User's Manual

CubeSuite+ V2.01.00

Integrated Development Environment

User's Manual: RH850 Debug

Target Device

RH850 Family

Renesas Electronics
www.renesas.com

Rev.1.00 Sep 2013

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

How to Use This Manual

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

CubeSuite+ 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 CubeSuite+ and design software and hardware application systems.

Purpose

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

Organization

This manual can be broadly divided into the following units.

CHAPTER 1 GENERAL
CHAPTER 2 FUNCTIONS
APPENDIX A WINDOW REFERENCE
APPENDIX B INDEX

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electricity, logic circuits, and microcontrollers.

Conventions

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

Related Documents

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

Document Name		Document No.
CubeSuite+ Integrated Development Environment User's Manual	Start	R20UT2682E
	RX Design	R20UT2683E
	V850 Design	R20UT2134E
	R8C Design	R20UT2135E
	RL78 Design	R20UT2684E
	78K0R Design	R20UT2137E
	78K0 Design	R20UT2138E
	RH850 Coding	R20UT2584E
	RX Coding	R20UT2470E
	V850 Coding	R20UT0553E
	Coding for CX Compiler	R20UT2659E
	R8C Coding	R20UT0576E
	RL78,78K0R Coding	R20UT2140E
	78K0 Coding	R20UT2141E
	RH850 Build	R20UT2585E
	RX Build	R20UT2472E
	V850 Build	R20UT0557E
	Build for CX Compiler	R20UT2142E
	R8C Build	R20UT0575E
	RL78,78K0R Build	R20UT2143E
	78K0 Build	R20UT0783E
	RH850 Debug	This manual
	RX Debug	R20UT2702E
	V850 Debug	R20UT2446E
	R8C Debug	R20UT0770E
	RL78 Debug	R20UT2445E
	78K0R Debug	R20UT0732E
	78K0 Debug	R20UT0731E
	Analysis	R20UT2686E
	Message	R20UT2687E

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

TABLE OF CONTENTS

CHAPTER 1 GENERAL

CubeSuite+ is a platform of an integrated developing environment for RH850 family, RX family, V850 family, R8C family (Localised support), RL78 family, 78K0R microcontrollers, 78K0 microcontrollers.

CubeSuite+ can run all the operations needed for developing the programs such as designing, cording, building, debugging, and flash programming.

In this manual, the debugging is explained out of those operations needed for the program development.

In this chapter, an overview of debugging products that CubeSuite+ provides is explained.

Remark Localised support
"Localised support " refers to specific regions support only.
CubeSuite+ for R8C (including NC30) is shipped and supported to the following regions only.
- Renesas Electronics Hong Kong Limited
- Renesas Electronics (China) Co., Ltd.
- Renesas Electronics (Shanghai) Co., Ltd.

1.1 Summary

You can effectively debug/simulate the program developed for the RH850 family, using the debugger which CubeSuite+ provides.

1.2 Features

The following are the features of the debugger provided by CubeSuite+.

- Synchronous execution and synchronous break in a microcontroller that supports multi-core
  Synchronous execution and synchronous break are available when the target microcontroller supports multi-core. Information regarding a desired core is displayed on the panel by switching the core selection.

- Connecting to the various debug tools
  A pleasant debugging environment for target systems is provided by connecting to the full-spec emulator (Full-spec emulator), the on-chip debugging emulator (E1/E20) and Simulator.

- C source text and disassembled text are shown mixed
  The C source text and the disassembled text are shown mixed on the same panel.

- Source level debugging and instruction level debugging
  The source level debugging and the instruction level debugging for a C source program can be done.

- Support of flash self programming emulation (Code flash)
  The code flash can be rewrited by using the flash self library of the flash self programming function.

- Real-time display update function
  The contents of memory, registers and variables are automatically updated not only when the program execution is stopped, but also in execution.

- Save/restore the debugging environment
  The debugging environment such as breakpoints, event configuration information, file download information, display condition/position of the panel, etc. can be saved.

CHAPTER 2 FUNCTIONS

This chapter describes a debugging process of CubeSuite+ and main functions for debugging.

2.1 Overview

The basic debugging sequence for programs using CubeSuite+ is as follows:

(1) Start CubeSuite+

Launch CubeSuite+ from the [Start] menu of Windows.

Remark For details on "Start CubeSuite+", see "CubeSuite+ Integrated Development Environment User's Manual: Start".

(2) Set a project

Create a new project, or load an existing one.

Remark For details on "Set a project", see "CubeSuite+ Integrated Development Environment User's Manual: Start".

(3) Create a load module

Create a load module by running a build after setting of the active project and the build tool to be used.

Remark For details on "Create a load module" with CC-RH, see "CubeSuite+ Integrated Development Environment User's Manual: RH850 Build".

(4) Confirm the connection to a host machine

Connect the debug tool (Full-spec emulator, E1, E20, or Simulator) to be used to a host machine.

(5) Select the debug tool to use

Select the debug tool to be used in a project.

Remark The selectable debug tool differs depending on the microcontroller type to be used in a project.

(6) Configure the operating environment of the debug tool

Configure the operating environment of the debug tool selected in steps (5).

- [Full-spec emulator]
- [E1]
- [E20]
- [Simulator]

(7) Connect to the debug tool

Connect the debug tool to CubeSuite+ to start communication.

(8) Execute downloading

Download the load module created in steps (3) to the debug tool.

(9) Display source files

Display the contents of the downloaded load module (source files) on the Editor panel or Disassemble panel.

(10) Execute programs

Execute the program by using the operation method corresponding to a purpose.
If you wish to stop the program at the arbitrary position, set a breakpoint/break event[Note] before executing the program (see "2.9.3 Stop the program at the arbitrary position (breakpoint)", "2.9.4 Stop the program at the arbitrary position (break event)", or "2.9.5 Stop the program with the access to variables/I/O registers").

Note These functions are implemented by setting events to the debug tool used.
See "2.16.6 Notes for setting events", when you use events.

Remark When the selected microcontroller version supports multi-core, select a core (PE: Processer Element) to be debugged before executing the program (see "2.7 Select a Core (PE)").

(11) Stop the program manually

Stop the program currently being executed.
Note that if a breakpoint/break event has been set in steps (10), the program execution will be stopped automatically when the set break condition is met.

(12) Check the result of the program execution

Check the following information that the debug tool acquired by the program execution.

- Display/Change the Memory, Register and Variable
- Display Information on Function Call from Stack
- Collect Execution History of Programs[Note]
- Measure Execution Time of Programs[Note]
- Measure Coverage [Simulator]

Note These functions are implemented by setting events to the debug tool used.
See "2.16.6 Notes for setting events", when you use events.

Debug the program, repeating steps (9) to (12) as required.
Note that if the program is modified during debugging, steps (3) and (8) also should be repeated.

Remarks 1. Other than the above, you can also check the result of the program execution by using the following functions.

- Set an Action into Programs
- Use Hook Function

2. The acquired information can be saved to a file.

- Save the disassembled text contents
- Save the memory contents
- Save the CPU register contents
- Save the I/O register contents
- Save the contents of local variables
- Save the contents of watch-expressions
- Save the contents of call stack information
- Save the contents of execution history

(13) Execute uploading

Save the program (the memory contents) to a file in the arbitrary format (e.g. Intel HEX file, Motorola S-record file, binary file, and etc.), as required.

(14) Disconnect from the debug tool

Disconnect the debug tool from CubeSuite+ to terminate communication.

(15) Save the project file

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

Remark For details on "Save the project file", see "CubeSuite+ Integrated Development Environment User's Manual: Start".

2.2 Preparation before Debugging

This section describes the preparation to start debugging the created program.

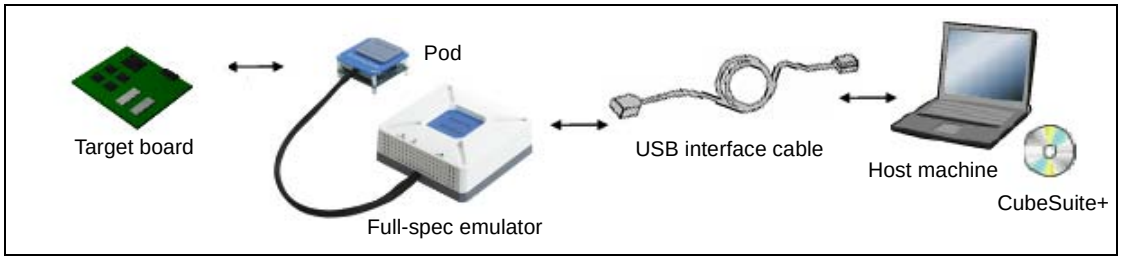
2.2.1 Confirm the connection to a host machine

Connection examples for each debug tool are shown.

(1) [Full-spec emulator]

Connect a host machine and Full-spec emulator. If required, connect a target board, too.
For details on the connection method, see the user's manual for Full-spec emulator.

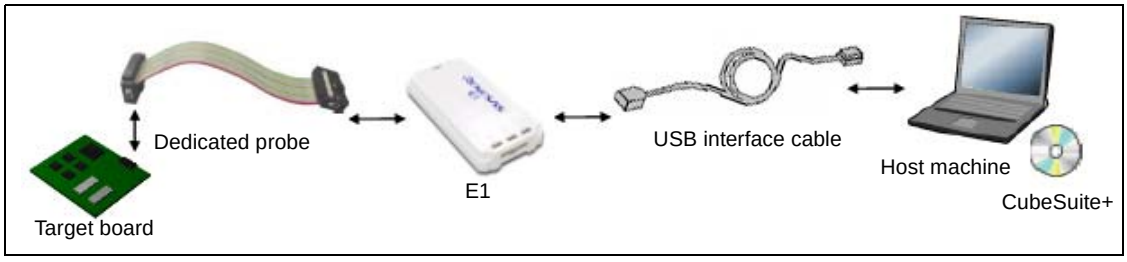
Figure 2-1. Connection Example [Full-spec emulator]



(2) [E1]

Connect a host machine and E1. If required, connect a target board, too.
For details on the connection method, see the user's manual for E1.

Figure 2-2. Connection Example [E1]



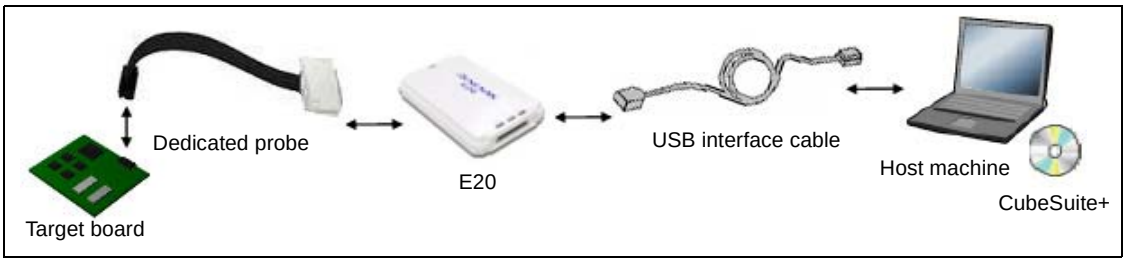
Cautions 1. Only the Low Pin Debug interface (hereafter referred to as LPD communications) is supported for communication with the target board.

2. For details on the connection using a debug MCU board, see the user's manual for debug MCU board.

(3) [E20]

Connect a host machine and E20. If required, connect a target board, too.
For details on the connection method, see the user's manual for E20.

Figure 2-3. Connection Example [E20]



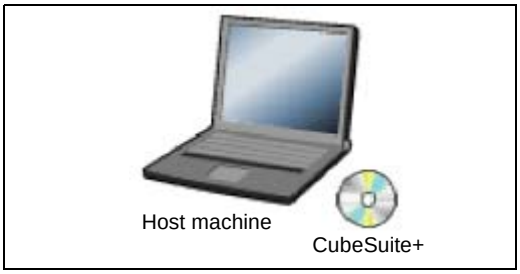
Cautions 1. Only the Low Pin Debug interface (hereafter referred to as LPD communications) is supported for communication with the target board.

2. For details on the connection using a debug MCU board, see the user's manual for debug MCU board.

(4) [Simulator]

A host machine is only needed for debugging (emulators are not needed).

Figure 2-4. Connection Example [Simulator]



2.3 Configuration of Operating Environment of the Debug Tool

This section describes the configuration of the operating environment for each debug tool.

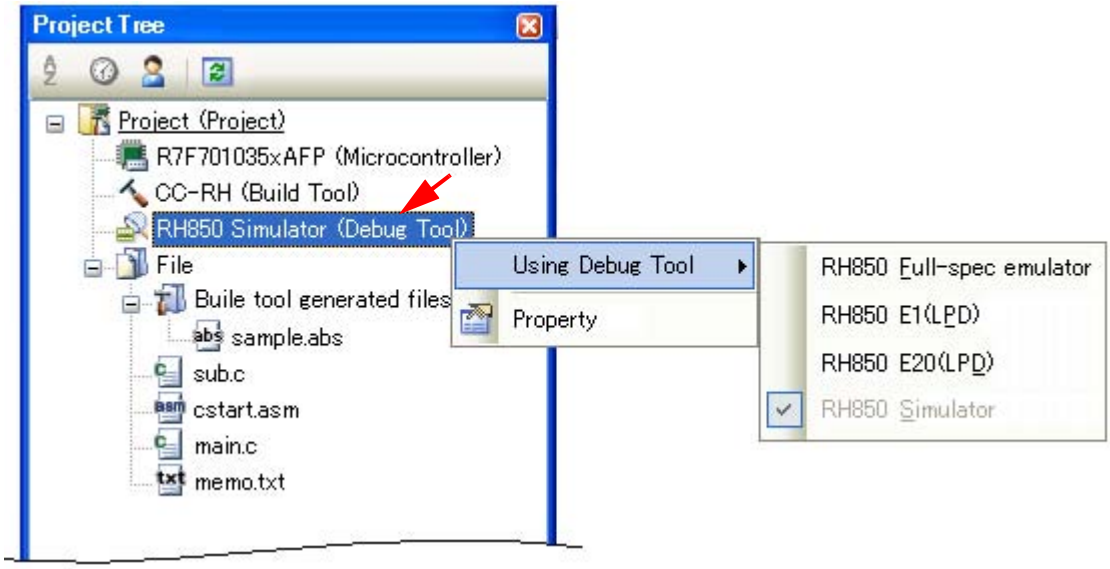
2.3.1 Select the debug tool to use

You can configure the operating environment in the Property panel corresponding to the debug tool to use.

Therefore, first, select the debug tool to be used in a project (the debug tool to be used can be specified in the individual main projects/subprojects).

To select or switch the debug tool, use the context menu shown by right clicking on the [*Microcontroller type Debug tool name* (Debug Tool)] node on the Project Tree panel.

Figure 2-5. Select/Switch Debug Tool to Use



Caution The context menu items displayed differ depending on the microcontroller selected in the project.

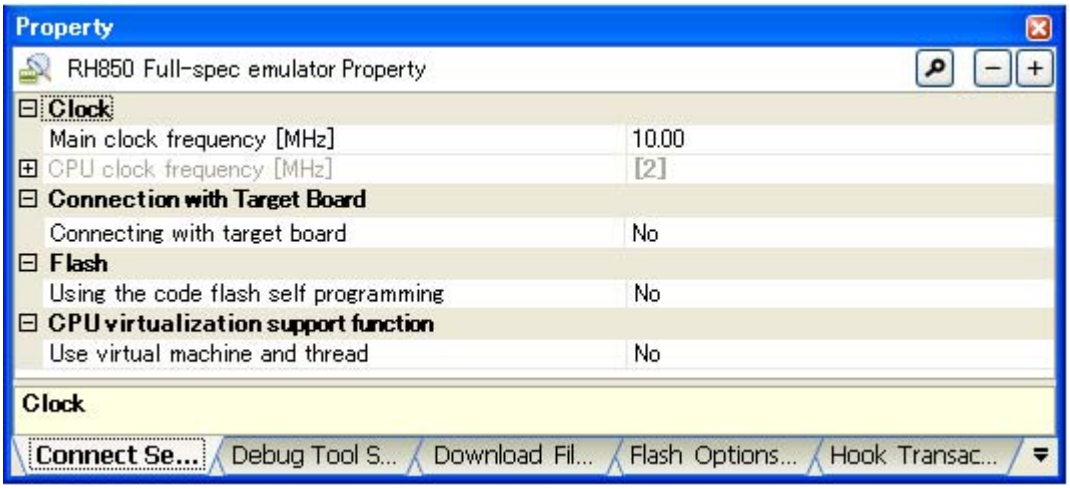
If the Property panel is already open, click the [*Microcontroller type Debug tool name* (Debug Tool)] node again. The view switches to the Property panel of the selected debug tool.

If the Property panel is not open, double-click the above mentioned node to open the corresponding Property panel.

2.3.2 [Full-spec emulator]

Configure the operating environment on the Property panel below when using Full-spec emulator.

Figure 2-6. Example of Property Panel [Full-spec emulator]



Follow the steps below by selecting the corresponding tab on the Property panel.

(1) [Connect Settings] tab
(2) [Debug Tool Settings] tab
(3) [Download File Settings] tab
(4) [Flash Options Settings] tab
(5) [Hook Transaction Settings] tab

(1) [Connect Settings] tab

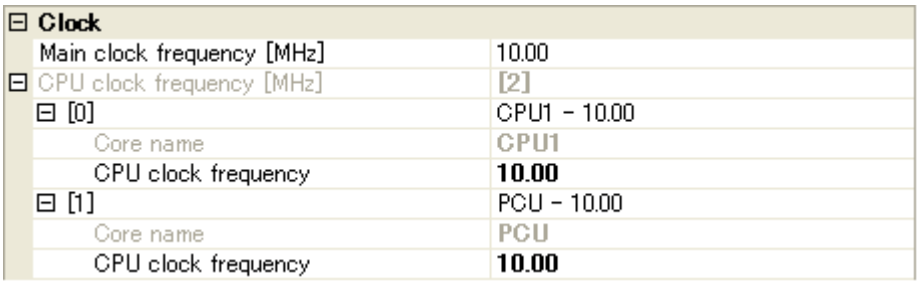
You configure the connection with the debug tool for each one of the following categories.

(a) [Clock]
(b) [Connection with Target Board]
(c) [Flash]
(d) [CPU virtualization support function]

(a) [Clock]

You can configure the clock.

Figure 2-7. [Clock] Category [Full-spec emulator]



<1> [Main clock frequency [MHz]]

Specify the main clock frequency (before multiplication).
You can specify the frequency from the drop-down list or by directly entering a frequency value between 0.001 and 999.999 (unit: MHz) (default: [10.00]).

<2> [CPU clock frequency [MHz]]

Specify the CPU clock frequency for each core.
The names of cores incorporated in the selected microcontroller are displayed as subproperties of this property. Specify the CPU clock frequency of each core by directly entering a frequency value between 0.001 and 999.999 (unit: MHz).
The number of subproperties displayed here differs with the selected microcontroller.

Remark The CPU clock frequency is used to convert the time stamp information for a trace to an actual time.

(b) [Connection with Target Board]

You can configure the connection between Full-spec emulator and the target board.

Caution Properties in this category cannot be changed when CubeSuite+ is connected to Full-spec emulator.

Figure 2-8. [Connection with Target Board] Category [Full-spec emulator]

<1> [Connecting with target board]

Select whether the target board is connected to Full-spec emulator or not.
Select [Yes] when the target board is connected to Full-spec emulator (default: [No]).

(c) [Flash]

You can configure the flash self programming function.
Note that this category appears only when the selected microcontroller supports the flash self programming function.

Caution Properties in this category cannot be changed when CubeSuite+ is connected to Full-spec emulator.

Figure 2-9. [Flash] Category

<1> [Using the code flash self programming]

Select whether to rewrite the code flash by using the flash self library of the flash self programming function.
Select [Yes] to rewrite the code flash (default: [No]).
Note that if [Yes] is selected in this property, the code flash will not be cashed.

(d) [CPU virtualization support function]

The property in this category is always disabled.

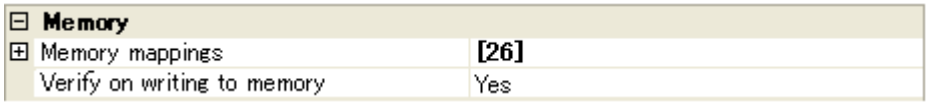
(2) [Debug Tool Settings] tab

You configure the basic settings of the debug tool for each one of the following categories.

(a) [Memory]

You can configure the memory.

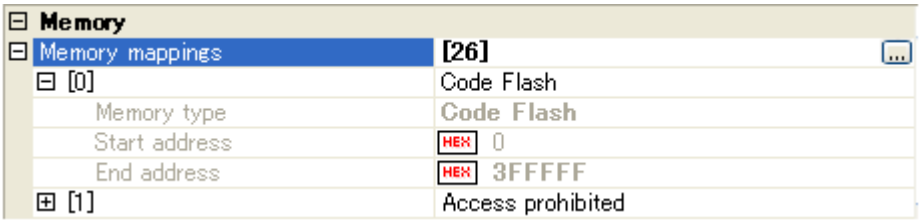
Figure 2-10. [Memory] Category [Full-spec emulator]



<1> [Memory mappings]

Current memory mapping status is displayed for each type of memory area by expanding this property.

Figure 2-11. Detailed Display of Memory Mapping



Caution The memory mapping cannot be added/deleted.

Remark When the selected microcontroller supports multi-core, this property displays the memory mapping status regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

<2> [Verify on writing to memory]

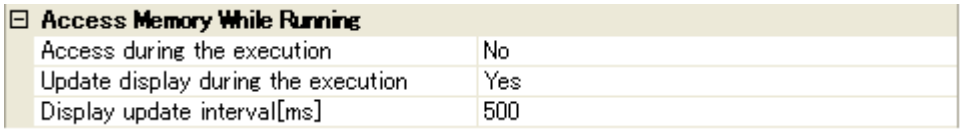
Select whether to perform a verify check when writing to the memory.

Select [Yes] to perform verification after download or when values are changed in the Watch panel/ Memory panel (default).

(b) [Access Memory While Running]

You can configure the memory access while executing a program (the real-time display update function). See "(4) Display/modify the memory contents during program execution" for details on the real-time display update function.

Figure 2-12. [Access Memory While Running] Category [Full-spec emulator]



<1> [Access during the execution]

Select whether to allow access to the internal RAM area during execution of a program.
Select [Yes] to allow access (default: [No]).

<2> [Update display during the execution]

Select whether to automatically update the contents in the Watch panel/Memory panel display during execution of a program.
Select [Yes] to update the display (default).

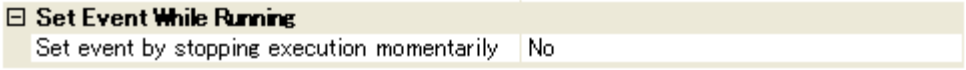
<3> [Display update interval[ms]]

This property appears only when the [Update display during the execution] property is set to [Yes].
Specify the interval to automatically update the contents in the Watch panel/Memory panel display during execution of a program.
Directly specify the Integer number between 100 and 65500 (rounding up the fractions less than 100ms) (default: [500]).

(c) [Set Event While Running]

You can configure the setting of events while executing a program in this category.

Figure 2-13. [Set Event While Running] Category



<1> [Set event by stopping execution momentarily]

Select whether to forcibly pause the execution for events that cannot be set while executing a program or operating the tracer/timer.
For details on the event types that are affected by this property, see "(2) Event types that can be set and deleted during execution".
Select [Yes] to set events above while execution (default: [No]).

(d) [Break]

You can configure the break function.
See "2.9 Stop Programs (Break)" for details on the break function and this category configuration.

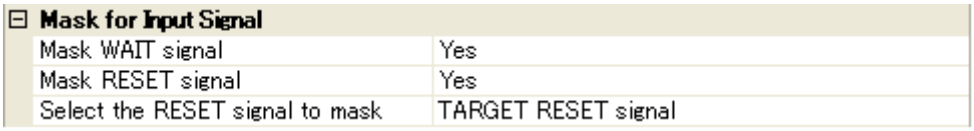
(e) [Trace]

You can configure the trace function.
See "2.12 Collect Execution History of Programs" for details on the trace function and this category configuration.

(f) [Mask for Input Signal]

You can configure the input signal masking.

Figure 2-14. [Mask for Input Signal] Category [Full-spec emulator]



<1> [Mask WAIT signal]

Select whether to mask the WAIT signal.

Select [Yes] so that the WAIT pin signal is not input to Full-spec emulator (default: [No]).

Caution If the [Connecting with target board] property in the [Connect Setting] tab is set to [No], this property is fixed to [Yes] (changes not allowed).

<2> [Mask RESET signal]

Select whether to mask the RESET signal.

Select [Yes] so that the RESET pin signal is not input to Full-spec emulator (default: [No]).

Caution If the [Connecting with target board] property in the [Connect Setting] tab is set to [No], this property is fixed to [Yes] (changes not allowed).

<3> [Select the RESET signal to mask]

This property appears only when the [Mask RESET signal] property is set to [Yes].

Select the type of RESET signal to be masked, from the following drop-down list.

- TARGET RESET signal (default)
- TARGET RESET signal and INTERNAL RESET signal

(3) [Download File Settings] tab

You can configure downloading to the debug tool.

See "2.5.1 Execute downloading" for details on each category configuration.

(4) [Flash Options Settings] tab

You can configure options for the flash memory incorporated in the microcontroller.

Note that this tab appears only when the selected microcontroller supports the flash options.

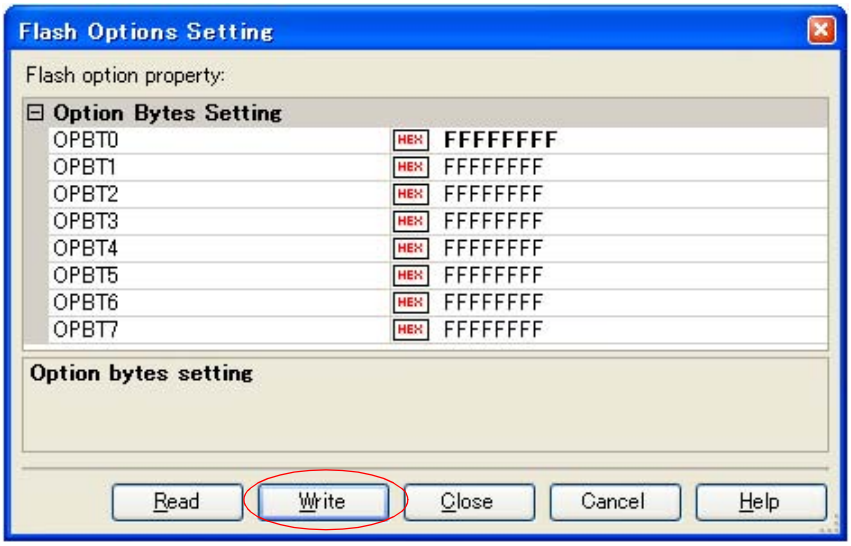
To configure options, specify the corresponding items on the Flash Options Setting dialog box [Full-spec emulator][E1][E20], that is opened by clicking the [...] button appears at the right of the field by selecting the [Flash options] property in the [Flash Options] category on this tab (the [...] button appears only while connected to the debug tool).

Click the [Write] button on this dialog box after specifying each item.

See the Flash Options Setting dialog box [Full-spec emulator][E1][E20] for details on the configuration.

Figure 2-15. Opening Flash Options Setting Dialog Box

Figure 2-16. Flash Options Settings (Flash Options Setting Dialog Box)



(5) [Hook Transaction Settings] tab

You can configure hook transaction for the debug tool.

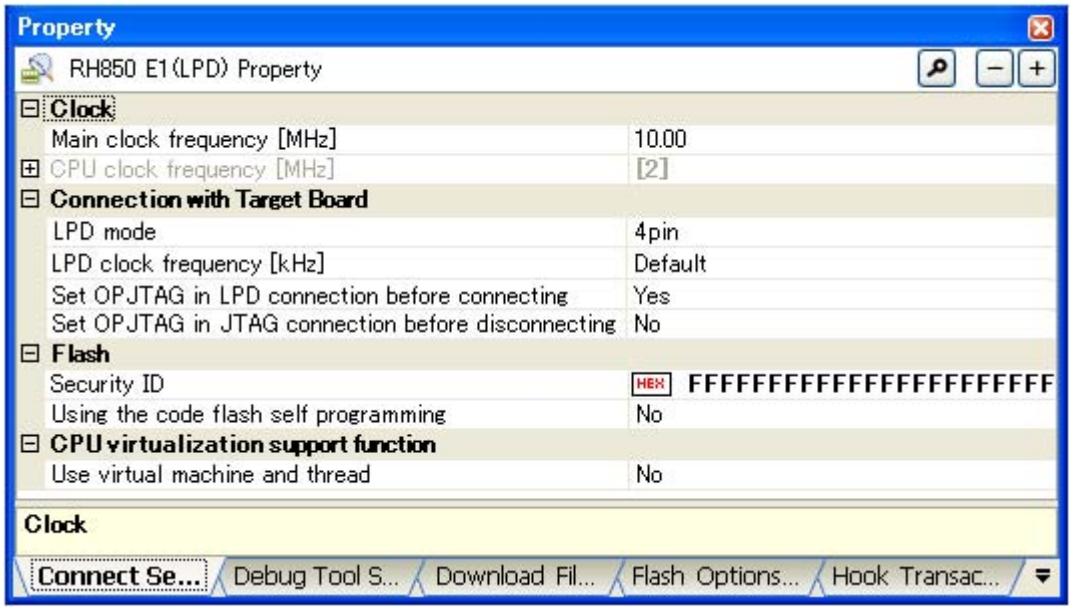
See "2.17 Use Hook Function" for details on each category configuration and the function of the hook transaction.

2.3.3 [E1]

Configure the operating environment on the Property panel below when using E1.

Caution Only LPD communications are supported for communication with the target board.

Figure 2-17. Example of Property Panel [E1]



Follow the steps below by selecting the corresponding tab on the Property panel.

(1) [Connect Settings] tab
(2) [Debug Tool Settings] tab
(3) [Download File Settings] tab
(4) [Flash Options Settings] tab
(5) [Hook Transaction Settings] tab

(1) [Connect Settings] tab

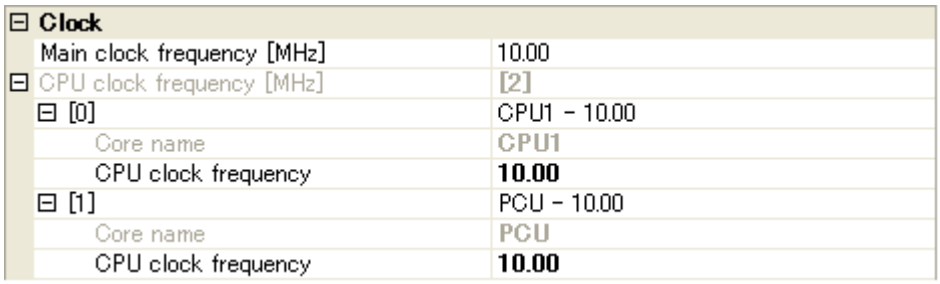
You configure the connection with the debug tool for each one of the following categories.

