Runs a build of the selected file.
		No	Does not run a build of the selected file.
Set individual compile option	Select whether to set the compile option that differs from the project settings to the selected C source file. This property is displayed only when a C source file is selected on the project tree and [Yes] in the [Set as build-target] property is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Sets the option that differs from the project settings to the selected C source file.
		No	Does not set the option that differs from the project settings to the selected C source file.
Set individual assemble option	Select whether to set the assemble option that differs from the project settings to the selected assembly source file. This property is displayed only when the assembly source file is selected on the project tree and [Yes] in the [Set as build-target] property tab is selected.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Sets the option that differs from the project settings to the selected assembly source file.
		No	Does not set the option that differs from the project settings to the selected assembly source file.
File type	The type of the selected file is displayed.		
	Default	C source file (when the C source file is selected) Assembly source file (when the assembly source file is selected) Link directive (when the link directive file is selected) Object file (when the object file is selected) Library file (when the library file is selected)	
	How to change	Changes not allowed	

[Individual Compile Options] 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 [Common Options] tab and [Compile Options] tab.

When the settings are changed from these tabs, the properties are displayed in boldface.

(1)[Debug Information]
(2)[Optimization]
(3)[Preprocess]
(4)[Output Code]
(5)[Output File]
(6)[Others]

Remark 1. 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.

Remark 2. For details on compiler driver options, refer to "MULTI: Building Applications for Embedded V850 and RH850" released by Green Hills Software, LLC. Please do not contact Green Hills Software, LLC with any inquiries about the CCRH850 build-tool plugin of CS+.

Caution This tab is displayed only when the current project has been created with [Empty Application(GHS CCRH850)] or [Library(GHS CCRH850)] selected as the project type.

[Description of each category]

(1) [Debug Information]

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

Debugging Level	Specify the -G option of the compiler driver.		
	Default	*Configuration of the compile option*	
	How to change	Select from the drop-down list.	
	Restriction	MULTI(-G)	Specifies the -G option.
		Not specify(No option specified)	Does not specify the -G option.
Generate MULTI and Native Information	Specify the -dwarf2 option of the compiler driver.		
	Default	*Configuration of the compile option*	
	How to change	Select from the drop-down list.	
	Restriction	On(-dwarf2)	Specifies the -dwarf2 option.
		Not specify(No option specified)	Does not specify the -dwarf2 option.

(2) [Optimization]

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

Optimization Strategy	Specify the -O option of the compiler driver.		
	Default	*Configuration of the compile option*	
	How to change	Select from the drop-down list.	
	Restriction	Optimize for Debug-gability (-Odebug)	Specifies the -Odebug option.
		Optimize for General Use (-O)	Specifies the -O option.
		Optimize for Size (-Osize)	Specifies the -Osize option.
		Optimize for Speed (-Ospeed)	Specifies the -Ospeed option.
		No Optimizations (-Onone)	Specifies the -Onone option.
		Not specify(No option specified)	Does not specify Optimization Strategy.

(3) [Preprocess]
The detailed information on preprocessing is displayed and the configuration can be changed.

Include Directories	Specify the -I option of the compiler driver. 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. The specified include path is displayed as the subproperty. Uppercase characters and lowercase characters are not distinguished for the include paths.	
	Default	Include Directories[*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 enter directly in the text box.
	Restriction	Up to 247 characters Up to 256 items can be specified.

Use whole include paths specified for build tool	Select whether to compile using the include path specified in the [Include Directories] property in the [Preprocess] 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 [Include Directories] property from this tab - Paths specified in the [Include Directories] property from the [Compile Options] tab - Paths displayed in the [System include paths] property from the [Compile Options] tab This corresponds to the -I option of the compiler driver.		
	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.
Define Preprocessor Symbol	Specify the -D option of the compiler driver. Specify in the format of "*macro name=defined value*", with one macro name per line. The specified macro is displayed as the subproperty.		
	Default	Define Preprocessor Symbol[*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 256 characters Up to 256 items can be specified.	

(4) [Output Code]
The detailed information on output codes is displayed and the configuration can be changed.