(a) [Clock]
(b) [Connection with Target Board]
(c) [Flash]
(d) [CPU virtualization support function]

(a) [Clock]

You can configure the clock.

Figure 2-18. [Clock] Category [E1]



<1> [Main clock frequency [MHz]]

Specify the main clock frequency (before multiplication).

You can specify the frequency from the drop-down list or by directly entering a frequency value between 0.001 and 999.999 (unit: MHz) (default: [10.00]).

<2> [CPU clock frequency [MHz]]

Specify the CPU clock frequency for each core.

The names of cores incorporated in the selected microcontroller are displayed as subproperties of this property. Specify the CPU clock frequency of each core by directly entering a frequency value between 0.001 and 999.999 (unit: MHz).

The number of subproperties displayed here differs with the selected microcontroller.

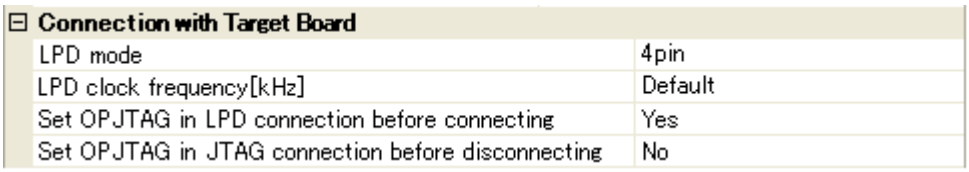
Remark The CPU clock frequency is used to convert the time stamp information for a trace to an actual time.

(b) [Connection with Target Board]

You can configure the connection between E1 and the target board.

Caution Properties in this category cannot be changed when CubeSuite+ is connected to E1.

Figure 2-19. [Connection with Target Board] Category [E1]



<1> [LPD mode]

Select LPD communication mode to use.

The selectable pin values differ depending on the selected microcontroller.

Note, however, that this property value cannot be changed when only one communication mode is available.

<2> [Baud rate[Kbps]]

This property appears only when the [LPD mode] property is set to [1pin].

Select the baud rate for LPD communication (default: [500]).

<3> [LPD clock frequency[kHz]]

This property appears only when the [LPD mode] property is set to [4pin].
Select the clock frequency for the LPD communication (default: [Default]).
When [Default] is selected, the default value specific to the microcontroller is used in connection to the target board.

<4> [Set OPJTAG in LPD connection before connecting]

Select whether to start up the microcontroller in serial programming mode upon connection to the debug tool and change the option byte settings to select LPD connection.
When [Yes] is selected, the debug tool starts up the microcontroller in serial programming mode upon its connection to CubeSuite+. The debug tool then checks the OPJTAG byte and, if LPD is not selected, changes the setting to select LPD. After that, the microcontroller enters debugging mode (default).
When [No] is selected, the debug tool starts up the microcontroller in debugging mode upon its connection to CubeSuite+. The debug tool then checks OPJTAG and, if LPD is not selected, shows a message dialog box.

<5> [Set OPJTAG in JTAG connection before disconnecting]

This property can be changed only when the [Set OPJTAG in LPD connection before connecting] property is set to [Yes].
Select whether to change the option byte settings to select JTAG connection before disconnection of the debug tool.
If you wish to change the option byte settings to select JTAG connection before disconnecting the debug tool, select [Yes].
When [No] (the default setting) is selected, the option byte settings are not changed before the debug tool is disconnected. In this case, LPD mode is applicable as the pin mode.

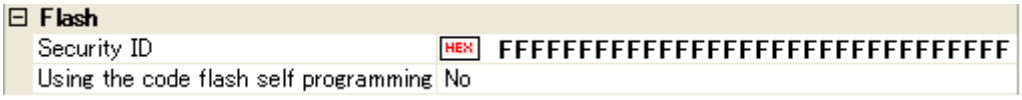
Remark On connection to E1, CubeSuite+ changes the option byte settings if LPD is not selected. For this reason, connecting and disconnecting E1 may change the value of the option bytes.

(c) [Flash]

You can configure the flash memory rewriting.

Caution Properties in this category cannot be changed when CubeSuite+ is connected to E1.

Figure 2-20. [Flash] Category [E1]



<1> [Security ID]

This property appears only when the selected microcontroller supports the ROM security function for flash memory.
Specify the key code for ID authentication when reading the code from the internal ROM or internal flash memory.
Directly enter 32 digits hexadecimal number (16 bytes) (default: [FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF]).

<2> [Using the code flash self programming]

Select whether to rewrite the code flash by using the flash self library of the flash self programming function.

Select [Yes] to rewrite the code flash (default: [No]).

Note that if [Yes] is selected in this property, the code flash will not be cashed.

(d) [CPU virtualization support function]

The property in this category is always disabled.

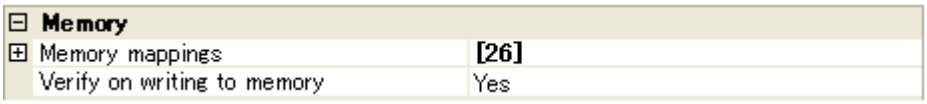
(2) [Debug Tool Settings] tab

You configure the basic settings of the debug tool for each one of the following categories.

(a) [Memory]

You can configure the memory.

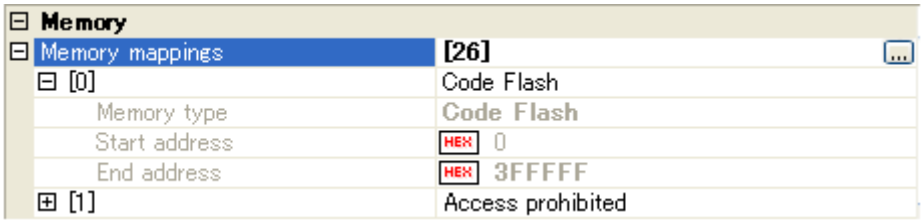
Figure 2-21. [Memory] Category [E1]



<1> [Memory mappings]

Current memory mapping status is displayed for each type of memory area by expanding this property.

Figure 2-22. Detailed Display of Memory Mapping



Caution The memory mapping cannot be added/deleted.

Remark When the selected microcontroller supports multi-core, this property displays the memory mapping status regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

<2> [Verify on writing to memory]

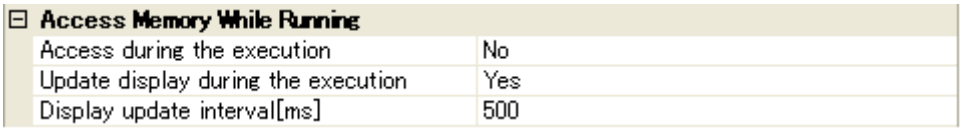
Select whether to perform a verify check when writing to the memory.

Select [Yes] to perform verification after download or when values are changed in the Watch panel/ Memory panel (default).

(b) [Access Memory While Running]

You can configure the memory access while executing a program (the real-time display update function). See "(4) Display/modify the memory contents during program execution" for details on the real-time display update function.

Figure 2-23. [Access Memory While Running] Category [E1]



<1> [Access during the execution]

Select whether to allow access to the internal RAM area during execution of a program.
Select [Yes] to allow access (default: [No]).

<2> [Update display during the execution]

Select whether to automatically update the display in the Watch panel/Memory panel while executing a program.
Select [Yes] to update the display (default).

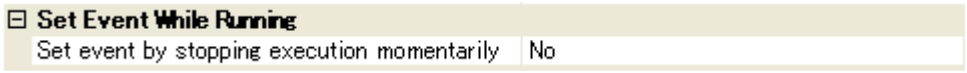
<3> [Display update interval[ms]]

This property is valid only when the [Update display during the execution] property is set to [Yes].
Specify the interval to automatically update the contents in the Watch panel/Memory panel display while executing a program.
Directly enter the Integer number between 100 and 65500 (rounding up the fractions less than 100ms) (default: [500]).

(c) [Set Event While Running]

You can configure the setting of events while executing a program.

Figure 2-24. [Set Event While Running] Category



<1> [Set event by stopping execution momentarily]

Specify whether to forcibly pause the execution for events that cannot be set while executing a program or operating the tracer/timer.
For details on the event types that are affected by this property, see "(2) Event types that can be set and deleted during execution".
Select [Yes] to set events above while execution (default: [No]).

(d) [Break]

You can configure the break function.
See "2.9 Stop Programs (Break)" for details on the break function and this category configuration.

(e) [Trace]

You can configure the trace function.

See "2.12 Collect Execution History of Programs" for details on the trace function and this category configuration.

(f) [Mask for Input Signal]

You can configure the input signal masking.

Figure 2-25. [Mask for Input Signal] Category [E1]

<1> [Mask WAIT signal]

Select whether to mask the WAIT signal.

Select [Yes] so that the WAIT pin signal is not input to E1 (default: [No]).

<2> [Mask RESET signal]

Select whether to mask the RESET signal.

Select [Yes] so that the RESET pin signal is not input to E1 (default: [No]).

<3> [Select the RESET signal to mask]

This property appears only when the [Mask RESET signal] property is set to [Yes].

The RESET signal to be masked is displayed.

You cannot change the value of this property.

(3) [Download File Settings] tab

You can configure downloading to the debug tool.

See "2.5.1 Execute downloading" for details on each category configuration.

(4) [Flash Options Settings] tab

You can configure options for the flash memory incorporated in the microcontroller.

Note that this tab appears only when the selected microcontroller supports the flash options.

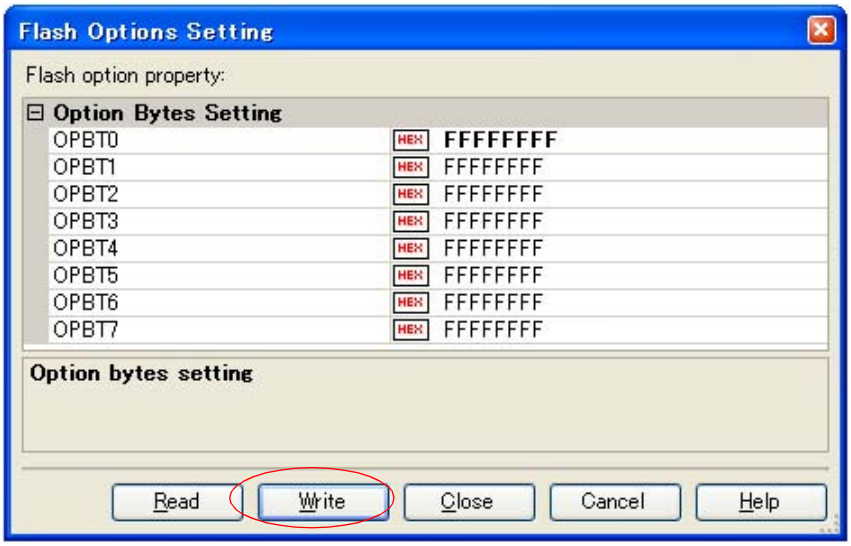
To configure options, specify the corresponding items via the Flash Options Setting dialog box [Full-spec emulator][E1][E20], that is opened by clicking the [...] button appears at the right of the field by selecting the [Flash options] property in the [Flash Options] category on this tab (the [...] button appears only while connected to the debug tool).

Click the [Write] button on this dialog box after specifying each item.

See the Flash Options Setting dialog box [Full-spec emulator][E1][E20] for details on the configuration.

Figure 2-26. Opening Flash Options Setting Dialog Box

Figure 2-27. Flash Options Settings (Flash Options Setting Dialog Box)



(5) [Hook Transaction Settings] tab

You can configure hook transaction for the debug tool.

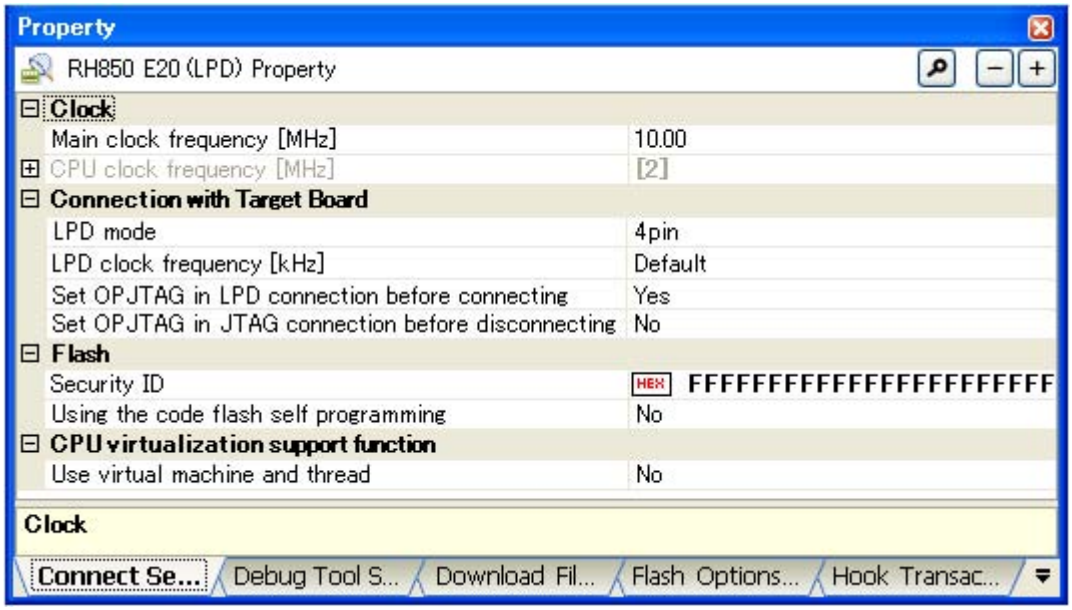
See "2.17 Use Hook Function" for details on each category configuration and the function of the hook transaction.

2.3.4 [E20]

Configure the operating environment on the Property panel below when using E20.

Caution Only LPD communications are supported for communication with the target board.

Figure 2-28. Example of Property Panel [E20]



Follow the steps below by selecting the corresponding tab on the Property panel.

(1) [Connect Settings] tab
(2) [Debug Tool Settings] tab
(3) [Download File Settings] tab
(4) [Flash Options Settings] tab
(5) [Hook Transaction Settings] tab

(1) [Connect Settings] tab

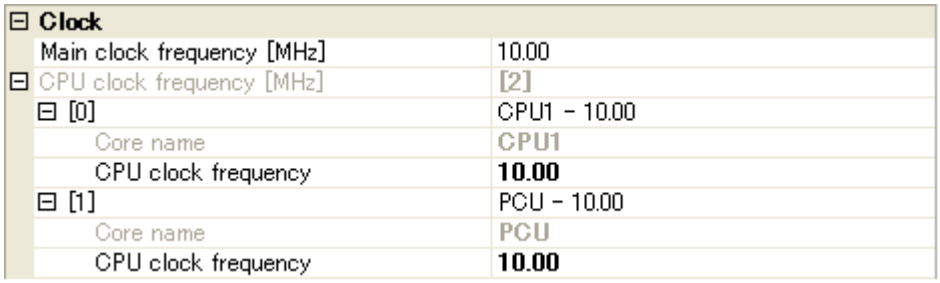
You configure the connection with the debug tool for each one of the following categories.

(a) [Clock]
(b) [Connection with Target Board]
(c) [Flash]
(d) [CPU virtualization support function]

(a) [Clock]

You can configure the clock.

Figure 2-29. [Clock] Category [E20]



<1> [Main clock frequency [MHz]]

Specify the main clock frequency (before multiplication).

You can specify the frequency from the drop-down list or by directly entering a frequency value between 0.001 and 999.999 (unit: MHz) (default: [10.00]).

<2> [CPU clock frequency [MHz]]

Specify the CPU clock frequency for each core.

The names of cores incorporated in the selected microcontroller are displayed as subproperties of this property. Specify the CPU clock frequency of each core by directly entering a frequency value between 0.001 and 999.999 (unit: MHz).

The number of subproperties displayed here differs with the selected microcontroller.

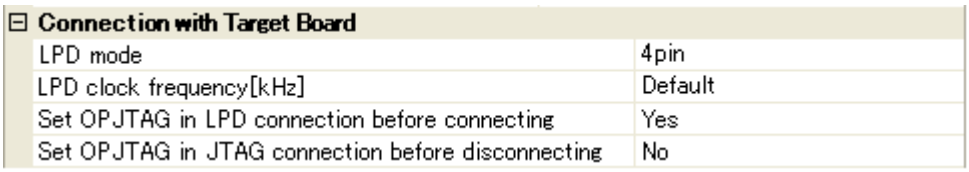
Remark The CPU clock frequency is used to convert the time stamp information for a trace to an actual time.

(b) [Connection with Target Board]

You can configure the connection between E20 and the target board.

Caution Properties in this category cannot be changed when CubeSuite+ is connected to E20.

Figure 2-30. [Connection with Target Board] Category [E20]



<1> [LPD mode]

Select LPD communication mode to use.

The selectable pin values differ depending on the selected microcontroller.

Note, however, that this property value cannot be changed when only one communication mode is available.

<2> [Baud rate[Kbps]]

This property appears only when the [LPD mode] property is set to [1pin].

Specify the baud rate for LPD communication (default: [500]).

<3> [LPD clock frequency[kHz]]

This property appears only when the [LPD mode] property is set to [4pin].

Specify the clock frequency for the LPD communication (default: [Default]).
When [Default] is selected, the default value specific to the microcontroller is used in connection to the target board.

<4> [Set OPJTAG in LPD connection before connecting]

Select whether to start up the microcontroller in serial programming mode upon connection to the debug tool and change the option byte settings to select LPD connection.
When [Yes] is selected, the debug tool starts up the microcontroller in serial programming mode upon its connection to CubeSuite+. The debug tool then checks the OPJTAG byte and, if LPD is not selected, changes the setting to select LPD. After that, the microcontroller enters debugging mode (default).
When [No] is selected, the debug tool starts up the microcontroller in debugging mode upon its connection to CubeSuite+. The debug tool then checks OPJTAG and, if LPD is not selected, shows a message dialog box.

<5> [Set OPJTAG in JTAG connection before disconnecting]

This property can be changed only when the [Set OPJTAG in LPD connection before connecting] property is set to [Yes].
Select whether to change the option byte settings to select JTAG connection before disconnection of the debug tool.
If you wish to change the option byte settings to select JTAG connection before disconnecting the debug tool, select [Yes].
When [No] (the default setting) is selected, the option byte settings are not changed before the debug tool is disconnected. In this case, LPD mode is applicable as the pin mode.

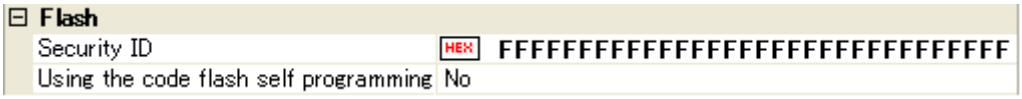
Remark On connection to E20, CubeSuite+ changes the option byte settings if LPD is not selected. For this reason, connecting and disconnecting E20 may change the value of the option bytes.

(c) [Flash]

You can configure the flash memory rewriting.

Caution Properties in this category cannot be changed when CubeSuite+ is connected to E20.

Figure 2-31. [Flash] Category [E20]



<1> [Security ID]

This property appears only when the selected microcontroller supports the ROM security function for flash memory.
Specify the key code for ID authentication when reading the code from the internal ROM or internal flash memory.
Directly enter 32 digits hexadecimal number (16 bytes) (default: [FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF]).

<2> [Using the code flash self programming]

Select whether to rewrite the code flash by using the flash self library of the flash self programming function.
Select [Yes] to rewrite the code flash (default: [No]).

Note that if [Yes] is selected in this property, the code flash will not be cashed.

(d) [CPU virtualization support function]

The property in this category is always disabled.

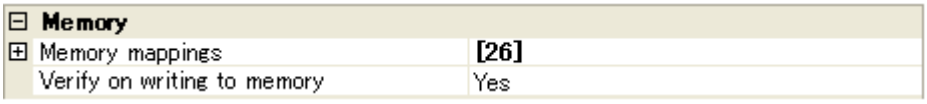
(2) [Debug Tool Settings] tab

You configure the basic settings of the debug tool for each one of the following categories.

(a) [Memory]

You can configure the memory.

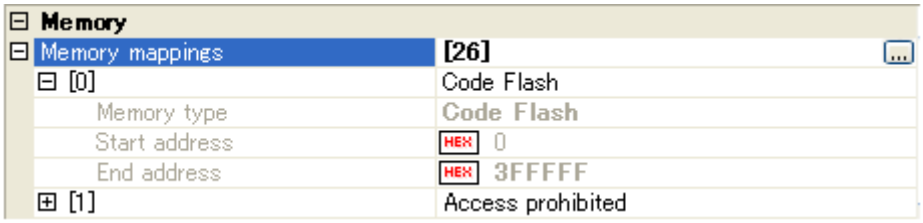
Figure 2-32. [Memory] Category [E20]



<1> [Memory mappings]

Current memory mapping status is displayed for each type of memory area by expanding this property.

Figure 2-33. Opening Memory Mapping Dialog Box



Caution The memory mapping cannot be added/deleted.

Remark When the selected microcontroller supports multi-core, this property displays the memory mapping status regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

<2> [Verify on writing to memory]

Select whether to perform a verify check when writing to the memory.

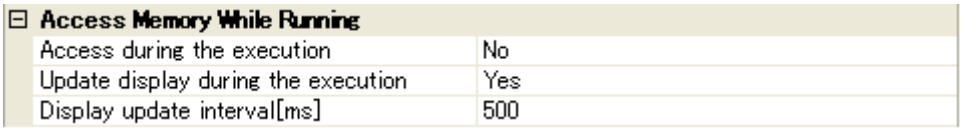
Select [Yes] to perform verification after download or when values are changed in the Watch panel/ Memory panel (default).

(b) [Access Memory While Running]

You can configure the memory access while executing a program (the real-time display update function).

See "(4) Display/modify the memory contents during program execution" for details on the real-time display update function.

Figure 2-34. [Access Memory While Running] Category [E20]



<1> [Access during the execution]

Select whether to allow access to the internal RAM area during execution of a program.
Select [Yes] to allow access (default: [No]).

<2> [Update display during the execution]

Select whether to automatically update the display in the Watch panel/Memory panel while executing a program.
Select [Yes] to update the display (default).

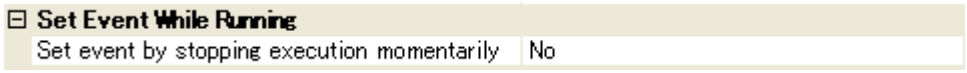
<3> [Display update interval[ms]]

This property is valid only when the [Update display during the execution] property is set to [Yes].
Specify the interval to automatically update the contents in the Watch panel/Memory panel display while executing a program.
Directly enter the Integer number between 100 and 65500 (rounding up the fractions less than 100ms) (default: [500]).

(c) [Set Event While Running]

You can configure the setting of events while executing a program.

Figure 2-35. [Set Event While Running] Category



<1> [Set event by stopping execution momentarily]

Specify whether to forcibly pause the execution for events that cannot be set while executing a program or operating the tracer/timer.
For details on the event types that are affected by this property, see "(2) Event types that can be set and deleted during execution".
Select [Yes] to set events above while execution (default: [No]).

(d) [Break]

You can configure the break function.
See "2.9 Stop Programs (Break)" for details on the break function and this category configuration.

(e) [Trace]

You can configure the trace function.
See "2.12 Collect Execution History of Programs" for details on the trace function and this category configuration.

(f) [Mask for Input Signal]

You can configure the input signal masking.

Figure 2-36. [Mask for Input Signal] Category [E20]

<1> [Mask WAIT signal]

Select whether to mask the WAIT signal.

Select [Yes] so that the WAIT pin signal is not input to E1 (default: [No]).

<2> [Mask RESET signal]

Select whether to mask the RESET signal.

Select [Yes] so that the RESET pin signal is not input to E1 (default: [No]).

<3> [Select the RESET signal to mask]

This property appears only when the [Mask RESET signal] property is set to [Yes].

The RESET signal to be masked is displayed.

You cannot change the value of this property.

(3) [Download File Settings] tab

You can configure downloading to the debug tool.

See "2.5.1 Execute downloading" for details on each category configuration.

(4) [Flash Options Settings] tab

You can configure options for the flash memory incorporated in the microcontroller.

Note that this tab appears only when the selected microcontroller supports the flash options.

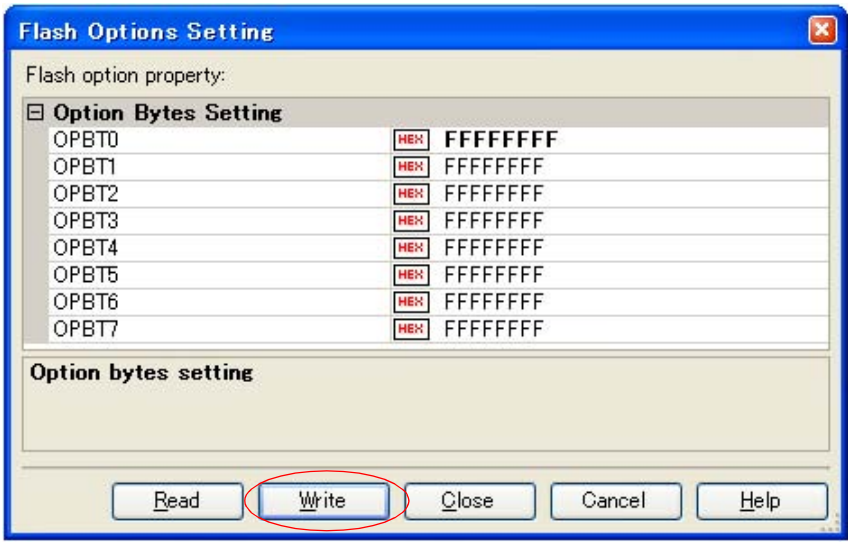
To configure options, specify the corresponding items via the Flash Options Setting dialog box [Full-spec emulator][E1][E20], that is opened by clicking the [...] button appears at the right of the field by selecting the [Flash options] property in the [Flash Options] category on this tab (the [...] button appears only while connected to the debug tool).

Click the [Write] button on this dialog box after specifying each item.

See the Flash Options Setting dialog box [Full-spec emulator][E1][E20] for details on the configuration.

Figure 2-37. Opening Flash Options Setting Dialog Box

Figure 2-38. Flash Options Settings (Flash Options Setting Dialog Box)



(5) [Hook Transaction Settings] tab

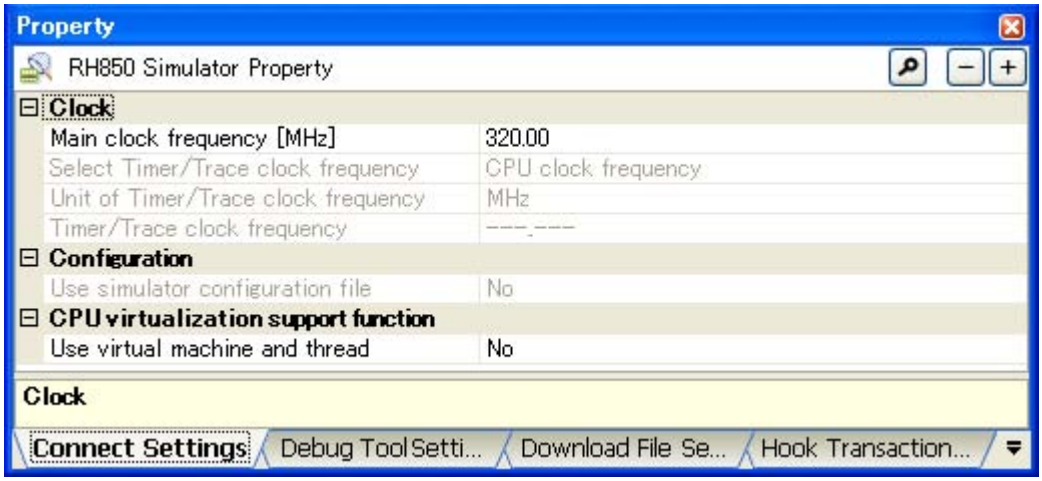
You can configure hook transaction for the debug tool.

See "2.17 Use Hook Function" for details on each category configuration and the function of the hook transaction.

2.3.5 [Simulator]

Configure the operating environment on the Property panel below when using Simulator.

Figure 2-39. Example of Property Panel [Simulator]



Follow the steps below by selecting the corresponding tab on the Property panel.

(1) [Connect Settings] tab
(2) [Debug Tool Settings] tab
(3) [Download File Settings] tab
(4) [Hook Transaction Settings] tab

(1) [Connect Settings] tab

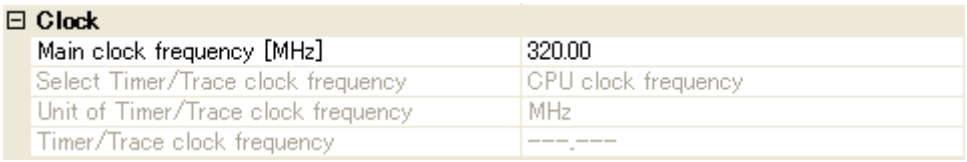
You configure the connection with the debug tool for each one of the following categories.

(a) [Clock]
(b) [Configuration]
(c) [CPU virtualization support function]

(a) [Clock]

You can configure the clock.

Figure 2-40. [Clock] Category [Simulator]



<1> [Main clock frequency [MHz]]

Specify the main clock frequency.
You can specify the frequency from the drop-down list or by directly entering a frequency value between 0.001 and 999.999 (unit: MHz) (default: [320.00]).

Caution When the instruction simulator for RH850 is used, the CPU clock frequency will always be the same as the value of the main clock frequency set in this property.

<2> [Select Timer/Trace clock frequency]

The clock frequency for using timer/trace function is displayed.
You cannot change the value of this property.

<3> [Unit of Timer/Trace clock frequency]

The unit of the clock frequency for using timer/trace function is displayed.
You cannot change the value of this property.

<4> [Timer/Trace clock frequency]

The value of the clock frequency for using timer/trace function is displayed.
Note, however, that "---_---" is displayed while disconnected from the debug tool.
You cannot change the value of this property.

(b) [Configuration]

The property in this category is always disabled.

(c) [CPU virtualization support function]

The property in this category is always disabled.

(2) [Debug Tool Settings] tab

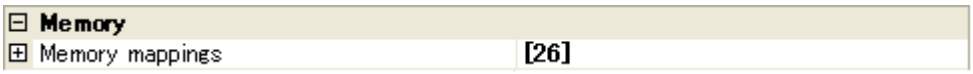
You configure the basic settings of the debug tool for each one of the following categories.

- (a) [Memory]
- (b) [Access Memory While Running]
- (c) [Trace]
- (d) [Timer]
- (e) [Coverage]
- (f) [Simulator GUI]

(a) [Memory]

You can configure the memory.

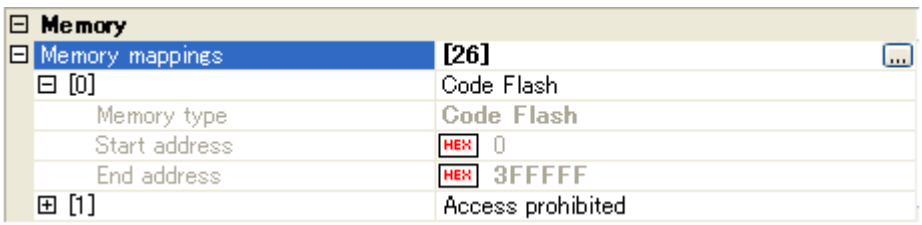
Figure 2-41. [Memory] Category [Simulator]



<1> [Memory mappings]

Current memory mapping status is displayed for each type of memory area by expanding this property.

Figure 2-42. Detailed Display of Memory Mapping



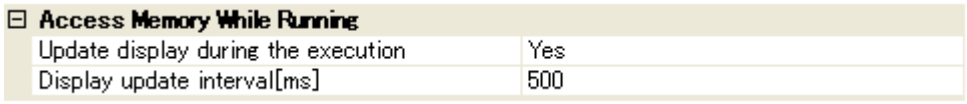
Caution The memory mapping cannot be added/deleted.

Remark When the selected microcontroller supports multi-core, this property displays the memory mapping status regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

(b) [Access Memory While Running]

You can configure the memory access while executing a program (the real-time display update function). See "(4) Display/modify the memory contents during program execution" for details on the real-time display update function.

Figure 2-43. [Access Memory While Running] Category [Simulator]



<1> [Update display during the execution]

Select whether to automatically update the display in the Watch panel/Memory panel during a program execution.

Select [Yes] to update the display (default).

<2> [Display update interval[ms]]

This property is valid only when the [Update display during the execution] property is set to [Yes].

Specify the interval to automatically update the contents in the Watch panel/Memory panel display while executing a program.

Directly enter the Integer number between 100 and 65500 (rounding up the fractions less than 100ms) (default: [500]).

(c) [Trace]

You can configure the trace function.

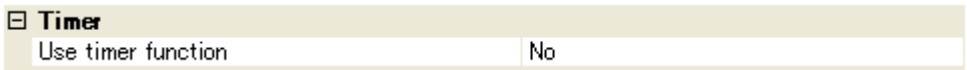
See "2.12 Collect Execution History of Programs" for details on the trace function and this category configuration.

(d) [Timer]

You can configure the timer function.

See "2.13 Measure Execution Time of Programs" for details on the timer function.

Figure 2-44. [Timer] Category



<1> [Use timer function]

Select whether to use the timer function.

Select [Yes] to use the timer function (default: [No]).

(e) [Coverage]

You can configure the coverage function.

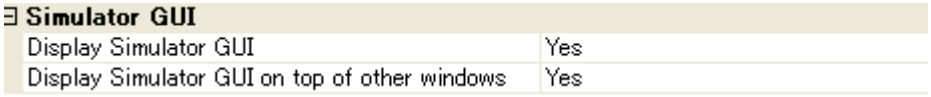
See "2.14 Measure Coverage [Simulator]" for details on the coverage function and this category configuration.

(f) [Simulator GUI]

You can configure the Simulator GUI function.

Caution If a microcontroller whose Simulator does not support peripheral function simulations (instruction simulation version) is selected, all properties in this category become invalid.

Figure 2-45. [Simulator GUI] Category



<1> [Display Simulator GUI]

Select whether to display the Simulator GUI window.

Select [Yes] to use the Simulator GUI function (default).

When you do not need to use the Simulator GUI, select [No] to close the Simulator GUI window.

<2> [Display Simulator GUI on top of other windows]

This property appears only when the [Display Simulator GUI] property is set to [Yes].

Select whether to display the Simulator GUI window in the forefront when program execution starts.

Select [Yes] to display it in the forefront (default).

(3) [Download File Settings] tab

You can configure downloading to the debug tool.

See "2.5.1 Execute downloading" for details on each category configuration.

(4) [Hook Transaction Settings] tab

You can configure hook transaction for the debug tool.

See "2.17 Use Hook Function" for details on each category configuration and the function of the hook transaction.

2.4 Connect to/Disconnect from the Debug Tool

This section describes how to connect to/disconnect from the debug tool.

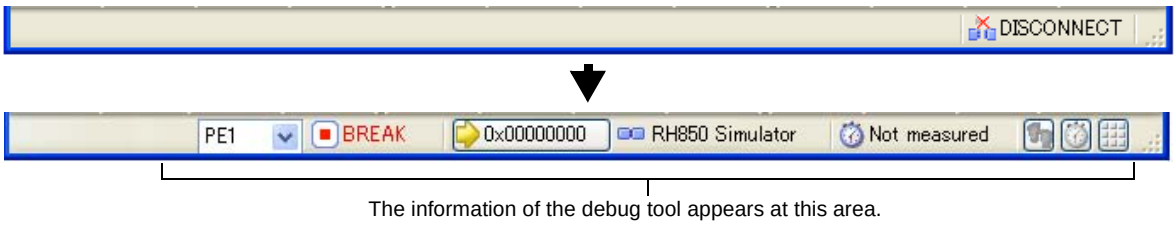
2.4.1 Connect to the debug tool

By selecting [Connect to Debug Tool] from the [Debug] menu, CubeSuite+ starts communicating with the debug tool selected in the active project.

After succeeding in the connection to the debug tool, the Statusbar of the Main window changes as follows:

For details on each item displayed on the Statusbar, see the section of the "Main window".

Figure 2-46. Statusbar Indicating Successful Connection to Debug Tool



Caution If the version of compiler being used is not supported by CubeSuite+, [Connect to Debug Tool] will be disabled.

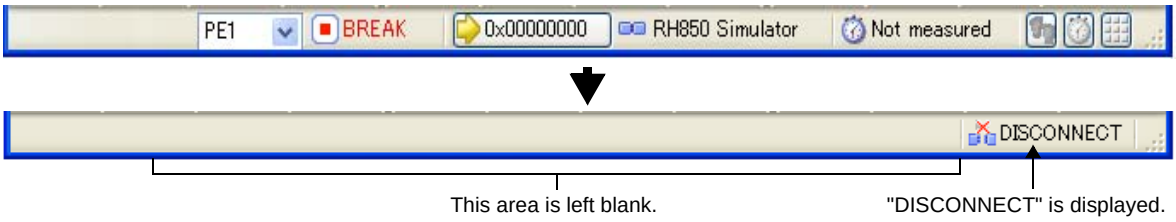
Remark When the button on the Debug toolbar is clicked, the specified file is downloaded automatically after connecting to the debug tool (see "2.5.1 Execute downloading").
When the button on this toolbar is clicked, the project is built automatically, and then the built file is downloaded after connecting to the debug tool.

2.4.2 Disconnect from the debug tool

By clicking the button on the Debug toolbar, CubeSuite+ cuts off the communication with the connected debug tool.

After disconnecting from the debug tool, the Statusbar of the Main window changes as follows:

Figure 2-47. Statusbar Indicating Disconnection from Debug Tool



Caution The debug tool cannot be disconnected from CubeSuite+ while the program is running.

Remark Disconnecting the debug tool will close all the panels and dialog boxes that can be displayed only during the connection.

2.5 Download/Upload Programs

This section describes how to download programs (such as load module files) to debug to CubeSuite+ and how to upload the memory contents being debugged from CubeSuite+ to files.

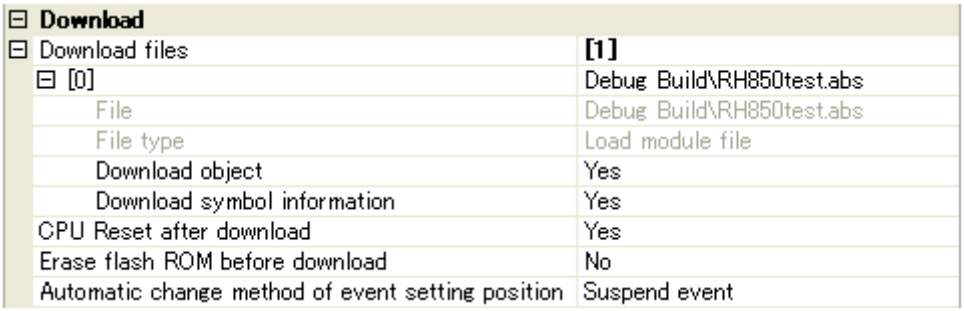
2.5.1 Execute downloading

Download the load module file to be debugged to the debug tool that is currently connected.

Follow the steps below on the [Download File Settings] tab in the Property panel for the downloading, and then execute the downloading.

(1) [Download] category setting

Figure 2-48. [Download] Category



(a) [Download files]

The names of files to be downloaded and download conditions are displayed (the number enclosed with "[ ]" indicates the number of files to be download).

Files that are specified as build target files in the main project or subprojects will automatically be selected as the files to be downloaded[Note].

However, you can manually change the download files and the condition. In this case, see "2.5.2 Advanced downloading".

Note To download the load module files created by an external build tool (e.g., compilers and assemblers other than the build tools supplied with CubeSuite+), a debug-dedicated project needs to be created. If you use a debug-dedicated project as the subject to debug, add your a download file to Download files node on project tree. The file to be downloaded will be reflected in this property.
See "CubeSuite+ Integrated Development Environment User's Manual: Start" for details on the using an external build tool and a debug-dedicated project.

(b) [CPU Reset after download]

Specify whether to reset the CPU after downloading.

Select [Yes] to reset the CPU (default).

(c) [Erase flash ROM before download] [Full-spec emulator][E1][E20]

Specify whether to erase the flash ROM before downloading.

Select [Yes] to erase the flash ROM ([No] is selected by default).

(d) [Automatic change method of event setting position]

If the file is downloaded again during debugging then the location (address) set for the currently configured event may change to midway in the instruction.

Specify with this property how to handle the target event in this circumstance.

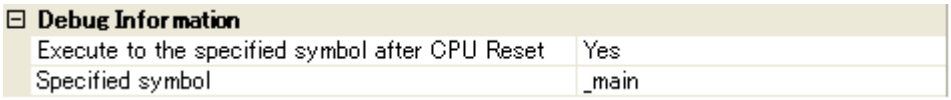
Select one of the options from the following drop-down list.

Move to the head of instruction	Resets the subject event at the beginning address of the instruction.
Suspend event	Leaves the subject event pending (default).

Note, however, that this property setting only applies to the location setting of events without debugging information. The location setting of events with debug information is always moved to the beginning of the source text line.

(2) [Debug Information] category setting

Figure 2-49. [Debug information] Category



(a) [Execute to the specified symbol after CPU Reset]

Specify whether to execute the program to the specified symbol position after CPU reset or downloading (for only when the [CPU Reset after download] property is set to [Yes]).
Select [Yes] to execute the program to the specified symbol position after CPU reset (default).

Remark When the [CPU Reset after download] property is set to [Yes], the operation after downloading is as follows:
- If [Yes] is selected for this property, the Editor panel will open automatically with displaying source text of the position specified with the [Specified symbol] property after downloading.
- If [No] is selected for this property, the Editor panel will open with displaying source text of the reset address (when if the source text has not been allocated to the reset address, the contents of the reset address is displayed in the Disassemble panel).

(b) [Specified symbol]

This property appears only when the [Execute to the specified symbol after CPU Reset] property is set to [Yes].
Specify the position at which the program is stop after CPU reset.
Directly enter an address expression between 0 and "*last address in address space*" ([_main] is specified by default).
Note, however, that the program will not be executed if the specified address expression cannot be converted into an address.

Remark Normally, specify the following.
For assembly source: Start label corresponding to main function
For C source: Symbol assigned to the start of the main function name

Caution By default, CPU reset automatically occurs after downloading the file, and then the program is executed to the specified symbol position. If this operation above is not needed, specify [No] with both of the [CPU Reset after download] and [Execute to the specified symbol after CPU Reset] property.

(3) Execute download

Click the  button on the Debug toolbar.

If this operation is performed while disconnecting from the debug tool, the application automatically connects to the debug tool, and then performs the download.

Remark When a program that has been modified during debugging is re-downloaded, you can easily build and download it by selecting [Build & Download] from the [Debug] menu on the Main window.

If the load module file is successfully downloaded, the Editor panel opens automatically, and the contents of the downloaded file's source text are displayed.

Remark You can automatically overwrite the value of I/O register/CPU register with the specified values before and after performing the download (see "2.17 Use Hook Function" for details).

2.5.2 Advanced downloading

You can change the download files and the condition to download.

With CubeSuite+, the following file types can be downloaded.

Table 2-1. Downloadable File Formats

Downloadable File	Extension	File Format
Load module file	.abs	Load module file format
	.out **[GHS]**Note	
Intel HEX file	.hex	Intel HEX file format
Motorola S-record file	.mot	Motorola S-record file format - (S0, S1, S9-16 bits) - (S0, S2, S8-24 bits) - (S0, S3, S7-32 bits)
Binary file	.bin	Binary file format

Note Notes on using GHS compiler (Green Hills Software, Inc., USA)

- Supported version

- MULTI (Ver.6.1.4), and GHS compiler (Ver.2012.5.5)

- Supported options

Debug option: -G, -dual_debug, -cpu=rh850/-cpu=v850e3

Optimization option: -Odebug, -O, -Ospeed, -Onone

Other than above: -prepare_dispose, -callt

- Non-supported options

Linker optimization option: -shorten_loads, -code_factor, -delete

(NOTE: Linker changes execution code. However, the changes are not reflected on the debug information.)

- Notes on debugging

- Add the created load module file to a debug-dedicated project.
- The followings are not supported:
  - C++
  - Programs with C99 own type or modifier
  - Programs with gnu c extensional specs

- **Step or Execution related functions**

  Executing return out functions from the following function may fail. And call history on the Call Stack panel is not shown incorrect.

  - Functions is called by callt
  - Interrupt Functions

- **Reference function of variables using expressions**

  - When long long type or double type variables are located to register, only lower 4 bit register name is shown in address column in the Watch panel. CubeSuite+ gets upper 4 bit value from next to lower 4 bit register.

    For example, if R4 is shown in the address column, then CubeSuite+ will get the upper 4 bit value from R5.
  - When structure type variables are located to a registers, correct value of structure members aren't shown on the Watch panel. See the value that is show in register of [address] area on the CPU Register panel.
  - Even using an expression with scope specify, it is impossible to refer defined static variables in functions. When a program counter exists in the function that the static variables are defined in, it is possible to refer it.

```
func(){
    static  sta = 100;
}
```

    In the above case, during debugging func(), it is possible to refer both "sta" and "func()#sta". During debugging functions except func(), it is possible to refer neither "sta" nor "func()#sta".
  - As stack frame is not generated at a start point of a function, passed address of a variable via the stack is not correct. Please refer value of a variable after stepping in the function.

- **Other than the above**

  - It is not possible to invalidate the Symbol name completion function (a specification of [Generate the information for input completion] item in the Download Files dialog box will be ignored).

You can change the download files or download conditions in the following Download Files dialog box.

The Download Files dialog box is opened by clicking the [...] button that appears at the right edge in the column of the [Download files] property when you select it in the [Download] category on the [Download File Settings] tab of the Property panel.

Figure 2-50. Opening Download Files Dialog Box

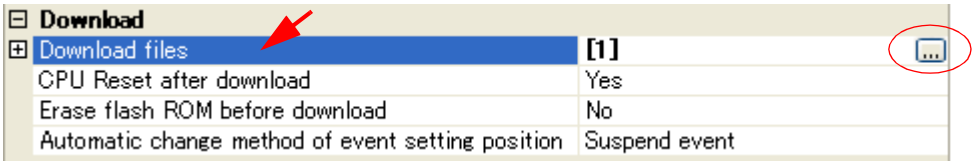
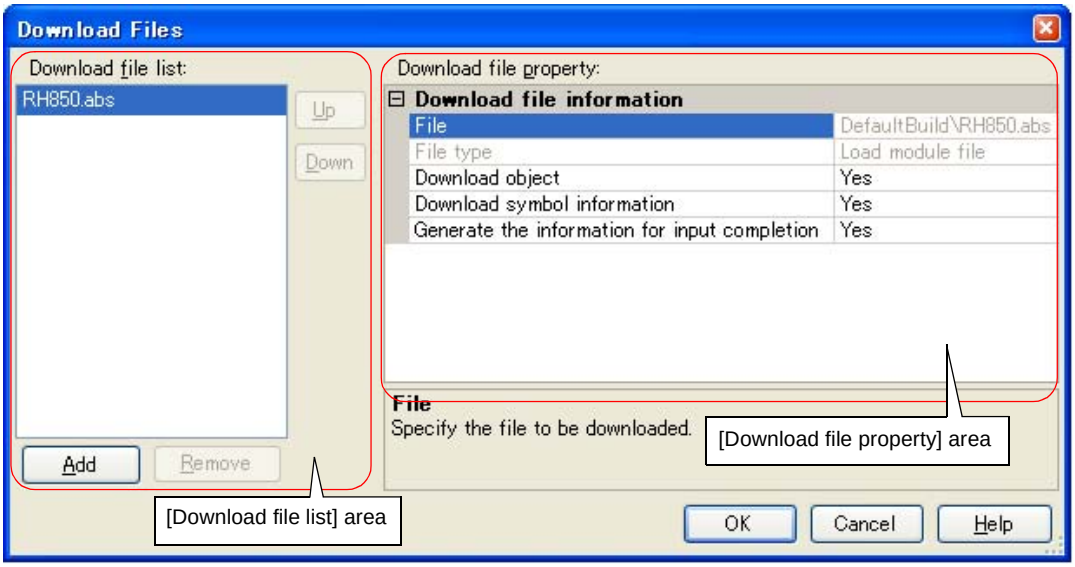


Figure 2-51. Advanced Downloading (Download Files Dialog Box)



This section describes how to configure on the Download Files dialog box above when the following cases.

(1) Change download conditions for load module files
(2) Add download files (*.hex/*mot/*.bin)
(3) Download multiple load module files
(4) Perform source level debugging with files other than the load module file format

(1) Change download conditions for load module files

Follow the steps below in the Download Files dialog box to change the download conditions (object information and symbol information) for load module files.

(a) Select a load module file

Select a load module file to download in the [Download file list] area.

(b) Change download conditions

Current download conditions for the selected load module file are displayed in the [Download file property] area.

Change each items displayed in the property.

Download object	Select whether to download the object information from the specified file.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Downloads object information.
		No	Does not download object information.
Download symbol information	Select whether to download the symbol information from the specified file[Note 1].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Downloads symbol information.
		No	Does not download symbol information.

Generate the information for input completion	Select whether to generate the information for the Symbol name completion function when downloading[Note 2].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Generates the information for the symbol name completion function. (i.e. uses the symbol name completion function.)
		No	Does not generate the information for the symbol name completion function. (i.e. does not use the symbol name completion function.)

Notes 1. If the symbol information have not been downloaded, the source level debugging cannot be performed.

2. When [Yes] is selected, the time taken for downloading and the memory usage on the host machine will increase. We recommend selecting [No] in this item if you do not intend to use the symbol name completion function.

(c) Click the [OK] button

Enable all the configuration in this dialog box and change the download conditions.

(2) Add download files (*.hex/*mot/*.bin)

Follow the steps below to add download files other than the load module file format (Intel HEX file (*.hex), Motorola S-record file (*.mot), or binary file (*.bin)) in the Download Files dialog box.

(a) Click the [Add] button

When the [Add] button is clicked, a blank list item "-" is displayed in the last line of the [Download file list] area.

(b) Property configuration of the download files to add

Configure the download conditions for the download file to add in the [Download file property] area.
Configure each item displayed with the following condition. When the configuration is completed, the file name specified in this property is displayed in the blank list of the [Download file list] area.

File	Specify the download file (Intel HEX file (*.hex), Motorola S-record file (*.mot), or binary file (*.bin)) to download (up to 259 characters).		
	Default	Blank	
	Modifying	Directly enter from the keyboard, or specify with the Select Download File dialog box opened by clicking the [...] button.	
	Available values	See "Table 2-1. Downloadable File Formats".	
File type	Select the type of the file to download. Here, select a item other than [Load module file].		
	Default	Load module file	
	Modifying	Select from the drop-down list.	
	Available values	Load module file	Specifies a load module file (*.abs).
		Hex file	Specifies an Intel HEX file (*.hex).
		S record file	Specifies a Motorola S-record file (*.mot).
		Binary data file	Specifies an binary file (*.bin).

Offset	Specify the offset from the address at which the file's download is to start. Note that this item appears only when [File type] is set to [Hex file] or [S record file].	
	Default	0
	Modifying	Directly enter from the keyboard.
	Available values	0x0 to 0xFFFFFFFFFF in hexadecimal number
Start address	Specify the address at which to start the file's download. Note that this item appears only when [File type] is set to [Binary file].	
	Default	0
	Modifying	Directly enter from the keyboard.
	Available values	0x0 to 0xFFFFFFFFFF in hexadecimal number

Remark The settings of whether to download the object information or symbol information can be made only when the type of the file to download is load module files.

(c) Check the order of download

The order of the download is the display order of the files displayed in the [Download file list] area.
If you want to change the order, use the [Up]/[Down] button.

(d) Click the [OK] button

Enable all the configuration in this dialog box and add a download file (the file name is displayed in the [Download] category on the [Download File Settings] tab of the Property panel).

(3) Download multiple load module files

Follow the steps below on the Download Files dialog box to download multiple load module files.

Caution When debugging a program consisting of multiple load module files, care should be taken to avoid overlapping of location addresses.

(a) Click the [Add] button

When the [Add] button is clicked, a blank list item "-" is displayed in the last line of the [Download file list] area.

(b) Property configuration of the download files to add

Configure the download conditions for the download file to add in the [Download file property] area.
Configure each item displayed with the following condition.
When the configuration is completed, the file name specified in this property is displayed in the blank list of the [Download file list] area.

File	Specify the name of the load module file to be added (up to 259 characters).	
	Default	Blank
	Modifying	Directly enter from the keyboard, or specify with the Select Download File dialog box opened by clicking the [...] button displayed at the right edge of this property when it is selected.
	Available values	See "Table 2-1. Downloadable File Formats".

File type	Specify the type of the file to download. Here, select [Load module file].		
	Default	Load module file	
Download object	Select whether to download the object information from the specified file.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Downloads object information.
		No	Does not download object information.
Download symbol information	Select whether to download the symbol information from the specified file[Note 1].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Downloads symbol information.
		No	Does not download symbol information.
Generate the information for input completion	Select whether to generate the information for the Symbol name completion function when downloading[Note 2].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Generates the information for the symbol name completion function. (i.e. uses the symbol name completion function.)
		No	Does not generate the information for the symbol name completion function. (i.e. does not use the symbol name completion function.)

Notes 1. If the symbol information have not been downloaded, the source level debugging cannot be performed.

2. When [Yes] is selected, the time taken for downloading and the memory usage on the host machine will increase. We recommend selecting [No] in this item if you do not intend to use the symbol name completion function.

Remark You can decrease the memory usage by selecting [No] for the [Download symbol information] item if the symbol information is not required for the module (in this case, however, the source level debugging can not be performed for the file).

(c) Check the order of download

The order of the download is the display order of the files displayed in the [Download file list] area.
If you want to change the order, use the [Up]/[Down] button.

(d) Click the [OK] button

Enable all the configuration in this dialog and add the specified load module file (the specified file name is displayed in the [Download] category on the [Download File Settings] tab of the Property panel).

(4) Perform source level debugging with files other than the load module file format

Even when an Intel HEX file (*.hex), Motorola S-record file (*.mot), or binary file (*.bin) is specified to be the subject file to download, it is possible to do source level debugging by downloading symbol information for the load module file from which the subject file was created, along with the subject file that you download.

To do so, follow the steps below on the Download Files dialog box.

(a) Click the [Add] button

When the [Add] button is clicked, a blank list item "-" is displayed in the last line of the [Download file list] area.

(b) Property configuration of the load module file to add

Configure each item displayed with the following condition in the [Download file property] area.

File	Specify a load module file from which the Intel HEX file (*.hex), Motorola S-record file (*.mot), or binary file (*.bin) that you want to download was created. Directly enter from the keyboard, or specify with the Select Download File dialog box opened by clicking the [...] button that appears at right by selecting this property.		
File type	Select [Load module file] (default).		
Download object	Select [No].		
Download symbol information	Select [Yes] (default).		
Generate the information for input completion	Select whether to generate the information for the Symbol name completion function when downloading[Note].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Generates the information for the symbol name completion function. (i.e. uses the symbol name completion function.)
		No	Does not generate the information for the symbol name completion function. (i.e. does not use the symbol name completion function.)

Note When [Yes] is selected, the time taken for downloading and the memory usage on the host machine will increase. We recommend selecting [No] in this item if you do not intend to use the symbol name completion function.

(c) Click the [OK] button

Enable all the configuration in this dialog box and add the specified load module file (Only the symbol information included in the load module file will be downloaded).

2.5.3 Execute uploading

The contents of the memory of the debug tool currently connected can be saved (uploaded) in an arbitrary file.
You can upload the data in the Data Save dialog box that is opened by selecting the [Debug] menu >> [Upload...].
In this dialog box, follow the steps below.

Figure 2-52. Execute Uploading (Data Save Dialog Box)

(1) Specify [File Name]

Specify the name of the file to save.

You can either type a filename directly into the text box (up to 259 characters), or select one from the input history via the drop-down list (up to 10 items). You can also specify the file by clicking the [...] button, and selecting a file via the Select Data Save File dialog box.

(2) Specify [File Type]

Select the format in which to save the file from the following drop-down list.

The following file formats can be selected.

Table 2-2. Uploadable File Formats

List Display	File Format
Intel Hex format (*.hex)	Intel HEX file format (The expanded linear address record is always used)
Motorola S-record (*.mot)	Motorola S-record file format
Binary data (*.bin)	Binary file format

(3) Specify [Save Range Address/Symbol]

Specify the range of addresses to save via "start address" and "end addresses".

Directly enter hexadecimal number/address expression in each text box or select from the input history displayed in the drop-down list (up to 10 items).

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").

(4) Click the [Save] button

Save the contents of the memory in the specified file in specified format as upload data.

2.6 Display/Change Programs

This section describes how to display and change programs when a load module file with the debug information is downloaded to a debug tool.

Downloaded programs can be displayed in the following panels.

- Editor panel

  The source file is displayed and can be edited.

  Furthermore, the source level debugging/instruction level debugging (see "2.8.3 Execute programs in steps") and the display of the code coverage measurement result **[Simulator]** (see "2.14 Measure Coverage [Simulator]") can be performed in this panel.

- Disassemble panel

  The result of disassembling the downloaded program (the memory contents) is displayed and can be edited (line assemble).

  Furthermore, the instruction level debugging (see "2.8.3 Execute programs in steps") and the display of the code coverage measurement result **[Simulator]** (see "2.14 Measure Coverage [Simulator]") can be performed in this panel. In this panel, the disassemble results can be displayed with the corresponding source text (default).

Remark It is normally necessary to download a load module file with debugging information in order to perform the source level debugging, but it is also possible to do so by downloading an Intel HEX file (*.hex), Motorola S-record file (*.mot), or binary file (*.bin) (see "(4) Perform source level debugging with files other than the load module file format").

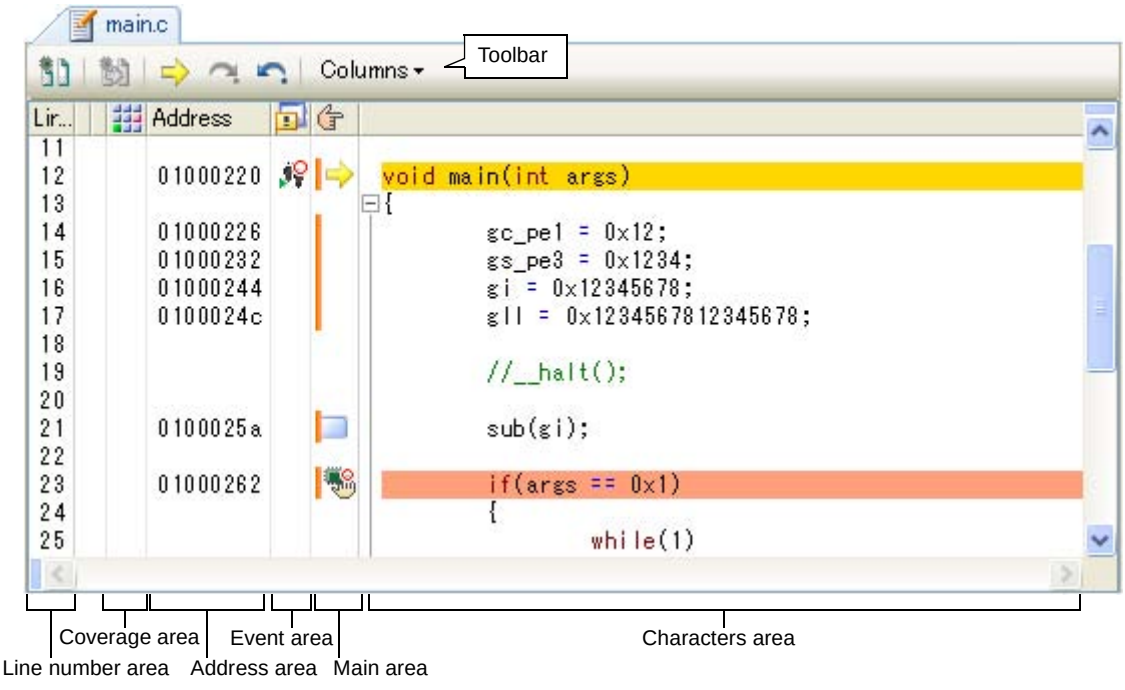
2.6.1 Display source files

The source file is displayed in the Editor panel below. The Editor panel automatically opens with displaying source text of the specified position (see "2.5.1 Execute downloading") when a load module file is successfully downloaded.

If you want to open the Editor panel manually, double-click on the source file in the Project Tree panel.

For details on the contents and function in each area, see the section for the Editor panel.

Figure 2-53. Display Source File (Editor Panel)



Remark You can open a file with a specific encoding selected in the Encoding dialog box that is opened by selecting the [File] menu >> [Open with encoding...].

This section describes the following.