Special Data Area	Specify the -sda and -zda options of the compiler driver.		
	Default	*Configuration of the compile option*	
	How to change	Select from the drop-down list.	
	Restriction	Small Data Area(-sda)	Specifies the -sda option.
		Small Data Area with Threshold(-sda=size)	Specifies the -sda=size option.
		Zero Data Area with Threshold(-zda=size)	Specifies the -zda=size option.
		Not specify(No option specified)	Does not specify Special Data Area.
Threshold Value	Specify Threshold Value of the -sda or -zda option of the compiler driver. This property is displayed when [Small Data Area with Threshold(-sda=size)] or [Zero Data Area with Threshold(-zda=size)] in the [Special Data Area] property is selected.		
	Default	*Configuration of the compile option*	
	How to change	Directly enter in the text box.	
	Restriction	Decimal number or blank	

23-bit SDA	Specify the -large_sda option of the compiler driver.		
	Default	*Configuration of the compile option*	
	How to change	Select from the drop-down list.	
	Restriction	Generate 23-bit SDA relocations for load/store instructions(-large_sda)	Specifies the -large_sda option.
		Not specify(No option specified)	Does not specify the -large_sda option.

(5) [Output File]
The detailed information on output files is displayed and the configuration can be changed.

Object file name	Specify the name of the object file generated after compilation. The extension other than ".o" cannot be specified. If the extension is omitted, ".o" is automatically added. If this is blank, the file name will be the source file name with the extension replaced by ".o". This corresponds to the -o option of the compiler driver.	
	Default	Blank
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

(6) [Others]
Other detailed information on compilation is displayed and the configuration can be changed.

Intermediate Forms of Output	Specify the -c option of the compiler driver.		
	Default	*Configuration of the compile option*	
	How to change	Select from the drop-down list.	
	Restriction	Object File(-c)	Specifies the -c option.
		Not specify(No option specified)	Does not specify the -c option.

<table>
<tr><td>Commands executed before compile processing</td><td colspan="2">Specify 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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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.</td></tr>
<tr><td></td><td>Default</td><td>Configuration of the compile option</td></tr>
<tr><td></td><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td></td><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
</table>

<table>
<tr><td rowspan="4">Commands executed after compile processing</td><td colspan="2">Specify 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.
%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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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 compile processing.
The placeholders can be described in the script.
The specified command is displayed as the subproperty.</td></tr>
<tr><td>Default</td><td>Configuration of the compile option</td></tr>
<tr><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
<tr><td rowspan="4">Other additional options</td><td colspan="2">Input the compile option to be added additionally.
The options set here are added at the end of the compile options group.</td></tr>
<tr><td>Default</td><td>Configuration of the compile option</td></tr>
<tr><td>How to change</td><td>Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.</td></tr>
<tr><td>Restriction</td><td>Up to 32767 characters</td></tr>
</table>

[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, [Compile Options] tab, and [Assemble Options] tab.

When the settings are changed from these tabs, the properties are displayed in boldface.

(1)[Debug Information]
(2)[Preprocess]
(3)[Output File]
(4)[Others]

Remark 1. This tab is displayed only when [Yes] in the [Set individual assemble option] property in the [Build] category from the [Build Settings] tab is selected.

Remark 2. For details on compiler driver options, refer to "MULTI: Building Applications for Embedded V850 and RH850" released by Green Hills Software, LLC. Please do not contact Green Hills Software, LLC with any inquiries about the CCRH850 build-tool plugin of CS+.

Caution This tab is displayed only when the current project has been created with [Empty Application(GHS CCRH850)] or [Library(GHS CCRH850)] selected as the project type.

[Description of each category]

(1) [Debug Information]
The detailed information on debug information is displayed and the configuration can be changed.

Debugging Level	Specify the -G option of the compiler driver.		
	Default	*Configuration of the assemble option*	
	How to change	Select from the drop-down list.	
	Restriction	MULTI(-G)	Specifies the -G option.
		Not specify(No option specified)	Does not specify the -G option.

(2) [Preprocess]
The detailed information on preprocessing is displayed and the configuration can be changed.