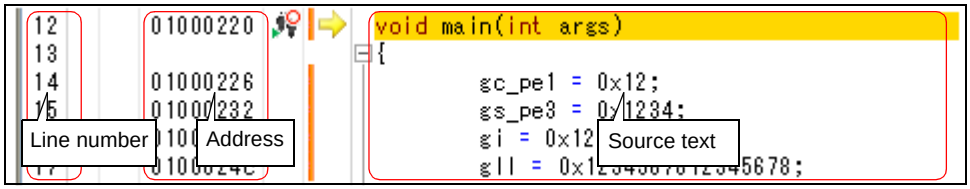
(1) Change display mode

You can change the display mode of the Editor panel by clicking the button (toggle) on the toolbar.

- Normal display mode

In this display mode (default), the line number, address and source text, etc. are displayed.

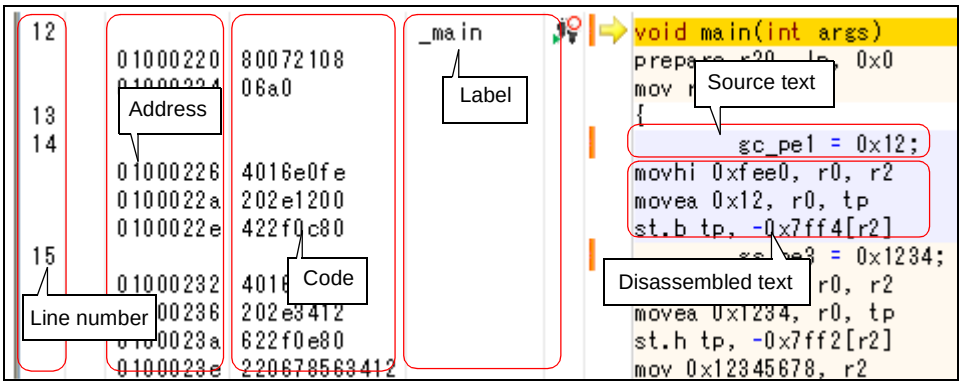
Figure 2-54. Normal Display Mode (Editor Panel)



- Mixed display mode

In this display mode, the code data, label and disassembled text are displayed combined with the source text.

Figure 2-55. Mixed Display Mode (Editor Panel)



Cautions 1. The mixed display mode is enabled only when connected to the debug tool and the downloaded source file is opened in this panel.

2. **In the mixed display mode, the source text and the corresponding code information that were acquired from the downloaded load module file are displayed. Therefore, to display the source text that has been modified in the mixed display mode, you need to run a rebuild and download it.**
3. **In the mixed display mode, the source files cannot be edited. In addition, [Redo]/[Cut]/[Paste]/[Delete]/[Select All]/[Replace...]/[Bookmark]/[Outlining]/[Advanced] from the [Edit] menu are disabled.**

Remark You can save the contents of the current mixed display to a file in the text or CSV format by selecting [Save Source Mixed Data As...] from the [File] menu (the contents of hidden columns cannot be saved).

(2) Set the columns to display

The columns or marks displayed on the Editor panel can be set by selecting the toolbar items shown below. Note that this setting applies to all of the Editor panels.

Columns	The following items are displayed to show or hide the columns or marks on all of the Editor panels. Remove the check to hide the items (all the items are checked by default).
Line Number	Shows the line number, in the line number area.
Selection	Shows the mark that indicates the line modification status, in the line number area.
Out of date module indicator	Shows the mark that indicates the update status of the downloaded load module file, in the line number area. Note that this item is enabled only when connected to the debug tool.
Coverage	Shows the coverage area. Note that this item is enabled only when connected to the debug tool.
Address	Shows the address area. Note that this item is enabled only when connected to the debug tool.
Op Code	Shows the code area. Note that this item is enabled only when connected to the debug tool and the mixed display mode is selected.
Label	Shows the label area. Note that this item is enabled only when connected to the debug tool and the mixed display mode is selected.
Event	Shows the event area. Note that this item is enabled only when connected to the debug tool.
Main	Shows the main area.
Column Header	Shows the column header.

(3) Display multiple source files in a single panel

If the current PC moves between multiple source files when debugging (e.g. when performing step execution), each of the source files will be opened in a separate editor panel. If this is the case, the recycle mode lets you display multiple source files in a single Editor panel.

Select the [Use window recycling] check box on the [General - Text Editor] category in the Option dialog box to enable this feature.

Figure 2-56. Normal Operation

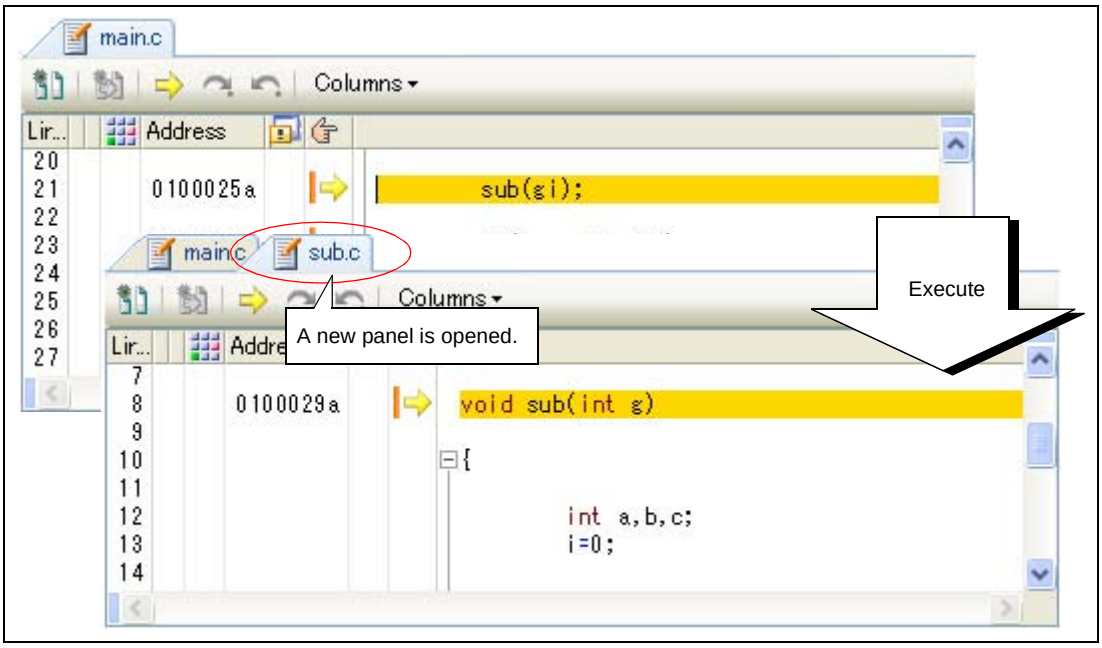
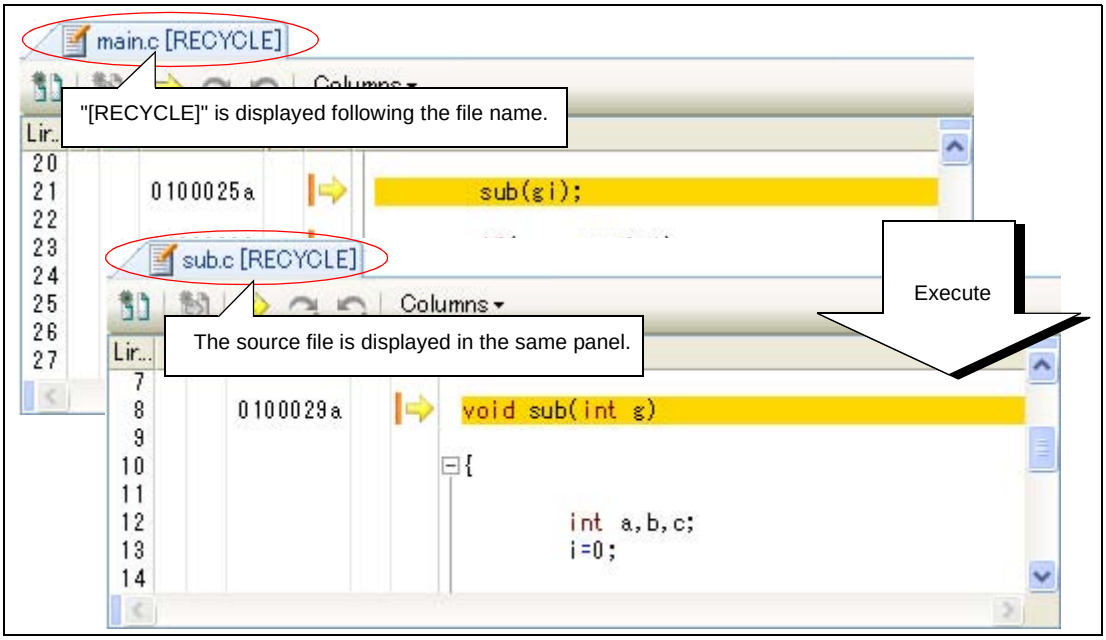


Figure 2-57. Recycle Mode Operation



Cautions 1. The recycle mode is enabled only when connected to the debug tool and the downloaded source file is opened in this panel.

2. When the current PC value in program execution corresponds to a line in the Editor panel while editing is being conducted in the recycle mode, that Editor panel is released from the recycle mode, and a new Editor panel is opened in the recycle mode.

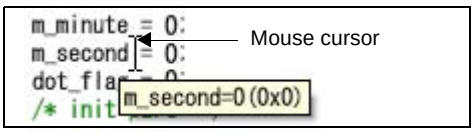
Remark If the Editor panel that displays the corresponding source file is already opened, then the source file is not opened in the panel of the recycle mode, but the Editor panel being opened is displayed.

(4) Display variables

When hovering the mouse cursor over a variable in the source text, a pop-up that shows the name and value of the variable is displayed ("<*variable name*>=<*variable value*>").

The display format of the variable value is same as "Table A-7. Display Format of Watch-Expressions (Default)" depending on the type of the variable.

Figure 2-58. Pop-up Display of Variables (Editor Panel)



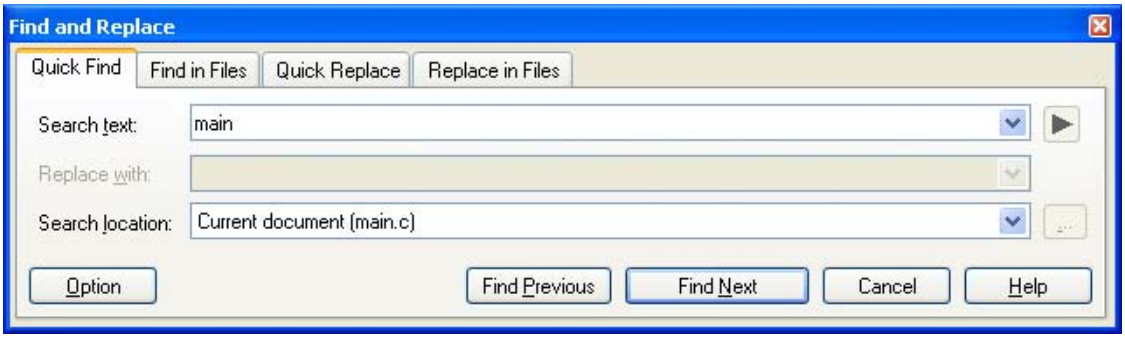
Caution A pop-up display is enabled only when connected to the debug tool and the downloaded source file is opened in this panel.

(5) Search characters

Character searching in the source text is taken place in the Find and Replace dialog box opens with selecting the button on the toolbar.

In this dialog box, follow the steps below.

Figure 2-59. Character Search in Source Text (Find and Replace Dialog Box)



(a) Specify [Search text]

Enter characters to search.

A word (variable/function) at the caret position in the Editor panel is specified by default.

If you want to change it, directly enter the characters into the text box (up to 1024 characters) or select from the input history in the drop-down list (up to 10 items).

(b) Specify [Search location]

Select [Current document (*file name*)] from the drop-down list.

(c) Click the [Find Previous]/[Find Next] button

When the [Find Previous] button is clicked, search will start in the order from the large address number to small and the search results are displayed selected in the Editor panel.

When the [Find Next] button is clicked, search will start in the order from the small address number to large and the search results are displayed selected in the Editor panel.

Remarks 1. Click the [Option] button to specify to use wild card, case sensitivity, word by word search, and so on.

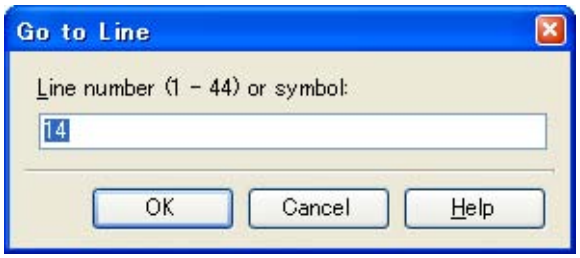
2. In the Find and Replace dialog box, various search/replace operation can be performed by selecting [Find in Files], [Quick Replace] or [Replace in Files] tab.

(6) Move to the specified line

You can move to the specified line in the source text in the Go to Line dialog box which opens when selecting [Go to...] from the context menu.

In this dialog box, follow the steps below.

Figure 2-60. Move to Specified Line in Source Text (Go to Line Dialog Box)



(a) Specify [Line number (*valid line range*) or symbol] area

Directly enter the line number (decimal number), symbol name[Note 1] or address[Note 2] to which you want to move the caret.

"(*valid line range*)" shows the range of valid lines in the current file.

By default, the number of the line where the caret is currently located in the Editor panel is displayed.

Notes 1. Note the following, when specifying a symbol name:
- Either a function name or a variable name can be specified as a symbol name.
- Set the build tool's Property panel so that the cross reference information is output ([Common Options] tab] >> [Output File Type and Path] category >> [Output cross reference information] property >> [Yes(-Xcref)]), and then run and complete a build.
- If an error in building occurs, the cross reference information before the error occurred is used.

2. Note the following, when specifying an address:
- Enclose the value of an address in brackets "[ ]".
- Run and complete a build.
- If an error in building occurs, the information before the error occurred is used.

(b) Click the [OK] button

Caret is moved to the specified line.

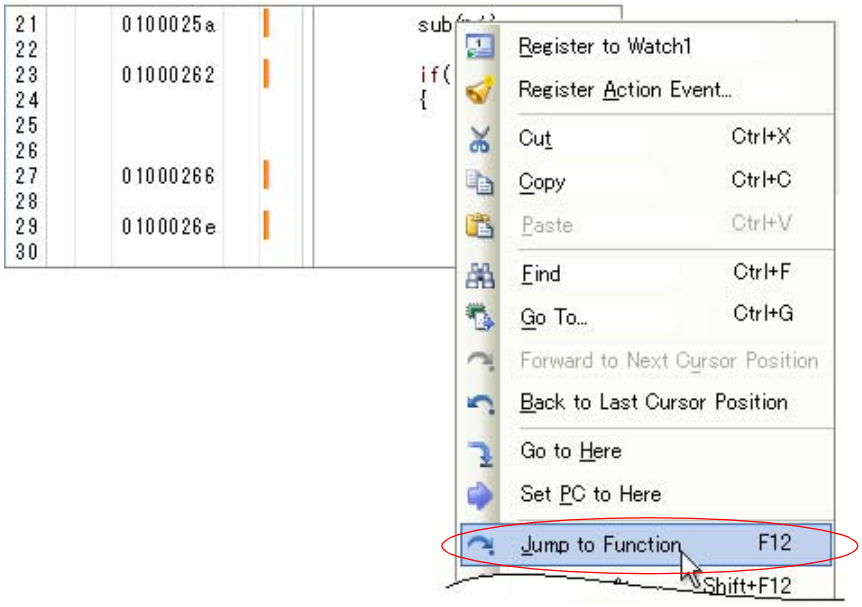
(7) Jump to functions

It automatically recognizes the currently selected characters or the word at the caret position as the function name and jumps to the first executable line of the target function.

Select [Jump to Function] from the context menu after moving the caret to the target function on the source text.

Caution When multiple statements are described in a line, a jump to an illegal location may be made.

Figure 2-61. Jump to Functions



Note that this function is available only when the following conditions are satisfied for each specific build tool.

(a) When CC-RH is used

- When disconnected from the debug tool:
  - The type of the project specified as the active project is "Application".
  - The target function is a global function.
  - The target function is defined in a file that is specified as the first file in the [Download files] property. In addition, the file includes the symbol information.
- When connected to the debug tool and downloaded the load module file:
  - The downloaded load module file includes the symbol information for the function.
  - Calling the target function from the file corresponding to the address of the program counter (PC). For example, a jump to a static function defined other than in the file corresponding to the address of the program counter (PC) is not possible.

Remark If functions with the same name exist, then the Jump to Function dialog box will be opened, and you can select the jump destination function.
Note, however, that this function is enabled only when the build tool's Property panel is set to output the cross reference information ([Common Options] tab] >> [Output File Type and Path] category >> [Output cross reference information] property >> [Yes(-Xcref)]).

(b) When an external build tool is used

- The target function resides in an active project.
- A file with the symbol information is selected for the [Download files] property (when the file is in the format other than the load module file, the setting for downloading the symbol information is required (see "(4) Perform source level debugging with files other than the load module file format")). In case it is disconnected from the debug tool, the above file is specified as the first file in the [Download files] property.

Caution A jump to a static function cannot be made when disconnected from the debug tool.

Remark The judgement of words will depend on the build tool being used.

(8) Jump to a desired line (tag jump)

If the information of a file name, a line number and a column number exist in the line at the caret position, you can open the file in another Editor panel and jump to the corresponding line and the corresponding column (if the Editor panel is already open, you can jump to the panel).

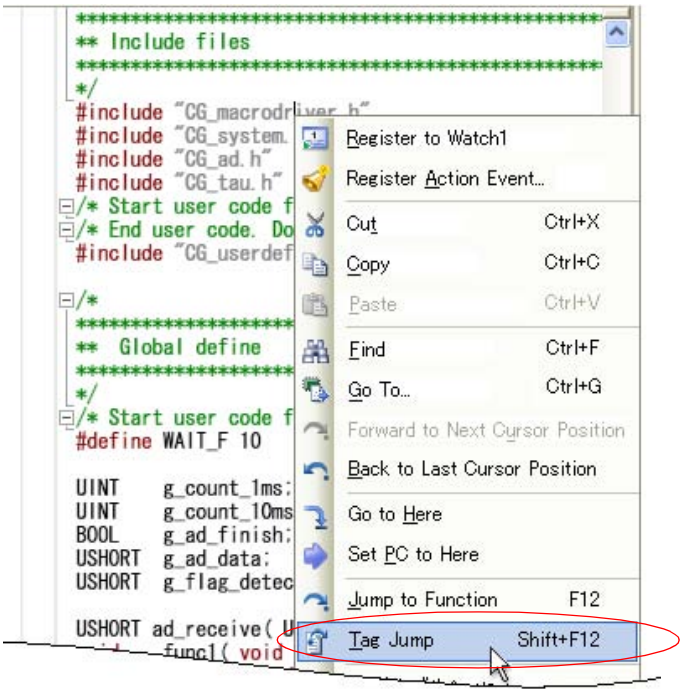
Select [Tag Jump] from the context menu after moving the caret to the line on the source text.

The tag jump is operated as follows:

Table 2-3. Operation of Tag Jump

Example of Character String	Operation
C:\work\src.c	Jumps to the top line of the file "C:\work\src.c".
Tmp\src.c	Jumps to the top line of the file "Tmp\src.c". (The reference point of the path is the project folder.)
C:\work\src.c(10)	Jumps to the tenth line from the top of the file "C:\work\src.c".
C:\"work sub\src.c"(10)	Jumps to the tenth line from the top of the file "C:\"work sub\src.c"".
C:\work\src.c(10,5)	Jumps to the fifth column of the tenth line from the top of the file "C:\work\src.c".

Figure 2-62. Tag Jump



Remarks 1. Jumps are case-insensitive.

2. The reference point of the path is the project folder in which the file is registered. If the file is not registered in any project, the reference point of the path will be the active folder.

3. Path specifications (path/file names) including space characters must be enclosed in "".

(9) Register a bookmark

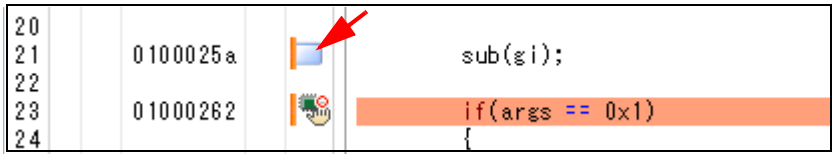
You can register a bookmark to the line at the current caret position by clicking the button on the bookmark toolbar. Once a bookmark is registered, the bookmark ( ) is displayed in the [Main] area.

When this operation is performed at a place where a bookmark is already being registered, that bookmark is deleted. Up to 50 bookmarks can be registered in one Editor panel.

Cautions 1. **When the Mixed display mode is selected, bookmarks cannot be registered nor displayed.**

2. **After a line with a bookmark is deleted, the bookmark cannot be restored even if the [Edit] menu >> [Undo] is selected.**

Figure 2-63. Register Bookmark



Remarks 1. The bookmark information is saved in the currently open project file and restored when that project is re-opened. Therefore, if bookmarks are set in a file that does not belong to the project, those bookmarks will not be restored.

2. Clicking on the and buttons on the bookmark toolbar moves the caret to the previous and next bookmarks, respectively. Note that the bookmarks are listed in the order of their registration (not in the order of line numbers).

3. Bookmarks currently being registered are listed on the Bookmarks dialog box that is opened by clicking the button on the bookmark toolbar.

2.6.2 Display the result of disassembling

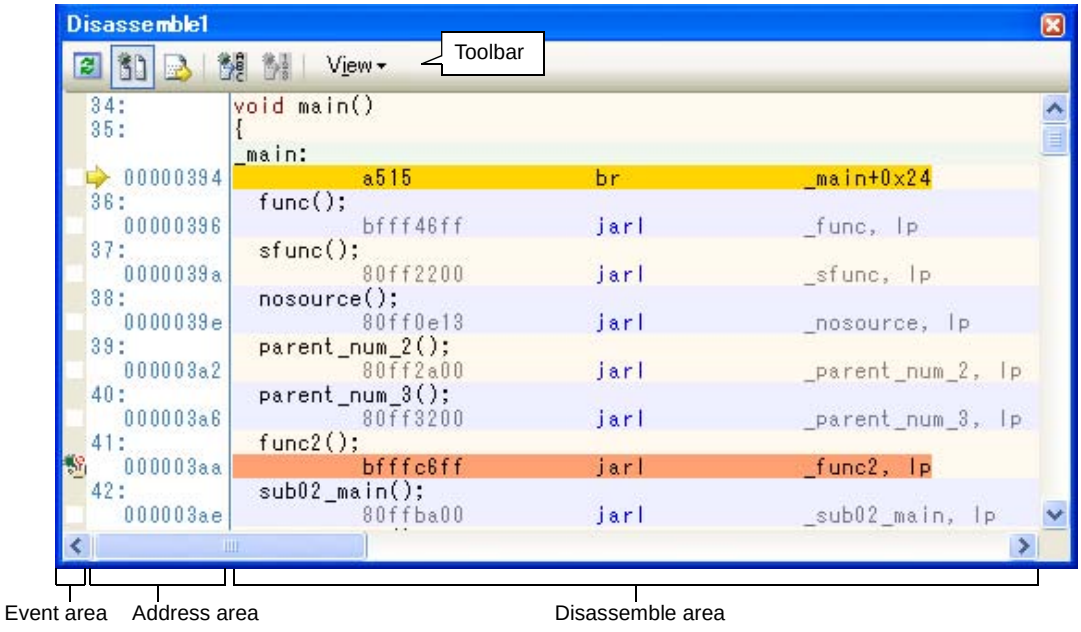
The result of disassembling the downloaded program (disassembled text) is displayed in the Disassemble panel below.

Select [View] menu >> [Disassemble] >> [Disassemble*1 - 4*].

The maximum of 4 Disassemble panels can be opened. Each panel is identified by the names "Disassemble1", "Disassemble2", "Disassemble3" and "Disassemble4" on the titlebar.

For details on the contents and function in each area, see the section for the Disassemble panel.

Figure 2-64. Display Result of Disassembling (Disassemble Panel)



Remark You can set the scroll range of the vertical scroll bar on this panel via the Scroll Range Settings dialog box which is opened by clicking the button from [View] on the toolbar.

This section describes the following.

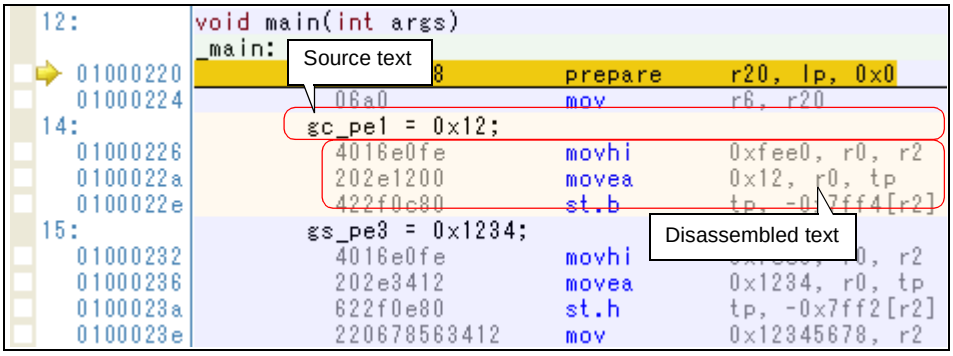
(1) Change display mode

You can change the display mode of the Disassemble panel by clicking the button (toggle) on the toolbar.

- Mixed display mode

In this display mode (default), the disassembled text is displayed combined with the source text.

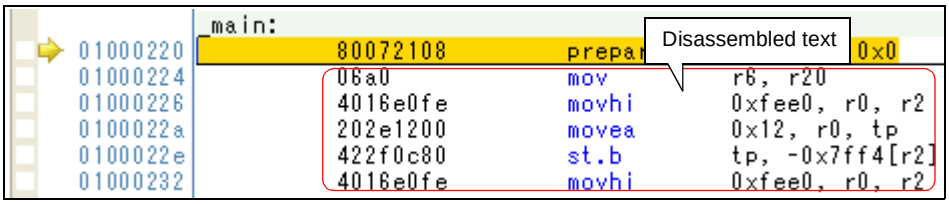
Figure 2-65. Mixed Display Mode (Disassemble Panel)



- Dissassemble display mode

In this display mode, the source text is hidden and only the disassembled text is displayed.

Figure 2-66. Disassemble Display Mode (Disassemble Panel)



(2) Change display format

The display format of the disassemble area can be changed using buttons below on the toolbar.

View	The following buttons to change the display format are displayed.
	Displays the offset value of the label. The offset value from the nearest label is displayed when a label is defined for the address.
	Displays the address value as the result of disassembling in the format "symbol + offset value" (default). Note that when a symbol has been defined as the address value, only the symbol is displayed.
	Displays the name of the register by its function name (default).
	Displays the name of the register by its absolute name.

(3) Move to the specified address

You can move to the specified address in the disassembled text in the Go to the Location dialog box which opens when selecting [Go to...] from the context menu.
In this dialog box, follow the steps below.

Figure 2-67. Move to Specified Address in Disassembled Text (Go to the Location Dialog Box)

(a) Specify [Address/Symbol]

Specify the address you want to move the caret to.
You can either type an address expression directly into the text box (up to 1024 characters), or select them from the input history via the drop-down list (up to 10 items).

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

(b) Click the [OK] button

Caret is moved to the specified address.

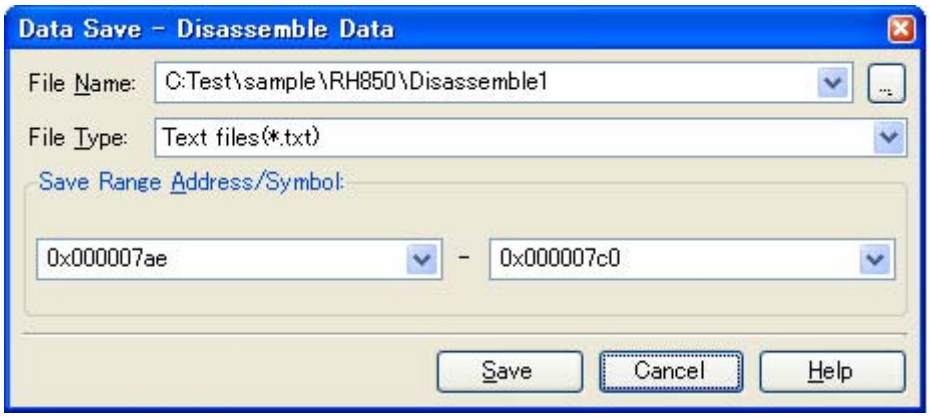
(4) Move to the symbol defined location

You can move the caret to the address where the symbol is defined.
Click the button on the toolbar after moving the caret to the instruction which refers to the symbol.
Furthermore, click the button on the toolbar following the previous operation returns the caret to the instruction which refers to the symbol at previous caret is defined.

(5) Save the disassembled text contents

Contents of the disassembled text can be saved in text files (*.txt)/CSV files (*.csv).
When saving to the file, the latest information is acquired from the debug tool, and it is saved in accordance with the display format on this panel.
The Data Save dialog box can be opened by selecting the [File] menu >> [Save Disassemble Data As...] (when this operation takes place with the range selected on the panel, the disassembled data can be saved only for the selected range).
In this dialog box, follow the steps below.

Figure 2-68. Save Disassembled Text Contents (Data Save Dialog Box)



(a) Specify [File Name]

Specify the name of the file to save.

You can either type a filename directly into the text box (up to 259 characters), or select one from the input history via the drop-down list (up to 10 items).

You can also specify the file by clicking the [...] button, and selecting a file via the Select Data Save File dialog box.

(b) Specify [File Type]

Select the format in which to save the file from the following drop-down list.

The following file formats can be selected.

List Item	Format
Text files (*.txt)	Text format (default)
CSV (Comma-Separated Variables)(*.csv)	CSV format[Note]

Note The data is saved with entries separated by commas (,).
If the data contains commas, each entry is surrounded by double quotes "" in order to avoid illegal formatting.

(c) Specify [Save Range Address/Symbol]

Specify the range of addresses to save via "start address" and "end addresses".

Directly enter hexadecimal number/address expression in each text box or select from the input history displayed in the drop-down list (up to 10 items).

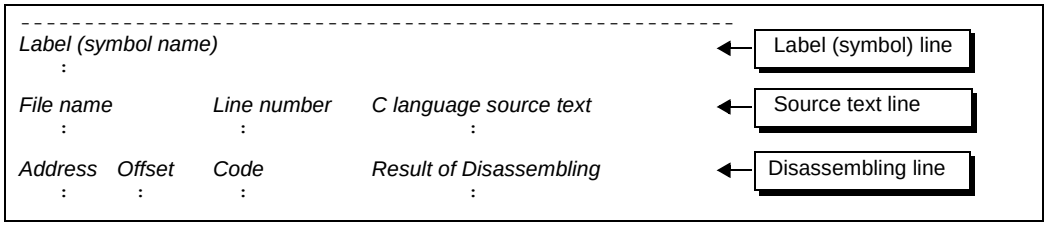
If a range is selected in the panel, that range is specified as the default. If there is no selection, then the range currently visible in the panel is specified.

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").

(d) Click the [Save] button

Disassembling data is saved in the specified file with the specified format.

Figure 2-69. Output Example of Disassembling Data



Remarks 1. When the contents of the panel are overwritten and saved by selecting the [File] menu >>[Save Disassemble Data], the Disassemble panels (Disassemble1-4) are handled individually for these respectively. In addition, saving range is same as the previously specified address range.

2. You can print the current screen image of this panel by selecting the [File] menu >> [Print...].

2.6.3 Run a build in parallel with other operations

CubeSuite+ can automatically start a build when one of the following events occurs (rapid build function).

- For other than the debug-only project

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

- For the debug-only project

- When you have edited and saved the C source file, assembly source file and header file that are added to the debug-dedicated project
- When a C source file, assembly source file, or header file has been added to or removed from the debug-dedicated project
- When the property of the debug-dedicated project is changed

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

Caution When an external text editor is used, check the [Observe registered files changing] check box on the [General - Build/Debug] category in the Option dialog box to enable this function.

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

2. Enable/Disable setting of the rapid build applies to the entire project (main project and subprojects).

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

2.6.4 Perform line assembly

Instructions and code displayed in the Disassemble panel can be edited (line assembly).
This section describes the following.

(1) Edit instructions
(2) Edit code

(1) Edit instructions

Follow the steps below to edit instructions.

(a) Switch to edit mode

Double-click the instruction to edit or select [Edit Disassemble] from the context menu after moving the caret to the instruction to edit.

(b) Edit instructions

Use keyboard to directly edit the instructions.

(c) Write to memory

Press the [Enter] key to line assemble the edited instructions after editing. The code is automatically written to the memory.

If the edited instruction is invalid, the instruction is shown in red and will not be written to the memory.

If there is a space because of overwriting the displayed result of disassembling by another instruction, its byte number is automatically compensated with nop instruction as follows:

Examples 1. Overwriting the prepare instruction (8-byte instruction) with the jarl instruction (4-byte instruction)

Before editing	0432	mov	0x4, r6
	1d38	mov	r29, r7
	8f071b0effff0000	prepare	r20, r21, r22, 0x1c, 0x0000ffff
	0132	mov	0x1, r6
After editing	0432	mov	0x4, r6
	1d38	mov	r29, r7
	bfffe265	jarl	0x100, lp
	0000	nop	
	0000	nop	
	0132	mov	0x1, r6

2. Overwriting the mov instruction (2-byte instruction) with the jarl instruction (4-byte instruction)

Before editing	0432	mov	0x4, r6
	1d38	mov	r29, r7
	8f071b0effff0000	prepare	r20, r21, r22, 0x1c, 0x0000ffff
	0132	mov	0x1, r6
After editing	0432	mov	0x4, r6
	bfffe265	jarl	0x100, lp
	0000	nop	
	0000	nop	
	0000	nop	
	0132	mov	0x1, r6

Caution Handling the prepare instruction and dispose instruction

The following table shows the instruction formats of the prepare instruction and dispose instruction. The operand "list12" comprises 12-bit value where a different register is assigned to each bit.

Instruction format of the prepare instruction	`prepare list12, imm5`
	`prepare list12, imm5, sp/imm`
Instruction format of the dispose instruction	`dispose imm5, list12`
	`dispose imm5, list12, [reg1]`

When displaying the results of disassembling the prepare instruction and dispose instruction in the Disassemble panel, the corresponding register names for the operand "list12" are displayed instead of its values as shown in the following examples.

Examples 1. When the code is "0xbf, 0x07, 0xe1, 0xff" (4-byte prepare instruction)

View	`prepare r20, r21, r22, r23, r24, r25, r26, r27,r28, r29, r30, r31, 0x20`
Syntax	`prepare 0xfff, 0x20`

2. When the code is "0x90, 0x07, 0xbb, 0xaa 0xff, 0xff, 0xff, 0xff" (8-byte prepare instruction)

View	`prepare r20, r22, r24, r26, r28, r31, 0x20, 0x7fffffff`
Syntax	`prepare 0x555, 0x20, 0x7fffffff`

3. When the code is "0x51, 0x06, 0xe0, 0xff" (4-byte dispose instruction)

View	`dispose 0x20, r20, r21, r22, r23, r24, r25, r26, r27, r28, r29, r30, r31`
Syntax	`dispose 0x20, 0xfff`

4. When the code is "0x50, 0x06, 0xaa, 0xaa" (4-byte dispose instruction)

View	`dispose 0x20, r20, r22, r24, r26, r28, r31, [r10]`
Syntax	`dispose 0x20, 0x555, [r10]`

Note, however, that it is possible to specify both the value and the register name for the operand "list12" when line assembling the prepare instruction and dispose instruction.

Examples 1. In both of the cases (1) and (2) below, the same set of values "0x91, 0x07, 0xe1, 0xff" will be generated as a result of line assembly.

(1)	`prepare r20, r21, r22, r23, r24, r25, r26, r27,r28, r29, r30, r31, 0x20`
(2)	`prepare 0xfff, 0x20`

2. In both of the cases (1) and (2) below, the same set of values "0xbe, 0x07, 0xbb, 0xaa 0xff, 0xff, 0xff, 0x7f" will be generated as a result of line assembly.

(1)	`prepare r20, r22, r24, r26, r28, r31, 0x20, 0x7fffffff`
(2)	`prepare 0x555, 0x20, 0x7fffffff`

3. In both of the cases (1) and (2) below, the same set of values "0x51, 0x06, 0xe0, 0xff" will be generated as a result of line assembly.

(1)	`dispose 0x20, r20, r21, r22, r23, r24, r25, r26, r27, r28, r29, r30, r31`
(2)	`dispose 0x20, 0xfff`

4. In both of the cases (1) and (2) below, the same set of values "0x50, 0x06, 0xaa, 0xaa" will be generated as a result of line assembly.

(1)	`dispose 0x20, r20, r22, r24, r26, r28, [r10]`
(2)	`dispose 0x20, 0x555, [r10]`

(2) Edit code

Follow the steps below to edit code.

(a) Switch to edit mode

Double-click the code to edit or select [Edit Code] from the context menu after moving the caret to the code to edit.

(b) Edit code

Use keyboard to directly edit the code.

(c) Write to memory

Press the [Enter] key to write the code to the memory after editing.
If the edited instruction is invalid, the instruction is shown in red and will not be written to the memory.
When the code is written to the memory, the result of disassembling is also updated.

2.7 Select a Core (PE)

This section describes how to select the target core (PE: Processer Element) to be debugged when the selected microcontroller supports multi-core.

CubeSuite+ displays information regarding a core (PE*n*) by switching selection between the target cores to be debugged (see "2.7.1 Switching between cores (PEs)"). Multiple panels are not provided to display each PE*n* in a single dedicated panel.

The following shows the behavior of each CubeSuite+ function for a microcontroller that supports multi-core.

(1) Program execution control

Synchronous execution and synchronous break are available in all PEs in principle.

However, the behavior in step execution differs as follows:

- **[Full-spec emulator][E1][E20]**
  Each instruction is executed step by step.
- **[Simulator]**
  Stepped execution is synchronized with the operating frequency.

Caution Step execution is available only in the currently selected PE*n*.
Step execution in source level units, however, may also proceed on other PEs (PE*n*).

(2) Event occurrence

Events are automatically set as valid in all PEs.

Cautions 1. [Full-spec emulator][E1][E20]
In the Local RAM self area, breakpoints are set in the currently selected PE*n*.
2. [Simulator]
Access to the Local RAM self area generates an event regardless of which PE*n* accesses the area. Note that when the entity of the Local RAM self area is directly accessed, no event will occur.

(3) Information of the memory, registers, or variables

(a) Memory mappings

Memory mappings may differ depending on the currently selected PE*n*.

In this case, switching to another PE*n* displays the corresponding memory mappings in the [Memory] category on the [Debug Tool Settings] tab of the Property panel or the Memory Mapping dialog box.

(b) Range and values of memory

The same value is displayed or set regardless of which PE*n* is currently selected.

In the Local RAM self area, note that the value in the currently selected PE*n* is acquired and displayed or set.

(c) Register values (including IOR/PC)

The value in the currently selected PE*n* is acquired and displayed or set.

(d) Symbols (including watch-expressions/variable names)

The address and value are determined based on the PC value in the currently selected PE*n* (for example, even when a symbol is valid only in a certain PE, its address and value are determined based on the PC value in the currently selected PE*n*).

(e) Call stack information

The value in the currently selected PE*n* is acquired and displayed or set.

(4) Other functions

(a) Collection of execution history of programs

- **[Full-spec emulator][E1][E20]**

The operation differs depending on the specification of the [Trace target setting] property in the [trace] category on the [Debug Tool Settings] tab of the Property panel.

- When [Debug core only] is selected:

Trace data regarding the currently selected PE*n* is collected.

Therefore, to collect desired trace data, select the target PE*n* before executing the program (if PE*n* is switched after collecting trace data, the display in the Trace panel will not be updated).

- When [All core] is selected:

Trace data is collected in all PEs.

After collecting trace data, switching to another PE*n* displays the corresponding trace data in the Trace panel.

- **[Simulator]**

Trace data regarding the currently selected PE*n* is collected.

Therefore, to collect desired trace data, select the target PE*n* before executing the program (if PE*n* is switched after collecting trace data, the display in the Trace panel will not be updated).

(b) Measurement of execution time of programs

The execution time is measured in all PEs.

After measurement, switching to another PE*n* displays the corresponding measurement time.

(c) Coverage measurements

The coverage is measured for access in all PEs.

In the Local RAM self area, note that the measurement results will be displayed only for the access in the currently selected PE*n*.

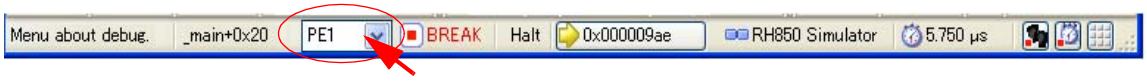
2.7.1 Switching between cores (PEs)

The core (PE) to be debugged can be selected in either of the following two ways.

(1) Switching through the statusbar

Select a desired PE*n* from the drop-down list (shown below) on the statusbar in the Main window.

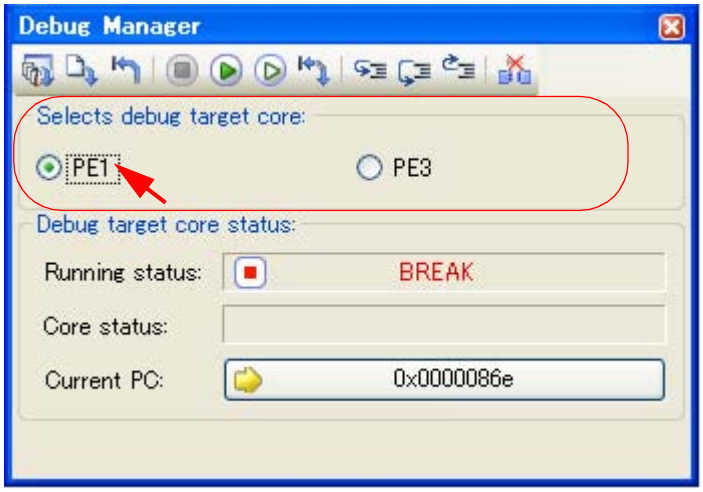
Figure 2-70. Statusbar on Main Window



(2) Switching through the Debug Manager panel

Select a desired PE*n* on the Debug Manager panel that are opened by selecting [Debug Manager] from the [View] menu.

Figure 2-71. Debug Manager Panel



2.8 Execute Programs

This section describes how to execute programs.

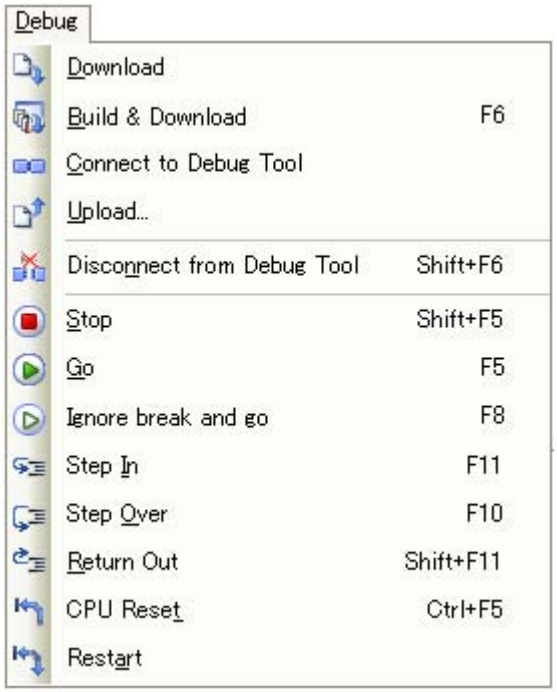
Main operations in this section are taken place from the debug toolbar or the [Debug] menu in the Main window, where commands to control the execution of programs are included.

Caution Items of the debug toolbar and the [Debug] menu are valid only while connected to the debug tool.

Remark For "program execution control" for a microcontroller that supports multi-core, see also to "2.7 Select a Core (PE)".

Figure 2-72. Debug Toolbar (Floating State)

Figure 2-73. [Debug] Menu



2.8.1 Reset microcontroller (CPU)

To reset CPU, click the button on the debug toolbar.

When CPU is reset, the current PC value is set to the reset address.

Remark You can automatically overwrite the value of I/O register/CPU register with the specified values after CPU reset under breaking (see "2.17 Use Hook Function" for details).

2.8.2 Execute programs

The following types of CubeSuite+ execution functions are provided.

Select any of the following operations according to the purpose of debugging.

See "2.9 Stop Programs (Break)" for details on how to stop the program in execution.

(1) Execute after resetting microcontroller (CPU)
(2) Execute from the current address
(3) Execute after changing PC value

Remark You can automatically overwrite the value of I/O register/CPU register with the specified values before starting program execution (see "2.17 Use Hook Function" for details).

(1) Execute after resetting microcontroller (CPU)

Click the  button on the debug toolbar.
Reset CPU and start execution of the program from the reset address.
When this operation is performed, the program continues to be executed until either of the following occurs:

- The  button has been clicked (see "2.9.2 Stop the program manually").
- The PC has reached a breakpoint (see "2.9.3 Stop the program at the arbitrary position (breakpoint)").
- A break event condition has been met (see "2.9.4 Stop the program at the arbitrary position (break event)" or "2.9.5 Stop the program with the access to variables/I/O registers").
- Other break causes have occurred.

Remark This operation is the same as when the  button is clicked after clicking the  button.

(2) Execute from the current address

Perform any of the following operations to start executing the program from the address at the current PC value.

(a) Normal execution

Click the  button on the debug toolbar.
When this operation is performed, the program continues to be executed until either of the following occurs:

- The  button has been clicked (see "2.9.2 Stop the program manually").
- The PC has reached a breakpoint (see "2.9.3 Stop the program at the arbitrary position (breakpoint)").
- A break event condition has been met (see "2.9.4 Stop the program at the arbitrary position (break event)" or "2.9.5 Stop the program with the access to variables/I/O registers").
- Other break causes have occurred.

(b) Execution ignoring break-related events

Click the  button on the debug toolbar.
When this operation is performed, the program continues to be executed until either of the following occurs:

- The  button has been clicked (see "2.9.2 Stop the program manually").
- Other break causes have occurred.

Remark If you have started the execution with this operation, the occurrence of Action event will also be ignored.

(c) Execution to the caret position

To start this operation, move the caret to the line/instruction to stop the program in the Editor panel/ Disassemble panel, then select [Go to Here] from the context menu.
When this operation is performed, the program continues to be executed until either of the following occurs:

- The PC has reached the address of the caret position.
- The  button has been clicked (see "2.9.2 Stop the program manually").

- Other break causes have occurred.

Caution **When the corresponding address of the line at the caret position does not exist, the program is executed to the corresponding address of the lower valid line (if the corresponding address does not exist, an error message will appear).**

Remark If you have started the execution with this operation, the occurrence of Action event will also be ignored.

(3) Execute after changing PC value

The program is executed after forcibly changing the current PC value to an arbitrary address.
To start this operation, move the caret to the line/instruction to start the program in the Editor panel/Disassemble panel, then select [Set PC to Here] from the context menu (the current PC value is set to the address of the line/instruction where the caret currently exists).
Then execute either one of the execution method described in "(2) Execute from the current address".

2.8.3 Execute programs in steps

When either of the following operation has occurred, the program will stop automatically after conducting step execution in the source level (1 line of source text) or in the instruction level (1 instruction).

Once the program is stopped, the contents of each panel will be updated automatically. As such, step execution is suited for debugging the program execution in transition either in source or instruction level.

The unit in which the program is step-executed depends on the setting as follows:

- When the  button on the Editor panel's toolbar is invalid (default):
  Step execution is conducted in source level.
  Note, however, that when the focus is in the Disassemble panel or the line information does not exist in the address specified by the current PC value, the step execution is conducted in instruction level.

- When the  button on the Editor panel's toolbar is valid:
  Step execution is conducted in instruction level.

Caution **The  button is only enabled if the Mixed display mode is selected on the Editor panel.**

Step execution is divided into the following types:

(1) Step in function (Step in execution)
(2) Step over function (Step over execution)
(3) Execute until return is completed (Return out execution)

Cautions 1. Breakpoints, break events, and action events that have been set do not occur during step execution.
2. An error message will appear while processing a function prologue or epilogue if the return address cannot be acquired.
3. [Full-spec emulator][E1][E20]
- Interrupts are not acknowledged during step execution.
- It will not go into standby mode during step execution.
4. [Simulator]
You may jump to an interrupt handler during step execution.

(1) Step in function (Step in execution)

When the function is called, the program is stopped at the top of the called function.
Click the  button on the debug toolbar to perform Step in execution.

Cautions 1. Step in execution for a function without the debug information is not possible.
2. If Step in execution is performed for the longjmp function, program execution may not complete and may wait for a time-out.
3. The beginning of the function (prologue processing) is not skipped. To skip prologue processing, perform Step in execution again.

(2) Step over function (Step over execution)

In the case of a function call by the jarl instruction, all the source lines/instructions in the function are treated as one step and executed until the position where execution returns from the function (step execution will continue until the same nest is formed as when the jarl instruction has been executed).
Click the  button on the debug toolbar to perform Step over execution.
In the case of an instruction other than jarl, operation is the same as when the  button is clicked.

Caution If Step over execution is performed for the longjmp function, program execution may not complete and may wait for a time-out.

(3) Execute until return is completed (Return out execution)

Step-execute the program so that the program will stop when it returns from the current function to the caller function. When the execution of source line/instruction that require checking has been completed, you can perform step execution using this instruction so that you can make the program return to the caller function without step executing the remaining instructions inside the function.
Click the  button on the debug toolbar to perform Return out execution.

Cautions 1. If Return out execution is performed in the main function, the program is stopped in the startup routine.
2. Return out execution cannot be performed immediately after stepping in a function.
3. Return out execution cannot be performed while processing a function prologue or epilogue.
4. If Return out execution is performed in a function that called the longjmp function, breaks may not occur.
5. If Return out execution is performed in a recursive function, the program will be executed in free-run mode.

2.9 Stop Programs (Break)

This section describes how to stop the program in execution.

Cautions 1. If a forced break is performed while in standby mode (HALT/STOP/IDLE), the current PC position will indicate the address of the next instruction after the standby mode instruction. This behavior differs depending on the debug tool used.

- [Full-spec emulator][E1][E20]
The forced break will release standby mode.

- [Simulator]
The forced break will not release standby mode.
It will appear that standby mode has been released. Check the CPU status on the Main window's statusbar to see if standby mode has been released.

2. [Full-spec emulator][E1][E20]
Do not decrease the voltage of the target system during a break. A reset that is generated by the low-voltage detector (LVI) or by power-on-clear (POC) during a break causes an incorrect operation of CubeSuite+ or communication errors.
A break during emulation of power supply off also causes communication errors.

Remarks 1. For "program execution control" or "event occurrence" for a microcontroller that supports multi-core, see also to "Select a Core (PE)".

2. When the program in execution is stopped, a statement of the cause of the break appears on the Statusbar in the Main window.

CubeSuite+ can stop the program in execution at the arbitrary position by using the following functions.

(1) Forced break function

Stops the program forcibly.

(2) Hardware break function

The debug tool consecutively checks the break condition while the program is in execution and stops the program when the condition is met.
This function is implemented using the debug tool resources.
If a Hardware Break event is set, the program will break before executing instruction at the specified address ("before execution" break).

Remark When a Hardware Break event (access-type) is used (see "(1) Set a break event (access type)"), "after execution" break will only follow the cases listed below.

- When the data condition is specified after selecting [Break Settings] >> [Set Read Break to] / [Set R/W Break to] from the context menu
- When a write access of the read-modify-write instruction is detected, after selecting [Break Settings] >> [Set Write Break to] / [Set R/W Break to] from the context menu

(3) Software break function [Full-spec emulator][E1][E20]

Temporarily replaces the instruction code for a specified address with a break instruction and stops the program when this instruction is executed.
If a Software Break event is set, the program will break before executing instruction at the specified address ("before execution" break).

Caution Since an instruction code is replaced by the break instruction, setting or deleting a software break event is followed by programming of the flash memory at the timing described below.
- **When the program execution is started (including the start of execution via [Ignore break and go] from the [Debug] menu)**
- **When the debug tool is disconnected from CubeSuite+**

2.9.1 Configure the break function [Full-spec emulator][E1][E20]

Before the break function can be used, it is necessary to make settings relating to the operation of a break.
This break operation can be configured in the [Break] category on the [Debug Tool Settings] tab of the Property panel.

Remark [Simulator]
The settings relating to the operation of a break are not necessary.

Figure 2-74. [Break] Category [Full-spec emulator][E1][E20]

Break	
Use software break	**Yes**
First using type of breakpoint	Hardware break
Stop emulation of peripherals when stopping	No

(1) [Use software break]
Select whether to use the Software break function [Full-spec emulator][E1][E20].
Select [Yes] to use the software break function (default: [No]).

Cautions 1. If this property is set to [No] after you have used the software break function, all software break events and Printf events that have been set will be disabled. Selecting [Yes] in this state does not automatically restore the events, so you will need to manually enable them.
2. This property cannot be changed during program execution.

(2) [First using type of breakpoint]
This property appears only when the [Use software break] property is set to [Yes].
Select from the following drop-down list the type of a breakpoint to use with priority when setting it with a one click operation of the mouse in the Editor panel/Disassemble panel.

Hardware break	Sets hardware breakpoint with priority, by using the Hardware break function (default). Once set, it is treated as a Hardware Break event (execution system).
Software break	Sets software breakpoint with priority, by using the Software break function [Full-spec emulator][E1][E20]. Once set, it is treated as a Software Break event.

Caution If the number of the set breakpoints of the specified type exceeds the limit settable (see "(1) Maximum number of enabled events"), a breakpoint of another type will be used.

(3) [Stop emulation of peripherals when stopping]
Select whether to terminate the peripheral emulation while stopping the program execution (Peripheral Break).
Select [Yes] to terminate (default: [No]).

2.9.2 Stop the program manually

The program in execution is forcibly stopped by clicking the  button on the debug toolbar (Forced break function).

2.9.3 Stop the program at the arbitrary position (breakpoint)

A breakpoint is one of the break events that can be set by one-clicking with the mouse.
The program in execution can be stopped at the arbitrary position easily by setting a breakpoint.
This section describes the following operations.

(1) Set a breakpoint
(2) Delete a breakpoint

(1) Set a breakpoint

Breakpoints can be set via the Editor panel/Disassemble panel in which the source text/disassembled text is displayed.
Within the Main area (Editor panel) or Event area (Disassemble panel) in which a valid address is displayed, click on the location where you want to set a breakpoint. A breakpoint whose type is being selected in the [First using type of breakpoint] property is set to the instruction at the start address corresponding to the clicked line.
When a breakpoint is set, the following event mark appears at the breakpoint location, and the source text line/ disassembled text line is highlighted.
It is interpreted as if a break event (Hardware Break or Software Break) has been set at the target address, and it is managed in the Events panel (see "2.16 Manage Events" for details).

Table 2-4. Event Marks of Breakpoint

Type of Breakpoint	Event Type	Event Mark
Hardware breakpoint	Hardware Break event[Note]	
Software breakpoint **[Full-spec emulator][E1][E20]**	Software Break event[Note]	

Note In the [Name] area of the Events panel, "Break" is displayed as the event type name.

Figure 2-75. Breakpoint Setting Example (Disassemble Panel)

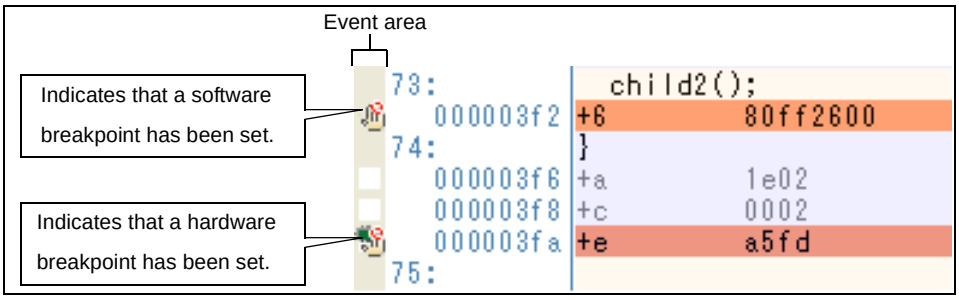
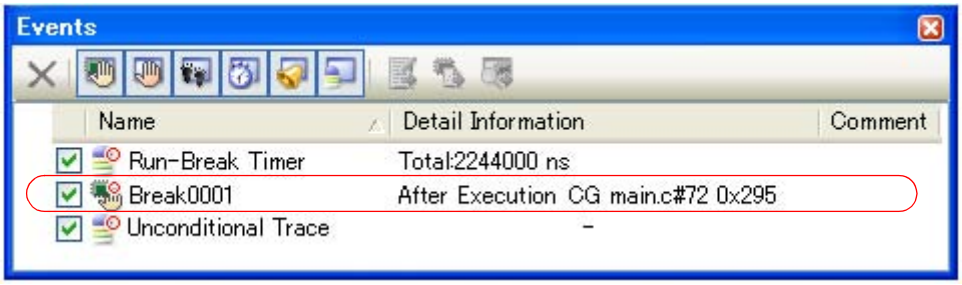


Figure 2-76. Example of Setting Breakpoint in Events Panel



Cautions 1. **Since a breakpoint is set as a break event and managed as a event, restrictions apply to the number of breakpoints that can be simultaneously set. Also see "2.16.6 Notes for setting events" for details on breakpoints (e.g. limits on the number of enabled events).**

2. Breakpoints can only be set at lines that have valid addresses.

3. [Full-spec emulator][E1][E20]
Software breakpoints can be set in only the code flash area.

Remarks 1. Event marks differ depending on the event state (see "2.16.1 Change the state of set events (valid/invalid)").
When an event is set at the point where other event is already set, the event mark ( ) is displayed meaning more than one event is set at the point.

2. [Full-spec emulator][E1][E20]
You can set hardware breakpoints/software breakpoints without depending on the selection of the [First using type of breakpoint] property by the operation described below.
Note, however, that "Operation1" is only available in the Disassemble panel.

Type	Operation1	Operation2
Hardware breakpoint	[Ctrl] + mouse click	Select [Break Settings] >> [Set Hardware Break] from the context menu.
Software breakpoint	[Shift] + mouse click	Select [Break Settings] >> [Set Software Break] from the context menu.

3. [Simulator]
The type of breakpoint that can be set is locked to hardware breakpoints.

(2) Delete a breakpoint

Click event marks displayed in the Editor panel/Disassemble panel to delete set breakpoints (the event mark will be erased).

2.9.4 Stop the program at the arbitrary position (break event)

The program in execution can be stopped at the arbitrary position by setting a break event (execution type).
This section describes the following operations.

(1) Set a break event (execution type)
(2) Delete a break event (execution type)

(1) Set a break event (execution type)

Perform this operation in the Editor panel/Disassemble panel in which the source text/disassembled text is displayed.
Follow the operation listed below from the context menu, in accordance with your desired event type, after moving the caret to the target line that has a valid address.

Event Type	Operation	Description
Hardware Break	Select [Break Settings] >> [Set Hardware Break]	Sets a Hardware Break event by using the Hardware break function.
Software Break **[Full-spec emulator] [E1][E20]**	Select [Break Settings] >> [Set Software Break]	Sets a Software Break event by using the Software break function [Full-spec emulator][E1][E20].

A break event is set to the instruction at the start address corresponding to the line of the caret position.
When a break event (execution type) is set, the following event mark is displayed in the event area of the line that an event is set., and the source text line or disassembled text line will be highlighted.
When you have performed this operation, the set break event is managed in the Events panel as a Hardware Break event (execution type)/Software Break event (execution type) (see "2.16 Manage Events" for details).

Table 2-5. Event Marks of Break Event (Execution Type)

Event Type	Event Mark
Hardware Break	
Software Break **[Full-spec emulator][E1][E20]**	

Figure 2-77. Break Event (Execution Type) Setting Example (Disassemble Panel)

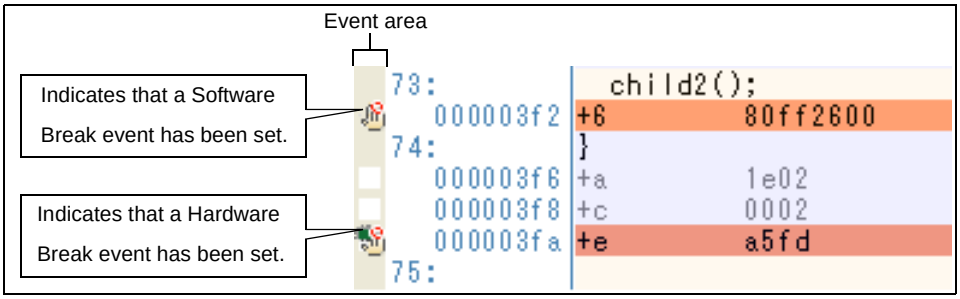
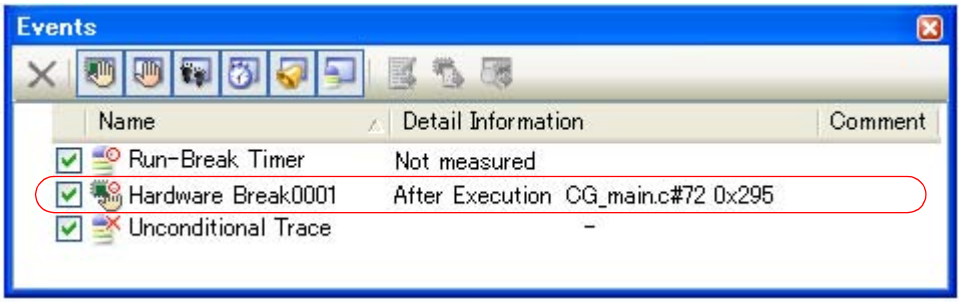


Figure 2-78. Example of Setting Hardware Break Event (Execution Type) in Events Panel



Cautions 1. When setting a break event (execution type), also see "2.16.6 Notes for setting events" for details (e.g. limits on the number of valid events).