Include Directories	Specify the -I option of the compiler driver. 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. The specified include path is displayed as the subproperty. Uppercase characters and lowercase characters are not distinguished for the include paths.		
	Default	Include Directories[*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 enter directly in the text box.	
	Restriction	Up to 247 characters Up to 256 items can be specified.	
Use whole include paths specified for build tool	Select whether to assemble using the include path specified in the [Include Directories] property in the [Preprocess] category from the [Assemble Options] tab of the build tool to be used. The include paths are added by the following procedure. - Paths specified in the [Include Directories] property from this tab - Paths specified in the [Include Directories] property from the [Assemble Options] tab - Paths displayed in the [System include paths] property from the [Assemble Options] tab This corresponds to the -I option of the compiler driver.		
	Default	Yes	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Assembles 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.
Define Preprocessor Symbol	Specify the -D option of the compiler driver. Specify in the format of "*macro name=defined value*", with one macro name per line. The specified macro is displayed as the subproperty.		
	Default	Define Preprocessor Symbol[*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 256 characters Up to 256 items can be specified.	

(3) [Output File]
The detailed information on output files is displayed and the configuration can be changed.

Object file name	Specify the name of the object file generated after assembling. The extension other than ".o" cannot be specified. If the extension is omitted, ".o" is automatically added. If this is blank, the file name will be the source file name with the extension replaced by ".o". This corresponds to the -o option of the compiler driver.	
	Default	Blank
	How to change	Directly enter in the text box.
	Restriction	Up to 259 characters

(4) [Others]
Other detailed information on assembly is displayed and the configuration can be changed.

Intermediate Forms of Output	Specify the -c option of the compiler driver.		
	Default	*Configuration of the assemble option*	
	How to change	Select from the drop-down list.	
	Restriction	Object File(-c)	Specifies the -c option.
		Not specify(No option specified)	Does not specify the -c option.
Commands executed before assemble processing	Specify 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. %MicomToolPath%: Replaces with the absolute path of the install folder of this product. %Options%: Replaces with the command line option under build execution. %OutputDir%: Replaces with the absolute path of the output folder. %OutputFile%: Replaces with the absolute path of the output file. %Program%: Replaces with the program name under execution. %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.		
	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 enter directly in the text box.	
	Restriction	Up to 1023 characters Up to 64 items can be specified.	

<table>
<tr><td rowspan="4">Commands executed after assemble processing</td><td colspan="2">Specify 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.
%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.
%Options%: Replaces with the command line option under build execution.
%OutputDir%: Replaces with the absolute path of the output folder.
%OutputFile%: Replaces with the absolute path of the output file.
%Program%: Replaces with the program name under execution.
%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 assemble processing.
The placeholders can be described in the script.
The specified command is displayed as the subproperty.</td></tr>
<tr><td>Default</td><td>Configuration of the assemble option</td></tr>
<tr><td>How to change</td><td>Edit by the Text Edit dialog box which appears when clicking the [...] button.
For the subproperty, you can enter directly in the text box.</td></tr>
<tr><td>Restriction</td><td>Up to 1023 characters
Up to 64 items can be specified.</td></tr>
<tr><td rowspan="4">Other additional options</td><td colspan="2">Input the assemble option to be added additionally.
Add -asm= as required.
The options set here are added at the end of the assemble options group.</td></tr>
<tr><td>Default</td><td>Configuration of the assemble option</td></tr>
<tr><td>How to change</td><td>Directly enter in the text box or edit by the Character String Input dialog box which appears when clicking the [...] button.</td></tr>
<tr><td>Restriction</td><td>Up to 32767 characters</td></tr>
</table>

[Build Options] tab

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

Remark For details on compiler driver options, refer to "MULTI: Building Applications for Embedded V850 and RH850" released by Green Hills Software, LLC. Please do not contact Green Hills Software, LLC with any inquiries about the CCRH850 build-tool plugin of CS+.

Caution 1. This tab is displayed only when the current project has been created with [Using Existing GHS Project File(GHS CCRH850)] selected as the project type.

Caution 2. If you have upgraded the GHS compiler version, confirm the setting of the [Compiler package folder] property from the [Path of Tools] category.

[Description of each category]

(1) [Build Mode]
The detailed information on the build mode is displayed and the configuration can be changed.

Build mode	Select the build mode to be used during a 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	Runs a build with the default build mode that is set when a new project is created.
		Build mode that is added to the project	Runs a build with the build mode that is added to the project (other than Default-Build).
Change property value for all build modes at once	Select whether to reflect the value newly set to all build modes when a value is set in this property. Be careful since the value set may not be an appropriate value for other build modes.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Reflects the value newly set to all build modes when a value is set in this property.
		No	Does not reflect the value newly set to all build modes when a value is set in this property.