2. [Full-spec emulator][E1][E20]
Software breakpoints can be set in only the code flash area.

Remark Event marks differ depending on the event state (see "2.16.1 Change the state of set events (valid/invalid)"). When an event is set at the point where other event is already set, the event mark ( ) is displayed meaning more than one event is set at the point.

(2) Delete a break event (execution type)

To delete a break event (execution type) you have set, click the event mark displayed in the Editor panel/ Disassemble panel.
Also, there is another way to delete a set break event. Select a Software Break event/Hardware Break event in the Events panel, and then click the button in the toolbar (see "2.16.4 Delete events").

2.9.5 Stop the program with the access to variables/I/O registers

By setting a break event with the access, the program can be stopped when an arbitrary variable or I/O register is accessed with the specified type.

You can also limit the accessed value.

The following types can be specified with the access.

Table 2-6. Types of Accesses to Variables

Access Type	Description
Read	The program is stopped with the read access to (after reading) the specified variable/I/O register.
Write	The program is stopped with the write access to (after writing) the specified variable/I/O register.
Read/Write	The program is stopped with the read access/write access to (after reading or writing) the specified variable/I/O register.

Caution The program is not stopped with the access via DMA (Direct Memory Access).

This section describes the following.

(1) Set a break event (access type)
(2) Delete a break event (access type)

(1) Set a break event (access type)

Use one of the following methods to set a break event (access type) that stops programs with the access to a variable/I/O register.

Caution Also see "2.16.6 Notes for setting events" for details on breakpoints (e.g. limits on the number of enabled events).

(a) Set a break event to a variable/I/O register in the Editor panel/Disassemble panel

Perform this operation in the Editor panel/Disassemble panel in which the source text/disassembled text is displayed.

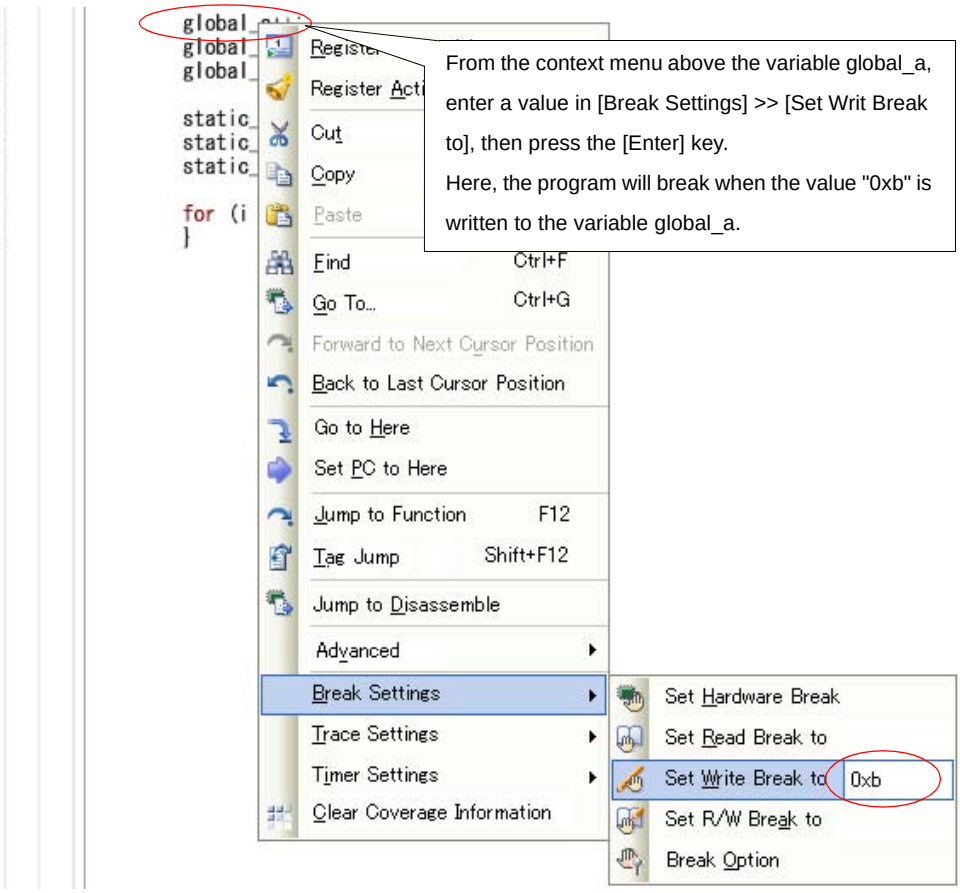
Follow the operation listed below from the context menu, in accordance with your desired access type, after selecting an arbitrary variable or I/O register on the source text or the disassembled text. Note, however, that only global variables, static variables inside functions, and file-internal static variables can be used.

Access Type	Operation
Read	Select [Break Settings] >> [Set Read Break to], and then press the [Enter] key.
Write	Select [Break Settings] >> [Set Write Break to], and then press the [Enter] key.
Read/Write	Select [Break Settings] >> [Set R/W Break to], and then press the [Enter] key.

At this time, if you have specified a value in the text box in the context menu, break will occur only when the specified value is used for the reading, writing or reading/writing. On the other hand, if no value is specified, reading., writing or reading/writing the selected variable by any value will cause the break to occur.

Cautions 1. Variables within the current scope can be specified.
2. Variables or I/O register at lines that have no valid addresses cannot be used for break events.

Figure 2-79. Example of Setting Break Event (Access Type) on Variable in Editor Panel



(b) Set a break event (access type) to a registered watch-expression

You can set break events in the Watch panel.

Follow the operation listed below from the context menu, in accordance with your desired access type, after selecting the registered watch-expression (multiple selections not allowed).

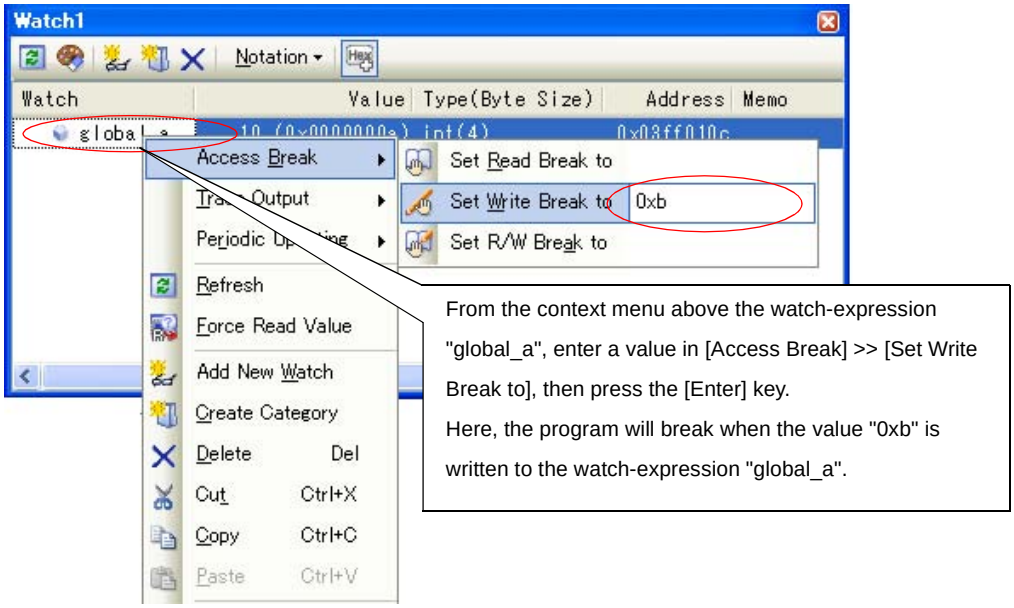
Note, however, that only global variables, static variables inside functions, file-internal static variables, and I/O register can be used.

Access Type	Operation
Read	Select [Access Break] >> [Set Read Break to], and then press the [Enter] key.
Write	Select [Access Break] >> [Set Write Break to], and then press the [Enter] key.
Read/Write	Select [Access Break] >> [Set R/W Break to], and then press the [Enter] key.

At this time, if you have specified a value in the text box in the context menu, break will occur only when the specified value is used for the reading., writing or reading/writing. On the other hand, if no value is specified, reading., writing or reading/writing the selected watch-expression by any value will cause the break to occur.

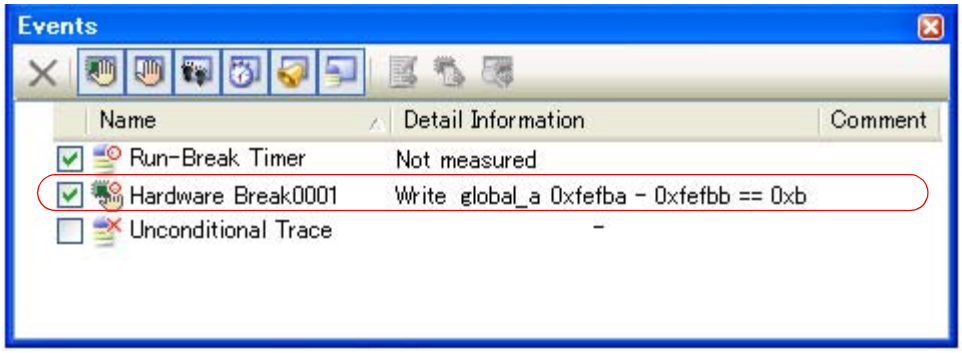
Caution A watch-expression within the current scope can be specified.
To target a watch-expression outside the current scope, select a watch-expression with a specified scope.

Figure 2-80. Example of Setting Hardware Break Event (Access Type) on Watch-Expression



When you have performed the above operation, the set break event (access type) is managed in the Events panel as a Hardware Break event (access type) (see "2.16 Manage Events" for details).

Figure 2-81. Example of Setting Hardware Break Event (Access Type) in Events Panel



(2) Delete a break event (access type)

To delete a break event (access type) you have set, select a Hardware Break event in the Events panel, and then click the button in the toolbar (see "2.16.4 Delete events").

2.9.6 Other break causes

The cause of the break other than the described above is as follows:

Moreover, you can confirm the break cause with the Status message on the statusbar in the Main window.

Table 2-7. Other Break Causes

Break Cause	Debug Tool		
	Full-spec emulator	E1/E20	Simulator
Full of the trace memory[Note 1]	✓	✓	✓
An access to non-mapped area	-	-	✓
A writing to write-protected area	-	-	✓
An occurrence of Temporary Break[Note 2]	✓	✓	✓
Step execution count-over	✓	✓	✓

Notes 1. The operation depends on the setting of the [Operation after trace memory is full] property in the [Trace] category on the [Debug Tool Settings] tab of the Property panel.

2. A break that is internally used by CubeSuite+. (Users cannot use it.)

2.10 Display/Change the Memory, Register and Variable

This section describes how to display/change the memory, register and variable.

Remark For "Information of the memory, registers, or variables" for a microcontroller that supports multi-core, see also to "Select a Core (PE)".

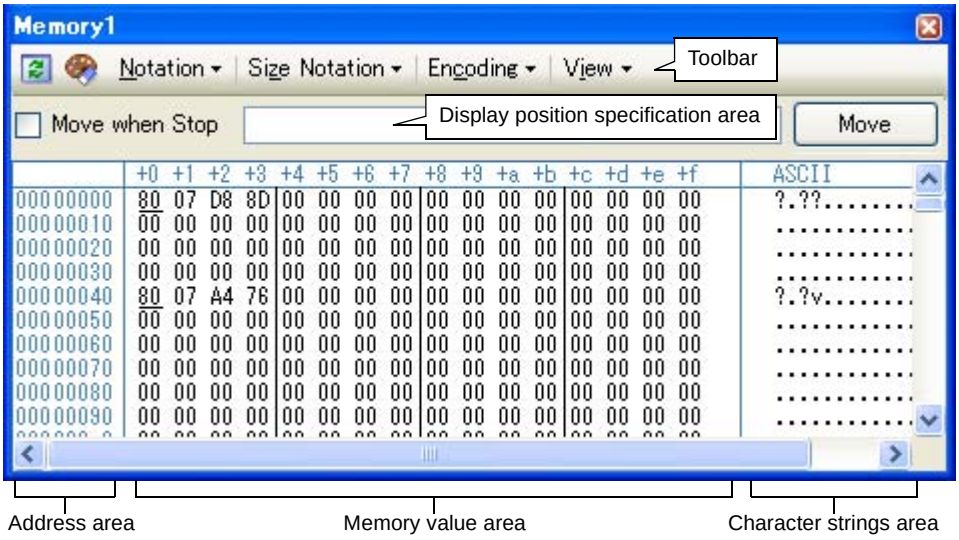
2.10.1 Display/change the memory

The contents of the memory can be displayed and its values can be changed in the Memory panel below.

Select the [View] menu >> [Memory] >> [Memory*1 - 4*]. The maximum of 4 Memory panels can be opened. Each panel is identified by the names "Memory1", "Memory2", "Memory3" and "Memory4" on the titlebar.

For details on the contents and function in each area, see the section for the Memory panel.

Figure 2-82. Display Memory Contents



Remark You can set the scroll range (as start and end address) of the vertical scroll bar on this panel via the Scroll Range Settings dialog box which is opened by clicking the button from [View] on the toolbar.

This section describes the following.

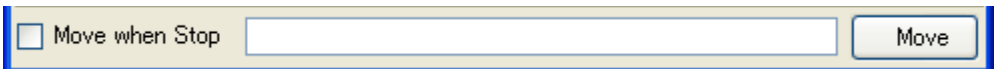
(1) Specify the display position
(2) Change display format of values
(3) Modify the memory contents
(4) Display/modify the memory contents during program execution
(5) Search the memory contents
(6) Modify the memory contents in batch (initialize)
(7) Save the memory contents

(1) Specify the display position

It is possible to specify the display start position of the memory contents by specifying an address expression in the display position specification area (starting with address 0x0 by default).

Remark An offset value of the display start position of memory values can be set via the Address Offset Settings dialog box that is opened by selecting [Address Offset Value Settings...] from the context menu.

Figure 2-83. Display Position Specification Area (Memory Panel)



(a) Specify an address expression

Directly enter the address expression of the memory value address to display in the text box. You can specify an input expression with up to 1024 characters. The result of the expression is treated as the display start position address.

Note that address values greater than the microcontroller address space cannot be specified.

Remarks 1. A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

2. If the specified address expression is the symbol and its size can be recognized, everything from the start address to the end address of that symbol is displayed selected.

(b) Specify automatic/manual evaluation of the address expression

The timing to change the display start position can be determined by specifying in the [Move when Stop] check box and the [Move] button.

[Move when Stop]	☑	The caret is moved to the address which is automatically calculated from the address expression after the program is stopped.
	☐	The address expression is not automatically evaluated after the program is stopped. Click the [Move] button to manually evaluate the address expression.
[Move]		When the [Move when Stop] check box is not checked, click this button to evaluate the address expression and move the caret to the result address of the evaluation.

(2) Change display format of values

The display format of the address area/memory value area/character strings area can be changed using buttons below on the toolbar.

Notation	The following buttons to change the notation of memory values are displayed.
Hex	Displays memory values in hexadecimal number (default).
S_Dec	Displays memory values in signed decimal number.
U_Dec	Displays memory values in unsigned decimal number.
Oct	Displays memory values in octal number.
Bin	Displays memory values in binary number.
Size Notation	The following buttons to change the notation of sizes of memory values are displayed.
4	Displays memory values in 4-bit width.
8	Displays memory values in 8-bit width (default).
16	Displays memory values in 16-bit width. Values are converted depending on the endian of the target memory area.
32	Displays memory values in 32-bit width. Values are converted depending on the endian of the target memory area.
64	Displays memory values in 64-bit width. Values are converted depending on the endian of the target memory area.

Encoding	The following buttons to change the encoding of character strings are displayed.
	Displays character strings in ASCII code (default).
	Displays character strings in Shift-JIS code.
	Displays character strings in EUC-JP code.
	Displays character strings in UTF-8 code.
	Displays character strings in UTF-16 code.
	Displays character strings as a single-precision floating-point value[Note].
	Displays character strings as a double-precision floating-point value[Note].
	Displays character strings as a complex number of single-precision floating-point[Note].
	Displays character strings as a complex number of double-precision floating-point[Note].
	Displays character strings as an imaginary number of single-precision floating-point[Note].
	Displays character strings as an imaginary number of double-precision floating-point[Note].
View	The following buttons to change the display format are displayed.
	Opens the Scroll Range Settings dialog box to set the scroll range for this panel.
Column Number Settings...	Opens the Column Number Settings dialog box to set the number of view columns in the memory value area.
Address Offset Value Settings...	Opens the Address Offset Settings dialog box to set an offset value for addresses displayed in the address area.

Note For details on the display of a floating-point value, see the section for the Memory panel.

(3) Modify the memory contents

The memory values can be edited.

Directly edit from the keyboard after moving the caret to the line to modify in memory value area/characters area. The color of the memory value changes when it is in editing. Press the [Enter] key to write the edited value to the target memory (if the [Esc] key is pressed before the [Enter] key is pressed, the editing is cancelled).

However, the character string that can be inputted during the editing is limited to that character string that can be handled by the display notation that has been currently specified. In the character strings area, modification can only be made with "ASCII" character code.

This operation can be taken place while the program is in execution. See "(4) Display/modify the memory contents during program execution" for details on how to operate it.

When you modify the values, be aware of the following examples.

Examples 1. The value exceeds the upper limit of the display bit wide

If you edit the display value "105" as "1" to "3" in the decimal 8-bit display, the value will be changed to the upper limit of "127".

2. The symbol, "-" is entered between numbers

If you edit the display value "32768" as "32-68" with signed decimal 16-bit display, "3" and "2" are changed to the blank and the value is changed to "-68".

3. The blank symbol (space) is entered between numbers

If you edit the display value "32767" as "32 67", "3" and "2" are changed to the blank and the value is changed to "67".

4. The same value is entered

Even if the same value as the current memory value is specified, the specified value is written to the memory.

(4) Display/modify the memory contents during program execution

The Memory panel/Watch panel has the real-time display update function that can update/modify the display contents of the memory/watch-expression in real-time while executing the program.

Using the real-time display update function allows you to display/modify the value of memory/watch-expression not only while the program is stopped, but also in execution.

The real-time display update function is realized by the CPU's/debug tool's RRM function (reading) [Simulator], RAM monitor function (reading) [Full-spec emulator][E1][E20] and DMM function (modifying). Each function has a different area that can be used for reading and writing.

Enable the real-time display update function by making the basic settings below on the [Debug Tool Settings] tab of the Property panel.

Table 2-8. Basic Settings for Real-time Display Update Function

Category	Property	Setting Value
[Access memory while running]	[Update display during the execution]	[Yes] (default)
	[Display update interval[ms]]	[*Integer number between 100 and 65500*]

Cautions 1. Local variables are not subject to the real-time display update function.

2. When a 2-, 4-, or 8-byte variable is to be read through the RRM or RAM monitor function, the process of assigning a value to the variable may be divided into two steps.
If reading of the variable takes place between the two steps, an incorrect value may be read out because the assignment is not completed.

3. The contents of memory or watch expressions can be read for access in all PEs when the selected microcontroller supports multi-core. In the Local RAM self area, note that they can be read only for the access in the currently selected PE*n*.

Remark See "(3) Modify the memory contents" or "(6) Modify the contents of watch-expressions" for details on how to modify values in the Memory panel/Watch panel.

(a) RRM function (reading) [Simulator]

This function is used to read the contents of memory or of watch-expressions in real-time during execution of a program.

The following area can be read by the RRM function. Memory and watch-expressions allocated to this area can always be displayed in real-time.

Table 2-9. Target Area of RMM Function

Area	Simulator
Internal ROM	✓
Internal RAM	✓
Peripheral I/O area	-
Data flash	-
Emulation memory	-
Target memory	-
CPU register	✓ Note
I/O register (except read-protected I/O registers)	✓

Note Impossible during tracer/timer execution

(b) RAM monitor function (reading) [Full-spec emulator][E1][E20]

This function is used to read the contents of memory or a watch-expression via the CPU's RAM monitor function.

The following area can be read by the RAM monitor function.

Caution If CPU status shifts to the standby mode (HALT/STOP/IDLE) mode, a monitor time-out error will occur.

Table 2-10. Target Area of RAM Monitor Function

Area	Full-spec emulator	E1/E20
Internal ROM	-	-
Internal RAM	✓	✓
Peripheral I/O area	-	-
Data flash (except ID tag)	-	-
Target memory	-	-
CPU register	-	-
I/O register	-	-

Note that to enable the RAM monitor function, the setting below is required in addition to the Basic Settings for Real-time Display Update Function.

Debug Tool	Category	Property	Setting Value
Full-spec emulator E1/E20	[Access memory while running]	[Access during the execution]	[Yes]

(c) DMM function (modifying)

This function is used to write to the memory or watch-expressions in real-time during execution of a program.

The following area can be modified by the DMM function.

Cautions 1. If a value is written through the DMM function, the atomicity cannot be guaranteed.

2. If CPU status shifts to the standby mode (HALT/STOP/IDLE) mode, a monitor time-out error will occur.

Table 2-11. Target Area of DMM Function

Area	Full-spec emulator	E1/E20	Simulator
Internal ROM	-	-	✓
Internal RAM	✓	✓	✓
Peripheral I/O area	-	-	-
Emulation memory	-	-	-
Target memory	-	-	-
CPU register	-	-	✓ Note

Area	Full-spec emulator	E1/E20	Simulator
I/O register (except read-protected I/O registers)	-	-	✓

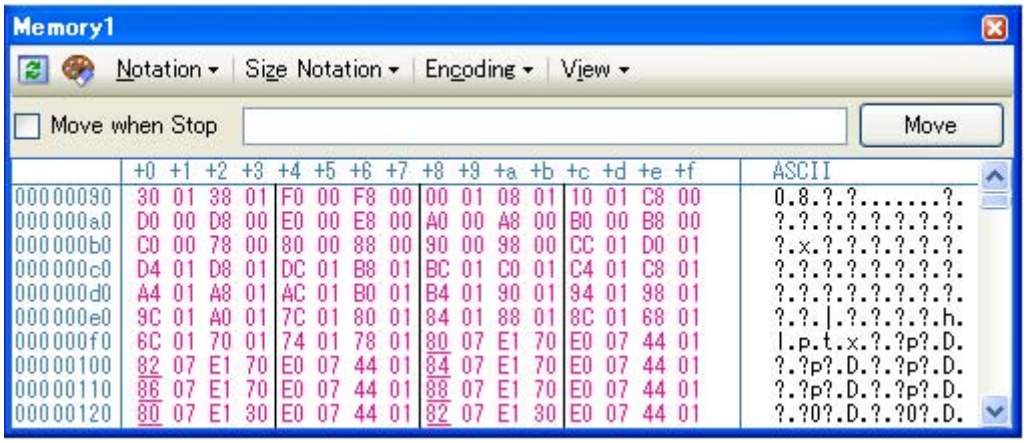
Note Impossible during tracer/timer execution

To enable the DMM function, the setting below is required in addition to the Basic Settings for Real-time Display Update Function.

Debug Tool	Category	Property	Setting Value
Full-spec emulator E1/E20	[Access memory while running]	[Access during the execution]	[Yes]
Simulator	No setting is required.		

The memory values/watch-expressions updated by the real-time display update function are highlighted in pink on the Memory panel/Watch panel.

Figure 2-84. Example of Memory Display by Real-time Display Update Function



(5) Search the memory contents

Values of memory can be searched in the Memory Search dialog box that is opened by selecting [Find...] from the context menu. The search is operated either in the memory value area or character strings area, in which the caret exists. In this dialog box, follow the steps below.

Figure 2-85. Search Memory Contents (Memory Search Dialog Box)

Cautions 1. The contents of the memory cannot be searched during execution of a program.
2. Character strings displayed as floating-point values cannot be searched.

(a) Specify [Search Data]

Specify data to search.
You can either type a value directly into the text box (up to 256 bytes), or select one from the input history via the drop-down list (up to 10 items).
If the search is performed in the memory value area, the value must be entered in the same display format (notation and size) as that area.
If the search is performed in the character strings area, then the target of the search must be a string. The specified string is converted into the encoding format displayed in that area, and searched for.
If a memory value was selected immediately prior to opening this dialog box, then that value will appear as default.

(b) Specify [Search Range]

Select the range to search from the following drop-down list.

Specify address range	Searches in the address range specified in the [Address] area.
Memory mapping	Searches within the selected memory mapping range. This list item displays individual memory mapping configured in the Memory Mapping dialog box. Display format: <*memory type*> <*address range*> <*size*>

(c) Specify [Address]

This item is only enabled if [Specify address range] is selected in the (b) Specify [Search Range].
Specify the range of memory address to search via the start and end addresses. You can either type address expressions directly into the text boxes (up to 1024 characters), or select them from the input history via the drop-down list (up to 10 items).
The results of calculating the address expressions you have entered are treated as start and end addresses, respectively.
Note, however, that the largest address that can be searched is the maximum address of the program space (0x03FFFFFF) (the mirror area cannot be searched).
An address value greater than the value expressed within 32 bits cannot be specified.

Remarks 1. A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").
2. If the start address field is left blank, it is treated as if "0x0" were specified.
3. If the end address field is left blank, then it is treated as if the maximum value in the microcontroller's address space were specified.

(d) Click the [Search Backward]/[Search Forward] button

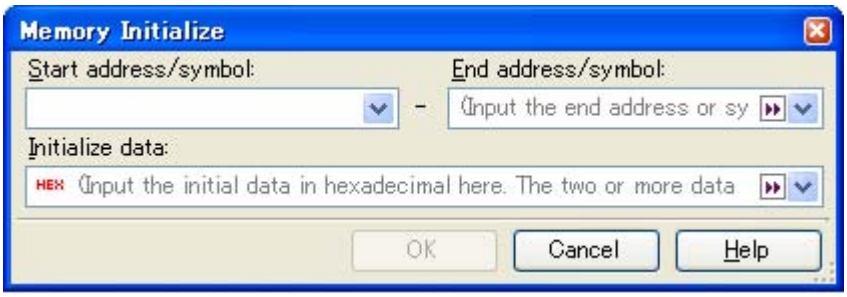
When the [Search Backward] button is clicked, search will start in the order from the large address number to small and the search results are displayed selected in the Memory panel.
When the [Search Forward] button is clicked, search will start in the order from the small address number to small and the search results are displayed selected in the Memory panel.

(6) Modify the memory contents in batch (initialize)

Contents of the memory can be modified in batch (initialize).

When [Fill...] from the context menu is selected, the Memory Initialize dialog box opens to modify the memory value of the specified address range in batch.
In this dialog box, follow the steps below.

Figure 2-86. Modify Memory Contents in Batch (Memory Initialize Dialog Box)



(a) Specify [Start address/symbol] and [End address/symbol]

Specify the range of memory address to initialize via the [Start address/symbol] and [End address/symbol].
You can either type address expressions directly into the text boxes (up to 1024 characters), or select them from the input history via the drop-down list (up to 10 items).
The results of calculating the address expressions you have entered are treated as start and end addresses, respectively.
Note that address values greater than the microcontroller address space cannot be specified.

Caution You cannot specify the range of address aligned across the different endian area.

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").

(b) Specify [Initialize data]

Specify the initializing data to write to the memory.
You can either type the initial value into the text box directly in hexadecimal number, or select one from the input history via the drop-down list (up to10 items).
You can specify more than one initial value. Specify up to 16 values of up to 4 bytes (8 characters) each, separated by spaces.
Each initial value is parsed from the end of the string, with each two characters interpreted as a byte.
If the string has an odd number of characters, then the first character is interpreted as one byte.
Note that if a initial value consists of more than one byte, then the target memory is overwritten with the value converted into an array of bytes of the specified address range's endian, as follows:

Input Character Strings (Initial Value)	How Data is Overwritten (in Bytes)	
	Little Endian	Big Endian
1	01	01
0 12	00 12	00 12
00 012 345	00 12 00 45 03	00 00 12 03 45
000 12 000345	00 00 12 45 03 00	00 00 12 00 03 45

(c) Click the [OK] button

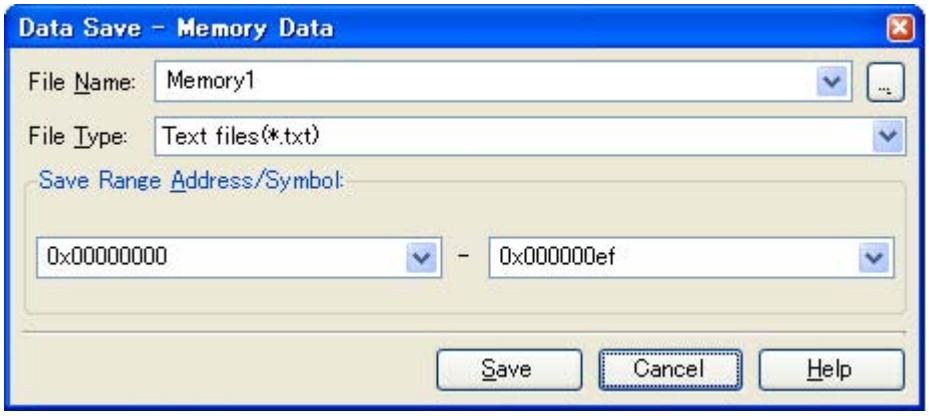
Click the [OK] button.

The memory area in the specified address range is repeatedly overwritten with the specified initial data pattern. If the end address is reached in the middle of the pattern, then writing ends at that point.
Note that if an illegal value is specified, a message will appear, and the memory value will not be initialized.

(7) Save the memory contents

Contents of the memory can be saved with range selection in text files (*.txt)/CSV files (*.csv).
When saving to the file, the latest information is acquired from the debug tool, and it is saved in accordance with the display format on this panel.
The Data Save dialog box can be opened by selecting the [File] menu >> [Save Memory Data As...] (when this operation is taken place with range selection on the panel, the memory data only in the selected range is saved).
In this dialog box, follow the steps below.

Figure 2-87. Save Memory Data (Data Save Dialog Box)



(a) Specify [File Name]

Specify the name of the file to save.
You can either type a filename directly into the text box (up to 259 characters), or select one from the input history via the drop-down list (up to 10 items). You can also specify the file by clicking the [...] button, and selecting a file via the Select Data Save File dialog box.