(2) [Build]
The detailed information on the build is displayed and the configuration can be changed.

MULTI project file to be built	Specify the MULTI project file referred when building a project. This is common to all the build modes. 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. Specify at least one of this property or [MULTI top project file to be built] property.	
	Default	Blank
	How to change	Directly enter in the text box or edit by the Specify MULTI Project File dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
MULTI top project file to be built	Specify the MULTI top project file referred when building a project. This is common to all the build modes. 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. Specify at least one of this property or [MULTI project file to be built] property. This corresponds to the -top option of the gbuild command.	
	Default	Blank
	How to change	Directly enter in the text box or edit by the Specify MULTI Project File dialog box which appears when clicking the [...] button.
	Restriction	Up to 259 characters
Additional option on cleaning	Specify the additional option to the command line when cleaning a project. This is common to all the build modes.	
	Default	-clean
	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

(3) [Build Method]
The detailed information on the build method is displayed and the configuration can be changed.

Build in parallel	Select whether to enable the parallel build facility. The parallel build facility enables CS+ to compile/assemble multiple source files in parallel using all processors mounted on the computer. This speeds up compilation/assemble. In addition, parallel build between projects can be set by selecting [Tool] menu >> [Option] and then making a setting in the [General - Build] category of the Option dialog box. This corresponds to the -parallel option of the gbuild command.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Enables the parallel build facility.
		No	Disables the parallel build facility.

(4) [Path of Tools]
The detailed information on the path of tools is displayed and the configuration can be changed.

Compiler package folder	Specify the folder of the compiler package to be used. This is common to all the build modes.	
	Default	*Value-according-to-installation-environment*
	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
Executable file name for building	Specify the executable file name for building. This is common to all the build modes.	
	Default	gbuild.exe
	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

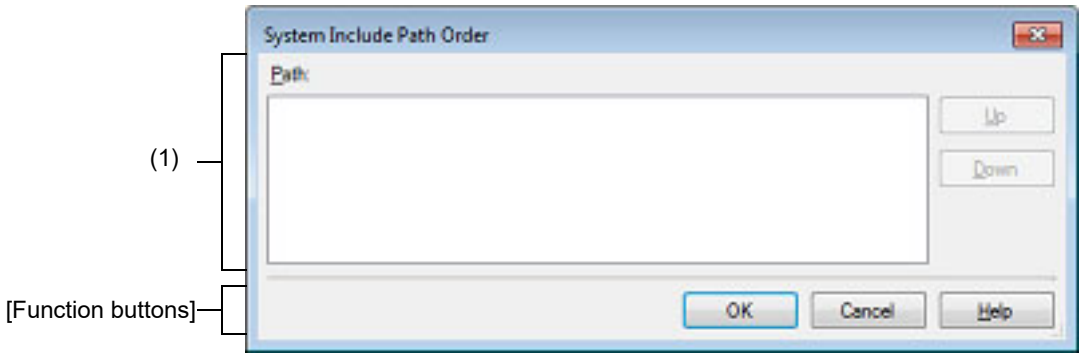
(5) [Notes]
The detailed information on notes is displayed and the configuration can be changed.

Memo	Add memos to the build tool. Add one item in one line. This setting is common to all the build modes. The specified memo is displayed as the subproperty.	
	Default	Memo[*number-of-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 256 characters Up to 256 items can be specified.

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.2 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 [Preprocess] category
  - From the [Assemble Options] tab, [System include paths] in the [Preprocess] 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.

Remark 1. Move the mouse cursor over a file name to display a tooltip with the absolute path of that file.

Remark 2. Newly added system include paths are added next to the last path of the list.

Remark 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.

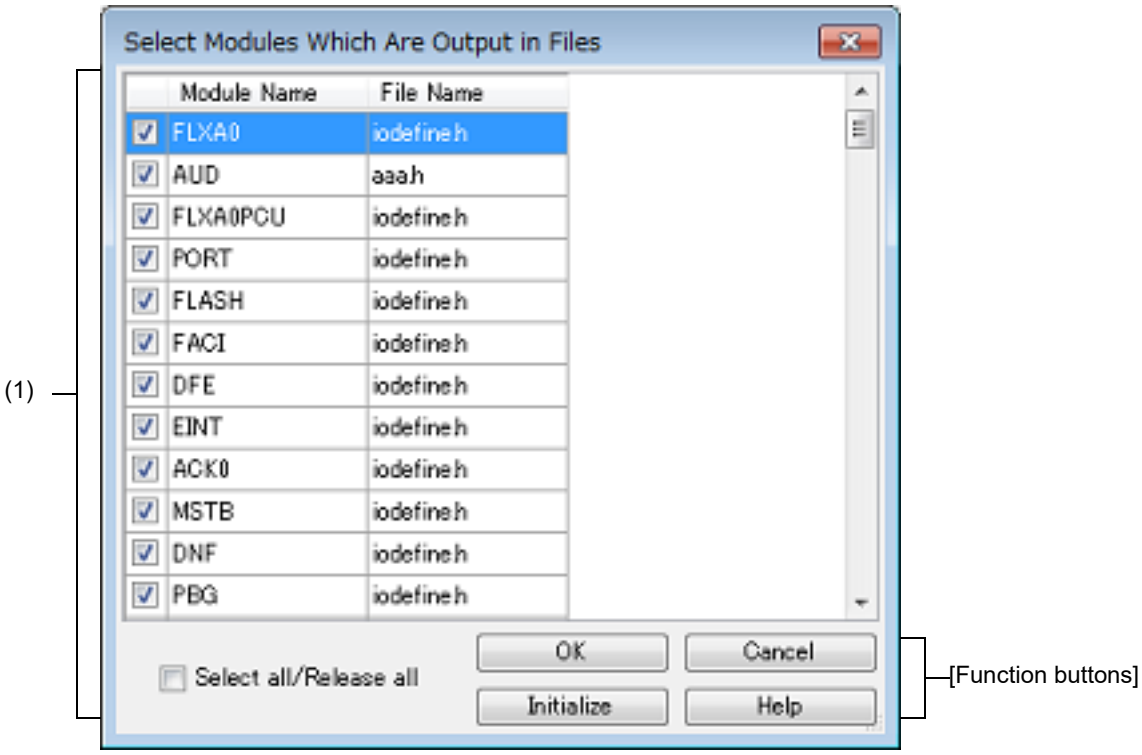
[Function buttons]

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.

Select Modules Which Are Output in Files dialog box

This dialog box is used to set modules which are output to the I/O header file.

Figure A.3 Select Modules Which Are Output in Files 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 [I/O Header File Generation Options] tab, [Modules which are defined in device file] in the [I/O Header File] category

[Description of each area]

(1) Area for specifying module name/file name
This area displays a list of the names of modules defined in the device file and the names of the I/O header files to which the modules are output.

(a) [Module Name]
This area displays the names of modules which are defined in the device file.
If any of the check boxes is selected, the relevant module is output to the I/O header file shown under [File Name].
The check boxes are selected by default.

(b) [File Name]
This area displays the names of the I/O header files to which the modules are output.
You can also directly enter a desired file name to change the I/O header file to which the module is output.
The default file name is “iodefine.h”.

(c) [Select all/Release all]
This check box is used to select or deselect all of the check boxes under [Module Name].
If this check box is selected, check boxes under [Module Name] will all be selected.
If this check box is deselected, selection of the check boxes under [Module Name] will all be cleared.

Remark When the same file name is specified for multiple modules, code for those modules is output to the same file. There is no case sensitivity for file names.

[Function buttons]

Button	Function
OK	Closes this dialog box and calls the settings to reflect them in the previous properties.
Initialize	Sets all check boxes under [Module Name] and [File Name] settings to their default values.
Cancel	Cancels the settings and closes this dialog box.
Help	Displays the help of this dialog box.

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Nov 01, 2020	-	First Edition issued

CS+ V8.05.00 User's Manual:
GHS CCRH850 Build Tool Operation

Publication Date: Rev.1.00 Nov 01, 2020
Published by: Renesas Electronics Corporation

CS+ V8.05.00

Renesas Electronics Corporation

R20UT4853EJ0100