(b) Specify [File Type]

Select the format in which to save the file from the following drop-down list.
The following file formats can be selected.

List Item	Format
Text files (*.txt)	Text format (default)
CSV (Comma-Separated Variables)(*.csv)	CSV format[Note]

Note The data is saved with entries separated by commas (,).
If the data contains commas, each entry is surrounded by double quotes "" in order to avoid illegal formatting.

(c) Specify [Save Range Address/Symbol]

Specify the range of addresses to save via "start address" and "end addresses".
Directly enter hexadecimal number/address expression in each text box or select from the input history displayed in the drop-down list (up to 10 items).

If a range is selected in the panel, that range is specified as the default. If there is no selection, then the range currently visible in the panel is specified.

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").

(d) Click the [Save] button

Saves the memory data to a file with the specified filename, in the specified format.

Figure 2-88. Output Example of Memory Data

[Text files (*.txt)]

(Hexadecimal notation/8-bit width/ASCII code)

```
           +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f
00000000 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
00000010 | 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 |
```

[CSV files (*.csv)]

(Hexadecimal notation/8-bit width/ASCII code)

```
00000000,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00000010,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,
```

Remark When the contents of the panel are overwritten by selecting the [File] menu>> [Save Memory Data], each Memory panel (Memory1-4) is treated as a different panel.
In addition, saving range is same as the previously specified address range.

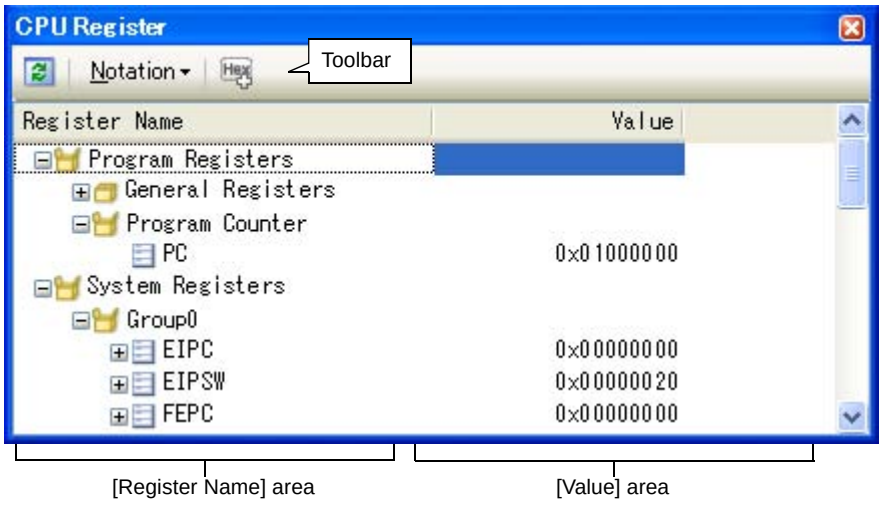
2.10.2 Display/change the CPU register

The contents of the CPU register (program registers/system registers) can be shown and the value can be changed in the CPU Register panel below.

Select the [View] menu >> [CPU Register].

For details on the contents and function in each area, see the section for the CPU Register panel.

Figure 2-89. Display the Contents of CPU Register (CPU Register Panel)



This section describes the following.

(1) Change display format of values

The display format of the [value] area can be changed using buttons below on the toolbar.

Notation	The following buttons to change the notation of a data value are displayed.
auto	Displays the value of the selected item (including sub-items) in the default notation (default).
Hex	Displays the value of the selected item (including sub-items) in hexadecimal number.
S Dec	Displays the value of the selected item (including sub-items) in signed decimal number.
U Dec	Displays the value of the selected item (including sub-items) in unsigned decimal number.
Oct	Displays the value of the selected item (including sub-items) in octal number.
Bin	Displays the value of the selected item (including sub-items) in binary number.
Asc	Displays the character strings of the selected item (including sub-items) in ASCII code. If the character size is 2 bytes and above, it is displayed with the characters for each 1 byte arranged side-by-side.
Flo	Displays the value of the selected item in float. Note that when the value is not 4-byte data, displays it in the default notation.
Dbl	Displays the value of the selected item in double. Note that when the value is not 8-byte data, displays it in the default notation.
Hex	Adds the value in hexadecimal number enclosing with "()" at the end of the value.

(2) Modify the CPU register contents

The CPU register values can be edited.

Select the value of the CPU register to edit in the [Value] area, then click on it again to switch the value to edit mode (press the [Esc] key to cancel the edit mode).

To write the edited value to the target memory, directly enter the value from the keyboard then press the [Enter] key.

Caution This operation cannot be performed during program execution.

(3) Display/modify the CPU register contents during program execution

By registering a CPU register to the Watch panel as a watch-expression, the value of the CPU register can be displayed/modified not only while the program is stopped, but in execution.

See "2.10.6 Display/change watch-expressions" for details on the watch-expression.

(4) Save the CPU register contents

The Save As dialog box can be opened by selecting the [File] menu >> [Save CPU Register Data As...], and all the contents in the CPU register can be saved to a text file (*.txt) or CSV file (*.csv).

When saving to files, retrieve the latest information from the debug tool.

Figure 2-90. Output Example of CPU Register Data

```
Register name            Value
------------------------------------------------------------------------
Category name
 -Register name          Value
      :                    :
```

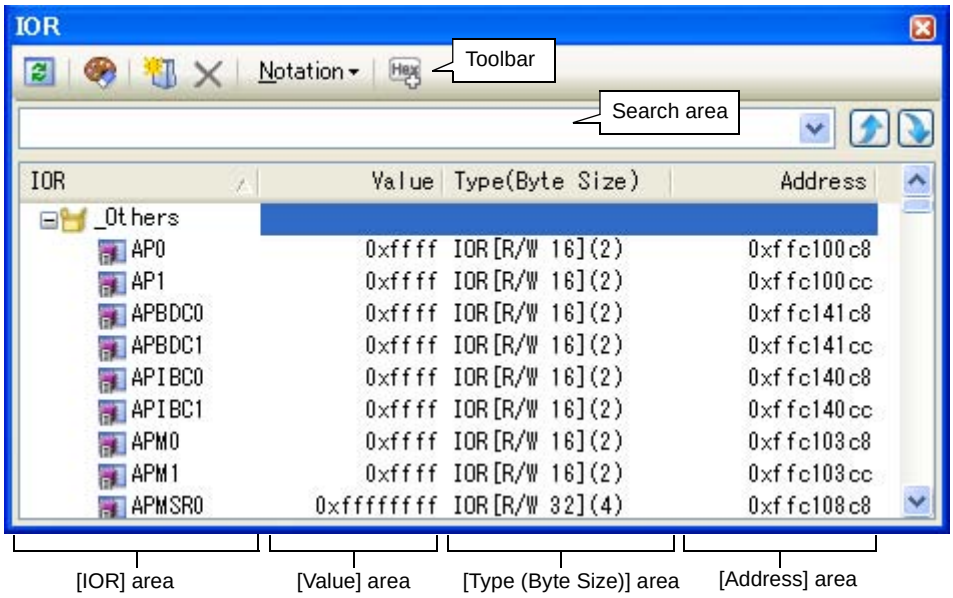
2.10.3 Display/change the I/O register

Contents of the I/O register can be displayed and its values can be changed in the IOR panel below.

Select the [View] menu >> [IOR].

For details on the contents and function in each area, see the section for the IOR panel.

Figure 2-91. Display the Contents of I/O register (IOR Panel)



This section describes the following.

(1) Search for an I/O register
(2) Organize I/O registers
(3) Change display format of values
(4) Modify the I/O register contents
(5) Display/modify the I/O register contents during program execution
(6) Save the I/O register contents

(1) Search for an I/O register

An I/O register can be searched for.

Specify the I/O register name to search with the text box in the search area (case-insensitive). You can either type character strings directly from the key board (up to 512 characters), or select one from the input history via the drop-down list (up to 10 items). Then, click either one of the following button.

	Searches up for the I/O register name containing the string specified in the text box, and selects the I/O register that is found.

	Searches down for the I/O register name containing the string specified in the text box, and selects the I/O register that is found.

Remarks 1. The hidden I/O register name being classified with a category can be searched (the category is opened and the I/O register is selected).

2. After typing character strings to search, to press the [Enter] key is the same function as clicking the  button, and to press the [Shift] + [Enter] key is the same function as clicking the  button.

(2) Organize I/O registers

The each I/O register can be categorized (by folders) and displayed in the tree view.

Cautions 1. Categories cannot be created within categories.

2. I/O registers cannot be added or deleted.

(a) Create new category

Move the caret to the I/O register name to create a new category then click the  button in the toolbar and directly enter the new category name.

(b) Edit category name

Click the category name to edit, and click it again, then directly modify the category name from the keyboard.

(c) Delete categories

Select categories to delete then click the  button in the toolbar.

However, the categories that can be deleted are only the empty categories.

(d) Change the display order

I/O register name is categorized when I/O register is dragged and dropped in the category.

Also, the display order of the categories and the I/O register names (upper or lower position) can be changed easily by drag and drop operation.

(3) Change display format of values

The display format of the [value] area can be changed using buttons below on the toolbar.

Notation	The following buttons to change the notation of a data value are displayed.
	Displays the value of the selected item in hexadecimal number (default).
	Displays the value of the selected item in signed decimal number.
	Displays the value of the selected item in unsigned decimal number.
	Displays the value of the selected item in octal number.
	Displays the value of the selected item in binary number.
	Displays the value of the selected item in ASCII code.
	Adds the value in hexadecimal number enclosing with "()" at the end of the value of the selected item.

(4) Modify the I/O register contents

The I/O register values can be edited.

Select the value of the I/O register to edit in the [Value] area, then click on it again to switch the value to edit mode (press the [Esc] key to cancel the edit mode).

To write the edited value to the target memory, directly enter the value from the keyboard then press the [Enter] key.

Cautions 1. This operation cannot be performed during program execution.

2. The value of the read-only I/O register cannot be edited.

Remarks 1. If a number with fewer digits than the size of the I/O register is entered, the higher-order digits will be padded with zeroes.

2. If a number with more digits than the size of the I/O register is entered, the higher-order digits will be masked.

3. ASCII characters can be entered to the I/O register value.

- When the numeric "0x41" is written to the I/O register "OSTM*nXX*"
  >> "0x41" is written in the port "OSTM*nXX*".
- When the ASCII character "'A'" is written to the I/O register "OSTM*nXX*"
  >> "0x41" is written in the port "OSTM*nXX*".

(5) Display/modify the I/O register contents during program execution

By registering an I/O register to the Watch panel as a watch-expression, the value of the I/O register can be displayed/modified not only while the program is stopped, but in execution.

See "2.10.6 Display/change watch-expressions" for details on the watch-expression.

(6) Save the I/O register contents

The Save As dialog box can be opened by selecting the [File] menu >> [Save IOR Data As...], and all the contents of the I/O register can be saved in a text file (*.txt) or CSV file (*.csv). At this time, the values of all I/O registers become targets irrespective of the setting of display/non-display on this panel.

When saving the contents to the file, the values of the I/O register are reacquired and save the latest values acquired.

Note that the values of read-protected I/O register are not re-read. If you want to save the latest values of those, select [Force Read Value] from the context menu then save the file.

Figure 2-92. Output Example of I/O register

```
IOR name          Value       Type (Byte Size)     Address
-------------------------------------------------------------
Category name
 -IOR name        Value       Type (Byte Size)     Address
    :               :               :                 :
```

2.10.4 Display/change global variables/static variables

Global variables or static variables are displayed and its values can be changed in the Watch panel.

Register the variables to display/modify their values to the Watch panel as the watch-expressions.

For details, see "2.10.6 Display/change watch-expressions".

2.10.5 Display/change local variables

Contents of local variables can be displayed and its values can be changed in the Local Variables panel below.

Select the [View] menu >> [Local Variable].

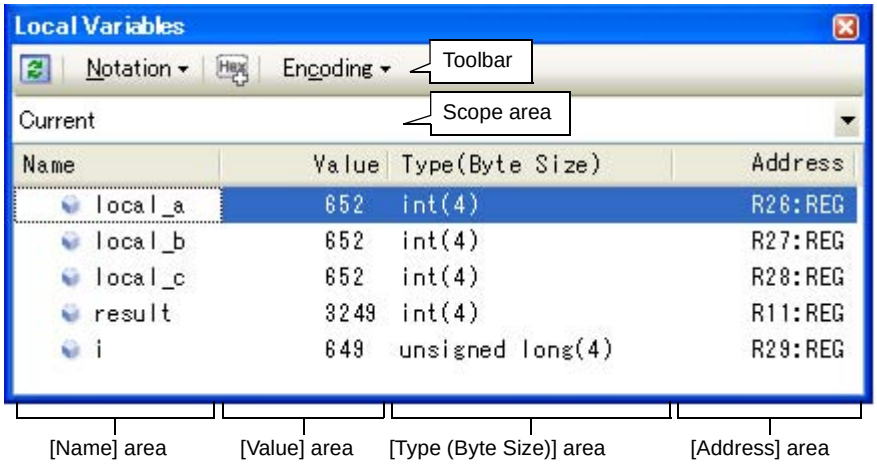
Specify the scope in the scope area to display the contents of the target local variable.

In the Local Variables panel, the name of local variables and functions are displayed. The argument of the function is also displayed as the local variable.

For details on the contents and function in each area, see the section for the Local Variables panel.

Caution Nothing is displayed on this panel during execution of a program.
When the program is stopped, items in each area are displayed.

Figure 2-93. Display the Contents of Local Variables (Local Variables Panel)



This section describes the following.

(1) Change display format of values
(2) Modify the contents of local variables
(3) Save the contents of local variables

(1) Change display format of values

The display format of the [value] area can be changed using buttons below on the toolbar.

Notation	The following buttons to change the notation of a data value are displayed.
auto	Displays values on this panel in the default notation according to the type of variable (default).
Hex	Displays values on this panel in hexadecimal number.
Dec	Displays values on this panel in decimal number.
Oct	Displays values on this panel in octal number.
Bin	Displays values on this panel in binary number.
Dec	Displays array indexes on this panel in decimal number (default).
Hex	Displays array indexes on this panel in hexadecimal number.
Flo	Displays values on this panel in float. Note that when the value is not 4-byte data, or has the type information, displays it in the default notation.
Dbl	Displays values on this panel in double. Note that when the value is not 4-byte data, or has the type information, displays it in the default notation.
Hex	Adds the value in hexadecimal number enclosing with "()" at the end of the value.
Encoding	The following buttons to change the encoding of character variables are displayed.
Asc	Displays character variables in ASCII code (default).
SJIS	Displays character variables in Shift-JIS code.
EUC	Displays character variables in EUC-JP code.
UTF 8	Displays character variables in UTF-8 code.
UTF 16	Displays character variables in UTF-16 code.

(2) Modify the contents of local variables

Values and arguments of local variables can be edited.

Select the value of the local variables/arguments to edit in the [Value] area, then click on it again to switch the value to edit mode (press the [Esc] key to cancel the edit mode).

To write the edited value to the target memory, directly enter the value from the keyboard then press the [Enter] key. At this time, the edited value is checked and if it is incompatible with the type, the editing is invalidated.

Caution This operation cannot be performed during program execution.

Remarks 1. If a number with fewer digits than the size of the variable is entered, the higher-order digits will be padded with zeroes.

2. If a number with more digits than the size of the variable is entered, the higher-order digits will be masked.

3. If the display format of a character array (type char or unsigned char) is set to ASCII, then the value can also be entered as a string (ASCII/Shift-JIS/EUC-JP/Unicode (UTF-8/UTF-16)).

4. ASCII characters can be entered to values of local variables.

- Entering via an ASCII character
  In the [Value] area for the variable "ch", enter "'A'"
  >> "0x41" will be written to the memory area allocated to "ch"
- Entering via a numeric value
  In the [Value] area for the variable "ch", enter "0x41"
  >> "0x41" will be written to the memory area allocated to "ch"
- Entering via an ASCII string
  Set the display format of character array "str" to ASCII, and in the [Value] area, enter ""ABC""
  >> "0x41, 0x42, 0x43, 0x00" will be written to the memory area allocated to "str"

(3) Save the contents of local variables

The Save As dialog box can be opened by selecting the [File] menu >> [Save Local Variables Data As...], and all the contents in the local variables can be saved in a text file (*.txt) or CSV file (*.csv).

When saving to files, retrieve the latest information from the debug tool. If arrays, pointer type variables, structures/unions, and CPU registers (only those with the part name) are displayed expanded, the value of each expanded element is also saved. When they are not expanded, "+" mark is added on the top of the item and the value becomes blank.

Figure 2-94. Output Example of Local Variables

```
Scope : Current scope
[V]Variable  [P]Parameter  [F]Function
Name                       Value   Type (Byte Size)   Address
-----------------------------------------------------------
[V]Variable name[1]        Value   Type               Address
  - [V]Variable name[0]    Value   Type               Address
          :                  :      :                    :
```

2.10.6 Display/change watch-expressions

By registering C language variables, CPU register, I/O register, and assembler symbols to the Watch panel as watch-expressions, you can always retrieve their values from the debug tool and monitor the values in batch.

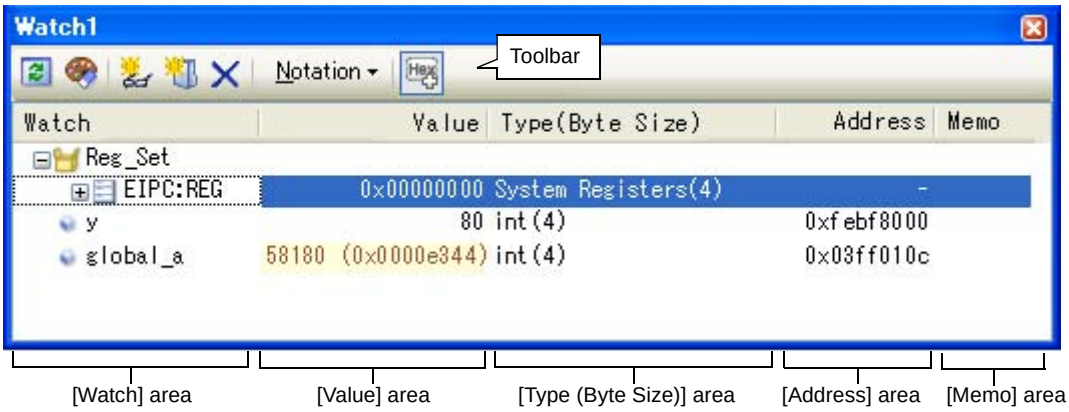
The values of watch-expressions can be updated during the program is in execution (see "(7) Display/modify the contents of watch-expressions during program execution").

Select the [View] menu >> [Watch] >> [Watch*1* - *4*] to open the Watch panel.

The Watch panel can be opened up to 4 panels. Each panel is identified by the names "Watch1", "Watch2", "Watch3" and "Watch4" on the titlebar, and the watch-expressions can be registered/deleted/moved individually, and they are saved as the user information of the project.

For details on the contents and function in each area, see the section for the Watch panel.

Figure 2-95. Display the Contents of Watch-Expression (Watch Panel)



This section describes the following.

(1) Register a watch-expression
(2) Organize the registered watch-expressions
(3) Edit the registered watch-expressions
(4) Delete a watch-expression
(5) Change display format of values
(6) Modify the contents of watch-expressions
(7) Display/modify the contents of watch-expressions during program execution
(8) Export/import watch-expressions
(9) Save the contents of watch-expressions

(1) Register a watch-expression

There are three ways as follows to register watch-expressions (watch-expressions are not registered as default).

Cautions 1. Watch-expressions can be registered up to 3000 in one watch panel (if this restriction is violated, a message appears).

2. Due to compiler optimization, the data for the target variable may not be on the stack or in a register in blocks where that variable is not used. In such cases, if the variable is registered as a watch-expression, then the value will be displayed as a question mark "?".

Remarks 1. Each watch-expression registered in each watch panel (Watch1 to Watch4) is managed in each panel and saved as the user information of the project.

2. More than one watch-expression with the same name can be registered.

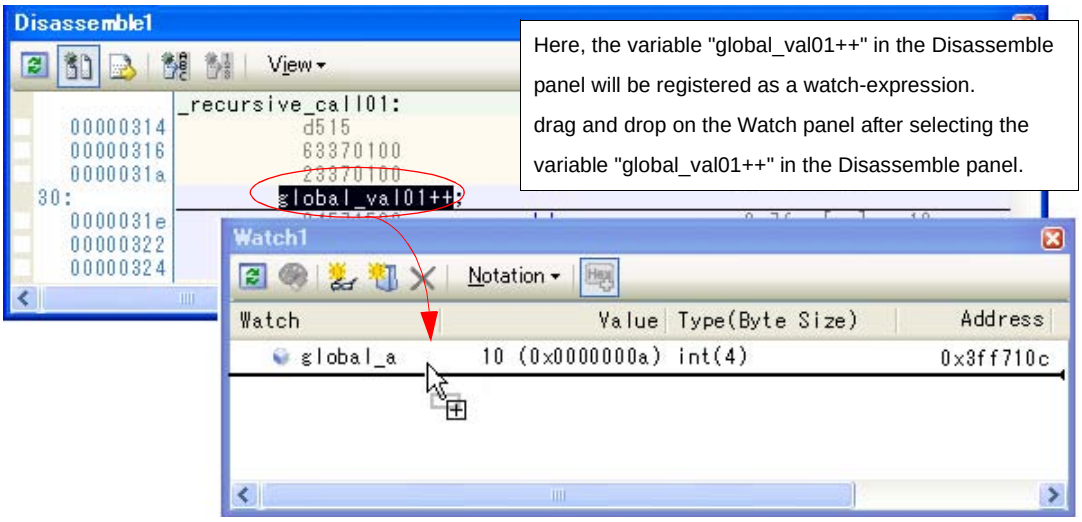
(a) Register from other panels

Watch-expressions can be registered from other panel in CubeSuite+.

In other panel, drag and drop the watch-expression to register in any watch panel (Watch1 to Watch4).

For the relationship between panels that can use this operation and targets that can be registered as watch-expressions, see "Table A-2. Relationship between Panels and Targets That Can be Registered as Watch-Expressions".

Figure 2-96. Registering Watch-Expressions from Other Panels

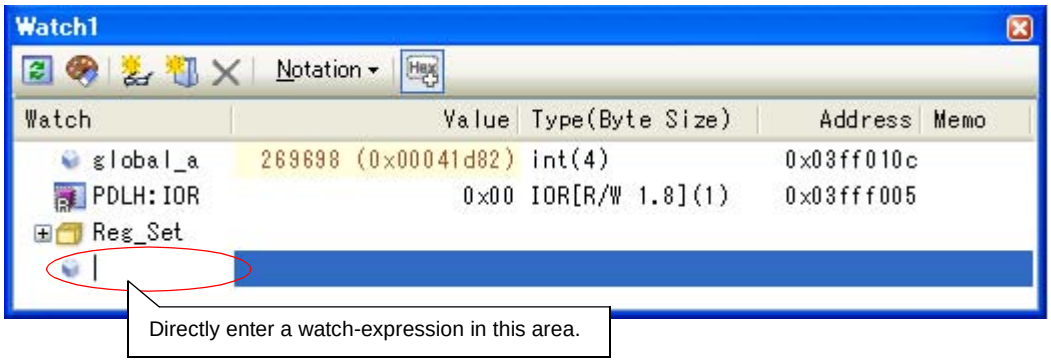


Remark You can also add a watch-expression by doing the following. First, select the target for which you wish to register a watch-expression, or move the caret to one of the target strings (the target is determined automatically). Next, from the context menu, select [Register Watch1] (but this is limited to the Watch panel (Watch 1)).

(b) Directly register in the Watch panel

Click the button in the toolbar in any the Watch panel (Watch1 to Watch4) to display the following entry box in the [Watch] area.

Figure 2-97. Entry Box of Watch-Expression



Directly input a watch-expression from the keyboard in the entry box then press the [Enter] key.
For the input forms of watch-expressions entered this way, see the tables listed below.

- "Table 2-27. Basic Input Format of Watch-expressions"
- "Table A-3. Handling of a C Language Function When Registered in Watch by Specifying Scope"
- "Table A-5. Handling of a CPU Register When Registered in Watch by Specifying Scope"
- "Table A-6. Handling of an I/O Register when Registered in Watch by Specifying Scope"

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this area (see "2.18.2 Symbol name completion function").

(c) Register from other application

Select the character strings of C language variables/CPU registers/I/O register/assembler symbols from an external editor and drag and drop it in the Watch panel (Watch 1 to Watch 4).
The dropped character strings are registered as a watch-expression.

(2) Organize the registered watch-expressions

Registered watch-expressions can be organized in categories (folders) and displayed in tree view (there is no category as default).

Cautions 1. Categories cannot be created within categories.
2. Up to 1500 categories can be created in one watch panel (if this restriction is violated, a message appears).

(a) Create new category

Move the caret to the position to create a new category then click the  button in the toolbar and directly enter the new category name.

(b) Edit category name

Click the category name to edit, and click it again, then directly modify the category name from the keyboard.

(c) Delete categories

Select categories to delete then click the  button in the toolbar.

(d) Change the display order

Registered watch-expressions are categorized when they are dragged and dropped in the category.
Also, the display order of the categories and the watch-expressions (upper or lower position) can be changed easily by drag and drop operation.

Remark Drag and drop the watch-expressions/categories in other watch panel (Watch1 to Watch4) to copy them.

(3) Edit the registered watch-expressions

Registered watch-expressions can be edited.
Double-click the watch-expression to edit to switch the watch-expression to edit mode (press the [Esc] key to cancel the edit mode).
Directly edit from the keyboard and then press the [Enter] key.

(4) Delete a watch-expression

To delete watch-expressions, select the one you want to delete in the Watch panel then click the  button in the toolbar.

(5) Change display format of values

The display format of the [value] area can be changed using buttons below on the toolbar.

Notation	The following buttons to change the notation of a data value are displayed.

auto	Displays the value of the selected watch-expression in the default notation (see "Table A-7. Display Format of Watch-Expressions (Default)") according the type of variable (default).
Hex	Displays the value of the selected item in hexadecimal number.
S Dec	Displays the value of the selected item in signed decimal number.
U Dec	Displays the value of the selected item in unsigned decimal number.
Oct	Displays the value of the selected item in octal number.
Bin	Displays the value of the selected item in binary number.
Asc	Displays the value of the selected item in ASCII code.
Flo	Displays the value of the selected item in float. Note that this item becomes valid only when the selected watch-expression value is 4-byte data.
Dbl	Displays the value of the selected item in double. Note that this item becomes valid only when the selected watch-expression value is 8-byte data.
Hex	Adds the value in hexadecimal number enclosing with "()" at the end of the value of the selected item (except the item displayed in hexadecimal number).

(6) Modify the contents of watch-expressions

The values of watch-expressions can be edited.

Double-click the value of the watch-expression to edit in the [Value] area to switch the value to edit mode (press the [Esc] key to cancel the edit mode).

To write the edited value to the target memory, directly enter the value from the keyboard then press the [Enter] key.

Note that only those values that correspond one by one to variables of C language, CPU registers, I/O register or assembler symbols can be edited. In addition, read-only I/O register values cannot be edited.

This operation can be taken place while the program is in execution. See "(4) Display/modify the memory contents during program execution" for details on how to operate it.

Remarks 1. If a number with fewer digits than the size of the variable is entered, the higher-order digits will be padded with zeroes.

2. If a number with more digits than the size of the variable is entered, the higher-order digits will be masked.

3. If the display format of a character array (type char or unsigned char) is set to ASCII, then the value can also be entered as a string (ASCII/Shift-JIS/EUC-JP/Unicode (UTF-8/UTF-16)).

4. ASCII characters can be entered to values of watch-expressions.

- Entering via an ASCII character
  In the [Value] area for the variable "ch", enter "'A'"
  >> "0x41" will be written to the memory area allocated to "ch"
- Entering via a numeric value
  In the [Value] area for the variable "ch", enter "0x41"
  >> "0x41" will be written to the memory area allocated to "ch"
- Entering via an ASCII string
  Set the display format of character array "str" to ASCII, and in the [Value] area, enter ""ABC""
  >> "0x41, 0x42, 0x43, 0x00" will be written to the memory area allocated to "str"

(7) Display/modify the contents of watch-expressions during program execution

The Memory panel/Watch panel has the real-time display update function that can update/modify the display contents of the memory/watch-expression in real-time while executing the program.

Using the real-time display update function allows you to display/modify the value of memory/watch-expression not only while the program is stopped, but also in execution.
See "(4) Display/modify the memory contents during program execution" for details on how to operate it.

(8) Export/import watch-expressions

This feature is for the export of currently registered watch-expressions to a file and the importing of such files, enabling the re-registration of watch-expressions.
To do this, follow the procedure described below.

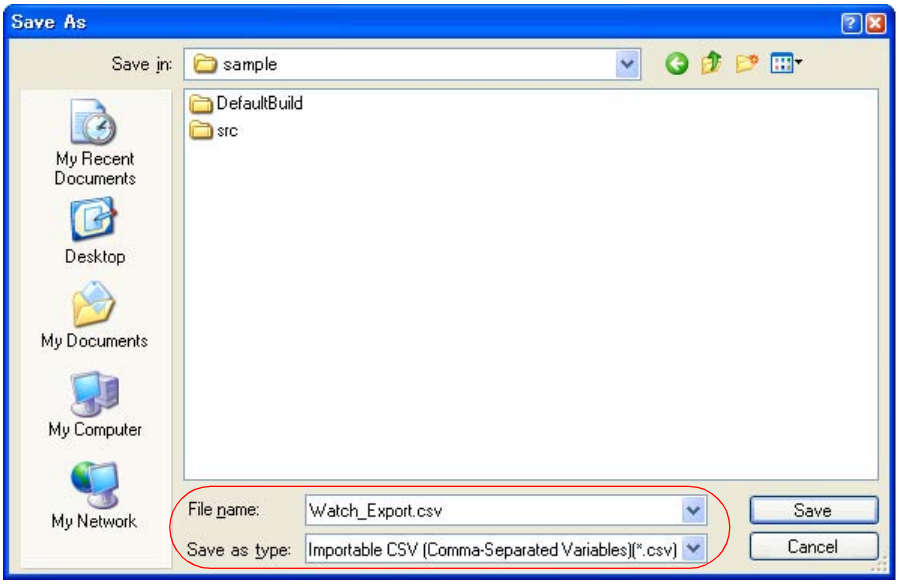
(a) Export watch-expressions

Save watch-expressions currently being registered (including categories) in a file format that is possible to import.
With the Watch panel in focus, select [Save Watch Data As...] from the [File] menu.
On the Save As dialog box that is automatically opened, specify the following items, and then click the [Save] button.

[File name]: Specify the name of a file to be saved (the file extension must be "csv").
[Save as type]: Select "Importable CSV (Comma-Separated Variables)(*.csv)"

Caution Neither values nor the type information of watch-expressions can be saved.
Items that are expanded after analyzing watch-expressions (i.e. an array, structure, and so on) cannot be saved.

Figure 2-98. Export of Watch-Expressions

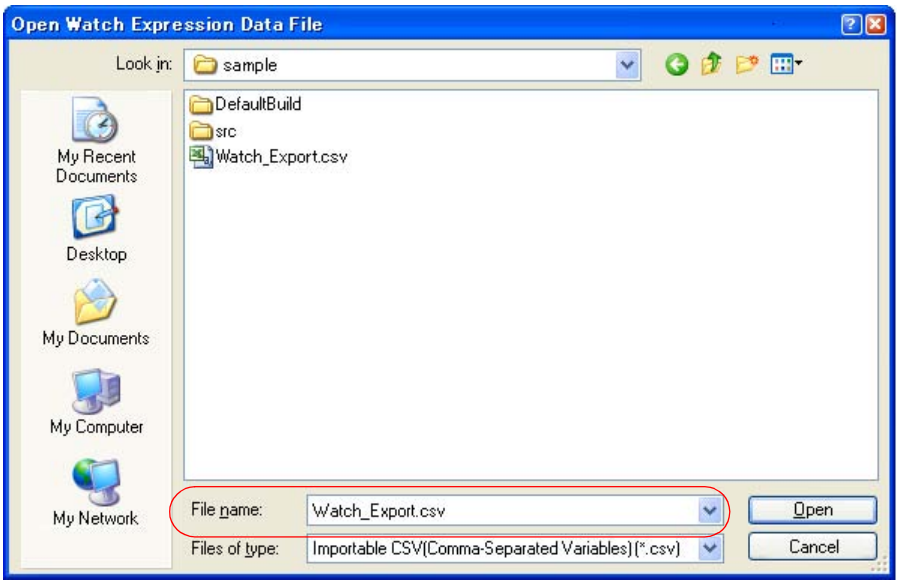


(b) Import watch-expressions

Import the file that exported in (a) to the Watch panel.
On the Watch panel to which you want to import watch-expressions, select [Import Watch Expression] from the context menu.
On the Open Watch Expression Data File dialog box that is automatically opened, specify the file that exported previously, and then click the [Open] button.

Remark If watch-expressions have been already registered, then imported watch-expressions will be registered at the bottom of them.

Figure 2-99. Import of Watch-Expressions



(9) Save the contents of watch-expressions

By selecting the [File] menu >> [Save Watch Data As...], the Save As dialog box can be opened, and all the contents of the watch-expression and its value can be saved in a text file (*.txt) or CSV file (*.csv).

When saving the contents to the file, all the values of the watch-expression are reacquired and save the latest values acquired.

If arrays, pointer type variables, structures/unions, and CPU registers (only those with the part name) are displayed expanded, the value of each expanded element is also saved. When they are not expanded, "+" mark is added on the top of the item and the value becomes blank.

Note that the values of read-protected I/O register are not re-read. If you want to save the latest values of those, select [Force Read Value] from the context menu then save the file.

Figure 2-100. Output Example of Watch Data

```
Watch-expression    Value       Type(Byte Size)     Address      Memo
-------------------------------------------------------------------------
 Watch-expression   Value       Type(Byte Size)     Address      Memo
-Category name
  Watch-expression  Value       Type(Byte Size)     Address      Memo
     :                :              :                 :           :
```

Remark When the contents of the panel are overwritten by selecting the [File] menu >> [Save Watch Data], each watch panel (Watch1 to Watch4) is treated as a different panel.

2.11 Display Information on Function Call from Stack

This section describes how to show the information on function call from the stack.

The CubeSuite+ compiler (CC-RH) pushes function-call information onto the stack, in accordance with the ANSI standard. It is thus possible to learn the function call depth, the location of the caller, parameters, and other information by analyzing the function-call information.

This "function-call information" is called the call stack information; this term will be used in the rest of this document.

Remark For "call stack information" for a microcontroller that supports multi-core, see also to "Select a Core (PE)".

2.11.1 Display call stack information

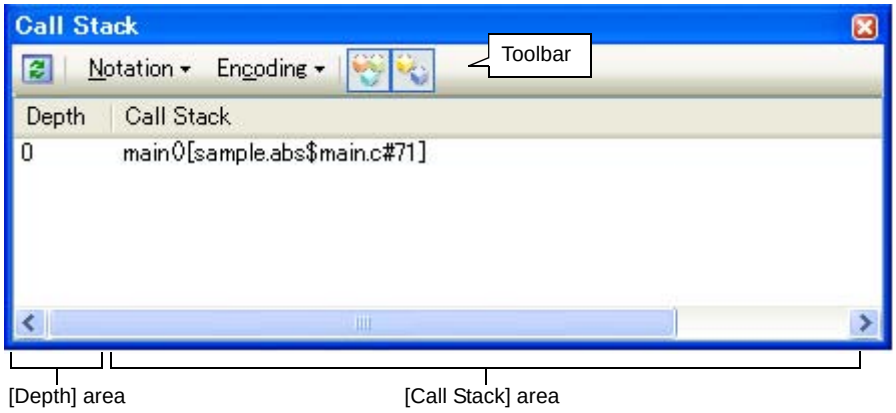
Call stack information is displayed in the Call Stack panel below.

Select the [View] menu >> [Call Stack].

For details on the contents and function in each area, see the section for the Call Stack panel.

Caution Nothing is displayed on this panel during execution of a program.
When the program is stopped, items in each area are displayed.

Figure 2-101. Display Call Stack Information (Call Stack Panel)



This section describes the following.

(1) Change display format of values
(2) Jump to the source line
(3) Display local variables
(4) Save the contents of call stack information

(1)Change display format of values

The display format of this panel can be changed using buttons below on the toolbar.

Notation	The following buttons to change the notation of a data value are displayed.
auto	Displays values on this panel in the default notation according to the type of variable (default).
Hex	Displays values on this panel in hexadecimal number.
Dec	Displays values on this panel in decimal number.
Oct	Displays values on this panel in octal number.
Bin	Displays values on this panel in binary number.

Encoding	The following buttons to change the encoding of character variables are displayed.
	Displays character variables in ASCII code (default).
	Displays character variables in Shift-JIS code.
	Displays character variables in EUC-JP code.
	Displays character variables in UTF-8 code.
	Displays character variables in UTF-16 code.

(2) Jump to the source line

Double-clicking on the line will open the Editor panel with the caret moved to the source line of the calling function indicated by the selected line (If the panel is already open, the screen will jump to the editor panel).

Remark Selecting [Jump to Disassemble] from the context menu will open the Disassemble panel (Disassemble 1) with the caret moved to address of the calling function indicated by the selected line (If the panel is already open, the screen will jump to the Disassemble panel (Disassemble 1)).

(3) Display local variables

Selecting [Jump to Local Variable at This Time] from the context menu will open the Local Variables panel that displays the local variables indicated by the currently selected line.

(4) Save the contents of call stack information

By selecting the [File] menu >> [Save Call Stack Data As...], the Save As dialog box can be opened, and all the contents in the call stack information can be saved in a text file (*.txt) or CSV file (*.csv).
When saving to files, retrieve the latest information from the debug tool.

Figure 2-102. Output Example of Call Stack Information

```
Depth      Call stack
-------------------------------------------------------------------------
 0         Call stack information
 1         Call stack information
 :               :
```

2.12 Collect Execution History of Programs

This section describes how to collect the execution history of the program.

A history of program execution is generally called a trace; this term will be used in the remainder of this document.

It is nearly impossible to find the cause of runaway program execution from the memory contents, stack information, and the like after the runaway has occurred. The collected trace data, however, can be used to trace program execution up to the runaway directly, making this an effective tool for discovering hidden bugs.

Remark For "collection of execution history of programs" for a microcontroller that supports multi-core, see also to "2.7 Select a Core (PE)".

2.12.1 Configure the trace operation

When the trace function starts, trace data which has recorded in it an execution history of the currently executed program is collected in trace memory (when program execution stops, the trace function also automatically stops).

Before the trace function can be used, it is necessary to make settings relating to the operation of a trace.

Note that the method on how to set differs depending on the debug tool used.

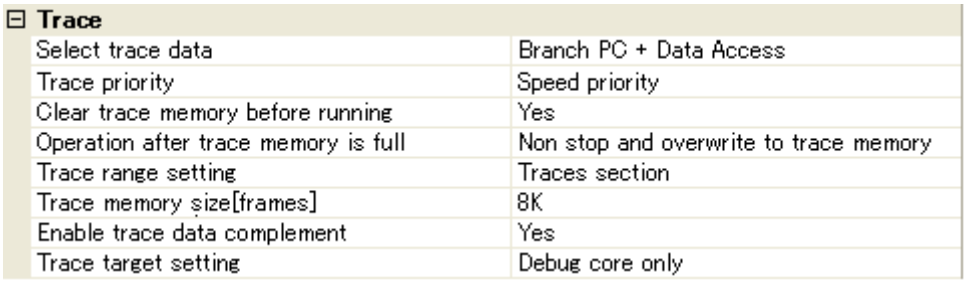
(1) [Full-spec emulator]
(2) [E1] /[E20]
(3) [Simulator]

(1) [Full-spec emulator]

This trace operation can be configured in the [Trace] category on the [Debug Tool Settings] tab in the Property panel.

Caution Properties in this category cannot be changed during program execution.

Figure 2-103. [Trace] Category [Full-spec emulator]



(a) [Select trace data]

Select the type of trace data to be collected from the following drop-down list.

Branch PC	PC values for source/destination instructions of branching during program execution are collected as trace data.
Data Access	Data information on access-related events that occurred during program execution are collected as trace data.
Branch PC + Data Access	PC values for source/destination instructions of branching during program execution, as well as data information on access-related events that occurred are collected as trace data (default).

Caution The trace memory is cleared when you change the setting of this property.

(b) [Trace priority]

Select which item should be given priority when using the trace function from the following drop-down list.

Speed priority	Traces giving priority to the real-time performance (default).
Data priority	Traces after stopping the execution pipeline of the CPU temporarily so that no data is missed.

Caution The trace memory is cleared when you change the setting of this property.

(c) [Clear trace memory before running]

Select whether to clear (initialize) the trace memory before tracing starts.

Select [Yes] to clear the memory (default).

Remark You can forcibly clear the trace memory when clicking the  button in the toolbar in the Trace panel.

(d) [Operation after trace memory is full]

Select the operation after the trace memory is full with the collected trace data from the following drop-down list.

Non stop and overwrite to trace memory	Continues overwriting the older trace data after the trace memory is full (default). When the [Clear trace memory before running] property is set to [Yes], at the time of a resumption, trace data is collected after clearing the trace memory.
Stop trace	When the trace memory is full, CubeSuite+ stops writing trace data (the program does not stop execution). Note, however, that this item is not displayed when the [Trace priority] property is set to [Data priority].
Stop	When the trace memory is full, CubeSuite+ stops writing trace data and the program stops execution.

Caution The trace memory is cleared when you change the setting of this property.

(e) [Trace range setting]

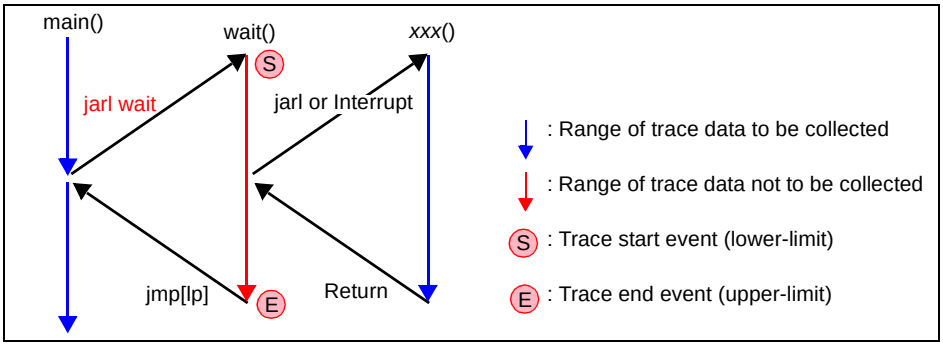
Select the range of trace data to be collected from the following drop-down list.

Note, however, that this property can be changed only when connected to the debug tool.

Traces section	Collects the execution history as trace data within the section specified with a trace start event and a trace end event (default).
Traces out of range	Collects the execution history as trace data outside the range specified with a trace start event and a trace end event.

Caution If this property is changed, then all trace start and trace end events currently being set will become invalid.

Remark When [Traces out of range] is selected, the range of trace data to be collected will be determined by a lower-limit address and an upper-limit address that are specified with a trace start event and a trace end event.



(f) [Trace memory size[frames]]

Specify from the drop-down list the size of trace memory (i.e. the number of trace frames) in this property.
The trace frame is a unit of trace data. One trace frame is used for each operation in fetch/write/read (default: [8K]).

Caution The trace memory is cleared when you change the setting of this property.

(g) [Enable trace data complement]

Select whether to enable complement display when displaying the collected trace data in the Trace panel.
By enabling complement display, instructions between branch instructions that cannot be traced by hardware can be displayed.
Select [Yes] to enable complement display (default).
This setting will be applied from the next acquisition of trace data.

(h) [Trace target setting]

Select the core to be traced from the following drop-down list.

Debug core only	Collects the trace data regarding the currently selected PE*n* (default). After collecting the trace data, the contents of the Trace panel is not changed even if you switch PE*n*.
All core	Collects the trace data for all PEs. After collecting the trace data, if you switch PE*n*, the contents of the Trace panel will be updated to the corresponding trace data.

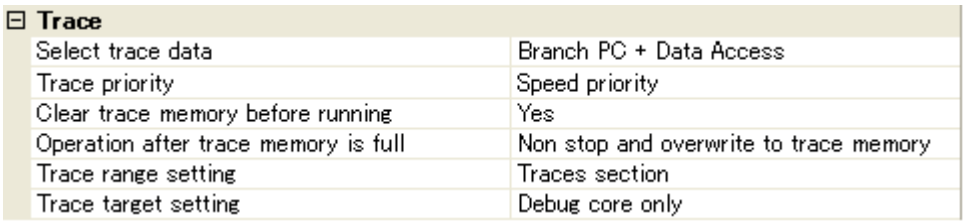
(2) [E1]/[E20]

This trace operation can be configured in the [Trace] category on the [Debug Tool Settings] tab of the Property panel.

Cautions 1. If the trace function is not mounted on the microcontroller used, all properties in this category become unchangeable after connecting to the debug tool (the trace function cannot be used).

2. Properties in this category cannot be changed during program execution.

Figure 2-104. [Trace] Category [E1][E20]



Trace	
Select trace data	Branch PC + Data Access
Trace priority	Speed priority
Clear trace memory before running	Yes
Operation after trace memory is full	Non stop and overwrite to trace memory
Trace range setting	Traces section
Trace target setting	Debug core only

(a) [Select trace data]

Select the type of trace data to be collected from the following drop-down list.

Branch PC	PC values for source/destination instructions of branching during program execution are collected as trace data.
Data Access	Data information on access-related events that occurred during program execution are collected as trace data.
Branch PC + Data Access	PC values for source/destination instructions of branching during program execution, as well as data information on access-related events that occurred are collected as trace data (default).

Caution The trace memory is cleared when you change the setting of this property.

(b) [Trace priority]

Select which item should be given priority when using the trace function from the following drop-down list.

Speed priority	Traces giving priority to the real-time performance (default).
Data priority	Traces after stopping the execution pipeline of the CPU temporarily so that no data is missed.

Caution The trace memory is cleared when you change the setting of this property.

(c) [Clear trace memory before running]

Select whether to clear (initialize) the trace memory before tracing starts.

Select [Yes] to clear the memory (default).

Remark You can forcibly clear the trace memory when clicking the  button in the toolbar in the Trace panel.

(d) [Operation after trace memory is full]

Select the operation after the trace memory is full with the collected trace data from the following drop-down list.

Non stop and overwrite to trace memory	Continues overwriting the older trace data after the trace memory is full (default). When the [Clear trace memory before running] property is set to [Yes], at the time of a resumption, trace data is collected after clearing the trace memory.
Stop trace	When the trace memory is full, CubeSuite+ stops writing trace data (the program does not stop execution).
Stop	When the trace memory is full, CubeSuite+ stops writing trace data and the program stops execution.

Caution The trace memory is cleared when you change the setting of this property.

(e) [Trace range setting]

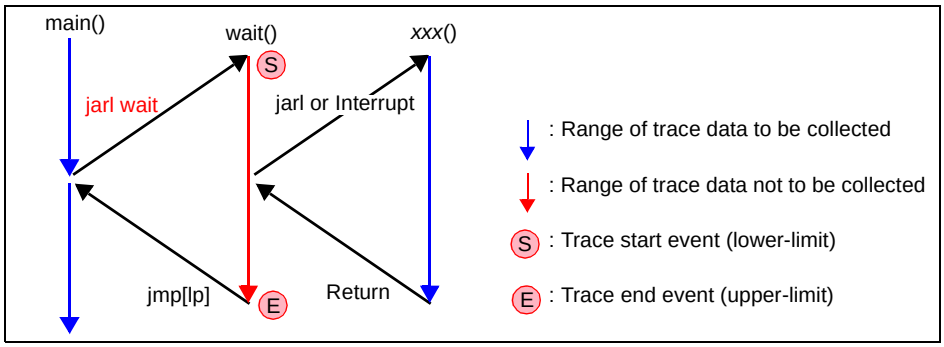
Select the range of trace data to be collected from the following drop-down list.

Note, however, that this property can be changed only when connected to the debug tool.

Traces section	Collects the execution history as trace data within the section specified with a trace start event and a trace end event (default).
Traces out of range	Collects the execution history as trace data outside the range specified with a trace start event and a trace end event.

Caution **If this property is changed, then all trace start and trace end events currently being set will become invalid.**

Remark When [Traces out of range] is selected, the range of trace data to be collected will be determined by a lower-limit address and an upper-limit address that are specified with a trace start event and a trace end event.



(f) [Trace target setting]

Select the core to be traced from the following drop-down list.

Debug core only	Collects the trace data regarding the currently selected PE*n* (default). After collecting the trace data, the contents of the Trace panel is not changed even if you switch PE*n*.
All core	Collects the trace data for all PEs. After collecting the trace data, if you switch PE*n*, the contents of the Trace panel will be updated to the corresponding trace data.

(3) [Simulator]

This trace operation can be configured in the [Trace] category on the [Debug Tool Settings] tab of the Property panel.

Figure 2-105. [Trace] Category [Simulator]

Trace	
Use trace function	**Yes**
Clear trace memory before running	Yes
Operation after trace memory is full	Non stop and overwrite to trace memory
Accumulate trace time	No
Trace memory size[frames]	4K
Rate of frequency division of trace time tag	1/1

(a) [Use trace function]

Select whether to use trace function.

Select [Yes] to use the trace function (default: [No]).

(b) [Clear trace memory before running]

Select whether to clear (initialize) the trace memory before tracing starts.

Select [Yes] to clear the memory (default).

Remark You can forcibly clear the trace memory when clicking the  button in the toolbar in the Trace panel.

(c) [Operation after trace memory is full]

Select the operation after the trace memory is full with the collected trace data from the following drop-down list.

Non stop and overwrite to trace memory	Continues overwriting the older trace data after the trace memory is full (default). When the [Clear trace memory before running] property is set to [Yes], at the time of a resumption, the trace data is collected after clearing the trace memory.
Stop trace	When the trace memory is full, CubeSuite+ stops writing trace data (the program does not stop execution).
Stop	When the trace memory is full, CubeSuite+ stops writing trace data and the program stops execution.

(d) [Accumulate trace time]

Select whether to display the trace time with accumulated time.

Select [Yes] to display trace time with accumulated time. Select [No] to display the trace time with differential time (default).

(e) [Trace memory size[frames]]

Select the trace memory size (trace frame number).

The trace frame is a unit of trace data. One trace frame is used for each operation in fetch/write/read.

Drop down list includes the following trace frame numbers.

4K (default), 8K, 12K, 16K, 20K, 24K, 28K, 32K, 36K, 40K, 44K, 48K, 52K, 56K, 60K, 64K, 128K 192K, 256K, 320K, 384K, 448K, 512K, 576K, 640K, 704K, 768K, 832K, 896K, 960K, 1M, 2M, 3M

(f) [Rate of frequency division of trace time tag]

Select the frequency division ratio of the counter to be used for time tag display (the [Time] item in the Trace panel) (default: [1/1]).

2.12.2 Collect execution history until stop of the execution

In the debug tool, there is a function to collect the execution history from the start of program execution to the stop.

Therefore, the trace data collection is automatically started when the program starts executing and stopped when the program stops.

See "2.12.6 Display the collected execution history" for how to check the collected trace data.

Remark This function is actuated by an Unconditional Trace event, one of the built-in events that are set in the debug tool by default.

Consequently, if the Unconditional Trace event is set to Invalid state by clearing the check box in the Events panel, trace data linked to the start of program execution will not be collected (the Unconditional Trace event is set to Valid state by default).

Note that Unconditional Trace event and Trace event described later (see "2.12.3 Collect execution history in the arbitrary section") are used exclusively of each other. Therefore, if Trace event with Valid state is set, Unconditional Trace event is automatically set to Invalid state.

2.12.3 Collect execution history in the arbitrary section

The execution history is collected as trace data only for the arbitrary section while the program is in execution by setting a Trace event.

This Trace event consists of a trace start event and a trace end event.

To use this function, follow the procedure described below.

(1) Set a Trace event
(2) Execute the program
(3) Delete a Trace event

Cautions 1. Also see "2.16.6 Notes for setting events" for details on Trace events (e.g. limits on the number of enabled events).

2. [Simulator]
Trace start events and trace stop events cannot be set/deleted while a tracer is running.

(1) Set a Trace event

To set a Trace event, set a trace start event and a trace end event that starts/stops collecting the trace data.

Use one of the following methods to set a trace start event and a trace end event.

(a) For execution-related events

By setting execution-related events for a trace start event and a trace end event, it is possible to start and stop the collection of trace data at any place.

Perform this operation in the Editor panel/Disassemble panel in which the source text/disassembled text is displayed.

Follow the operation listed below from the context menu, in accordance with your desired event type, after moving the caret to the target line that has a valid address.

Event Type	Operation
Trace start	Select [Trace Settings] >> [Start Tracing]
Trace end	Select [Trace Settings] >> [Stop Tracing]

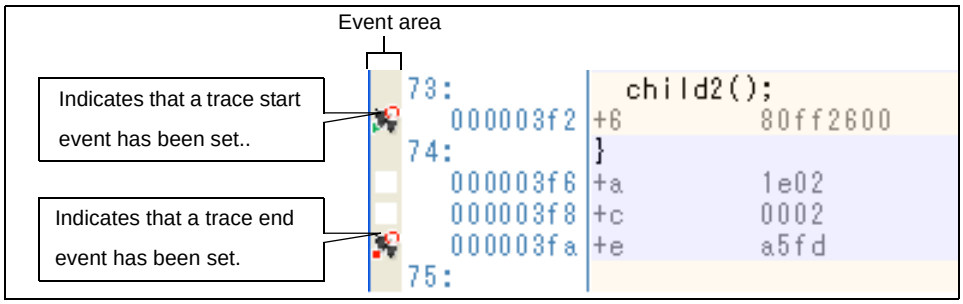
Caution [Simulator]
Simulator will not display a trace end event as the results of a trace. For this reason, set a trace end event to one line below the range that you wish to display as the trace data.

A trace start event or a trace end event is set to the instruction at the start address corresponding to the line of the caret position. Once a trace start event or a trace end event is set, the following event mark is displayed in the event area of the line/address that an event is set.

Table 2-12. Event Marks of Trace Start Event and Trace End Events

Event Type	Event Mark
Trace start	
Trace end	

Figure 2-106. Trace Start and Trace End Events Setting Example (Disassemble Panel)



Remark [Full-spec emulator][E1][E20]

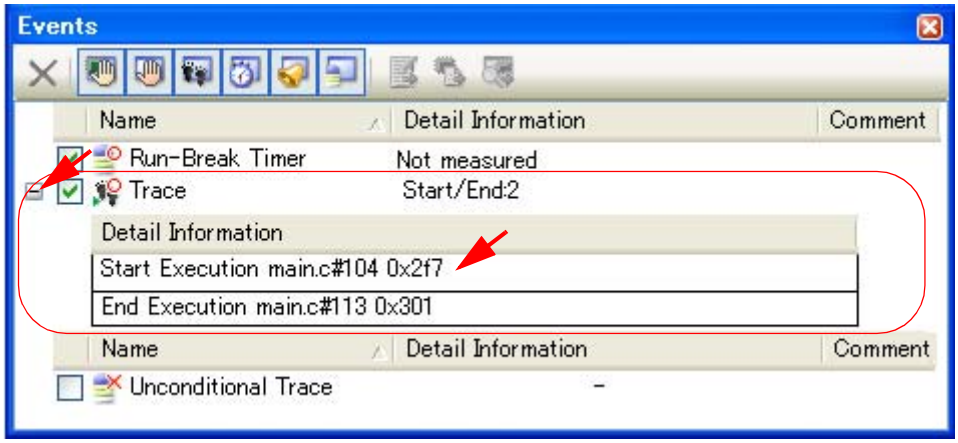
By specifying [Traces out of range] in the [Trace range setting] property in the [Trace] category on the [Debug Tool Settings] tab of the Property panel, you can collect the execution history as trace data outside the specified range.

(b) For access-related events

In this product version, this function is not supported.

When a trace start event and a trace end event are set, they are managed collectively on the Events panel as one instance of a Trace event (see "2.16 Manage Events"). When you click the "+" mark at a Trace event item, detailed information on the trace start event and the trace end event you have set is displayed.

Figure 2-107. Example of Trace Start and Trace End Events (Execution Type) in Events Panel



Remarks 1. If either one of a trace start event and a trace end event is set as Valid state, the check box of Unconditional Trace event in the Events panel is automatically cleared, therefore, trace data collection does not automatically start with the start of the program execution (the tracer will not run until the condition of the trace start event that has been set is met).

2. A trace end event is not indispensable for a Trace event.

3. Event marks differ depending on the event state (see "2.16.1 Change the state of set events (valid/invalid)").

When an event is set at the point which other event is already set, the event mark ( ) is displayed meaning more than one event is set at the point.

4. [Simulator]

If either one of a trace start event and a trace end event is set to Valid state, the [Use trace

function] property in the [Trace] category on the [Debug Tool Settings] tab of the Property panel is automatically set to [Yes] and the trace function will be enabled.

(2) Execute the program

Execute the program (see "2.8 Execute Programs").
Collection of trace data is started or finished when the condition set for a trace start event or a trace end event is met.
See "2.12.6 Display the collected execution history" for how to check the collected trace data.

(3) Delete a Trace event

To delete a Trace event you have set, on the Editor panel/Disassemble panel, right-click the event mark in the event area and select [Delete Event] from the context menu that is displayed.
Also, there is another way to delete a set event. Select the Trace event you want to delete on the Events panel, and then click the  button in the toolbar (see "2.16.4 Delete events").

Caution It is not possible to delete only a trace start event or a trace end event (i.e. if either a trace start event or a trace end event is deleted from the event marks on the event area, all of the corresponding event marks are deleted).

2.12.4 Collect execution history only when the condition is met [Simulator]

The program execution history can be collected only when a condition is met.
By setting a Point Trace event, the execution history is collected as trace data only when an arbitrary variable or I/O register is accessed with the specified type.
To use this function, follow the procedure described below.

(1) Set a Point Trace event
(2) Execute the program
(3) Delete a Point Trace event

(1) Set a Point Trace event

Use one of the following methods to set a Point Trace event.

Cautions 1. Also see "2.16.6 Notes for setting events" for details on Point Trace events (e.g. limits on the number of enabled events).
2. You cannot set/delete Point Trace events while a tracer is running.
3. Accesses via DMA cannot be traced.

Remark When a Point Trace event is set to Valid state, the [Use trace function] property in the [Trace] category on the [Debug Tool Settings] tab of the Property panel is automatically set to [Yes] and the trace function will be enabled.

(a) For access to a variable or I/O register on the Editor panel/Disassemble panel

Perform this operation in the Editor panel/Disassemble panel in which the source text/disassembled text is displayed.
Follow the operation listed below from the context menu, in accordance with your desired access type, after selecting the variable or I/O register as the subject to access.
Note, however, that only global variables, static variables inside functions, and file-internal static variables can be used.

Access Type	Operation
Read	Select [Trace Settings] >> [Record Reading Value].
Write	Select [Trace Settings] >> [Record Writing Value].
Read/Write	Select [Trace Settings] >> [Record R/W Value].

Caution Variables within the current scope can be specified.

(b) For access to a registered watch-expression

Perform this operation in the Watch panel.

Select the watch-expression as the subject to access and perform the following operation from the context menu (see "2.10.6 Display/change watch-expressions").

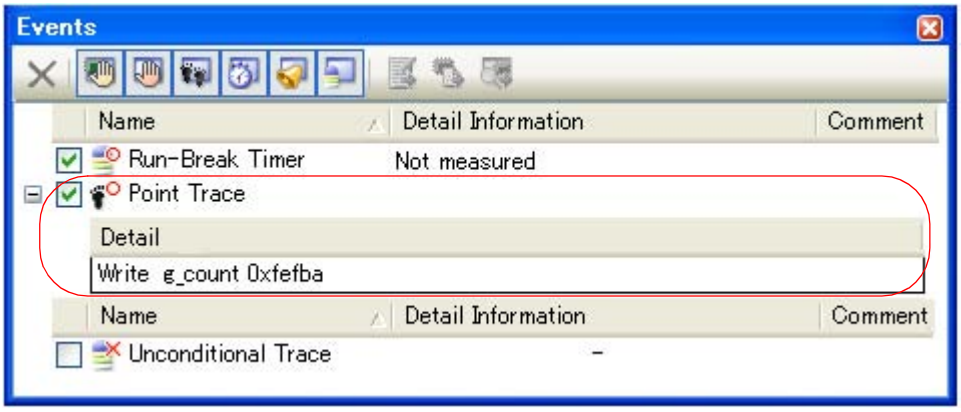
Note, however, that only global variables, static variables inside functions, file-internal static variables, and I/O register can be used.

Access Type	Operation
Read	Select [Trace Output] >> [Record Reading Value].
Write	Select [Trace Output] >> [Record Writing Value].
Read/Write	Select [Trace Output] >> [Record R/W Value].

Caution A watch-expression within the current scope can be specified.
To target a watch-expression outside the current scope, select a watch-expression with a specified scope.

By performing the above operation, it is interpreted as if a Point Trace event has been set at the target variable/I/O register/watch-expression, and it is managed in the Events panel (see "2.16 Manage Events" for details).

Figure 2-108. Example of Setting Point Trace Event in Events Panel



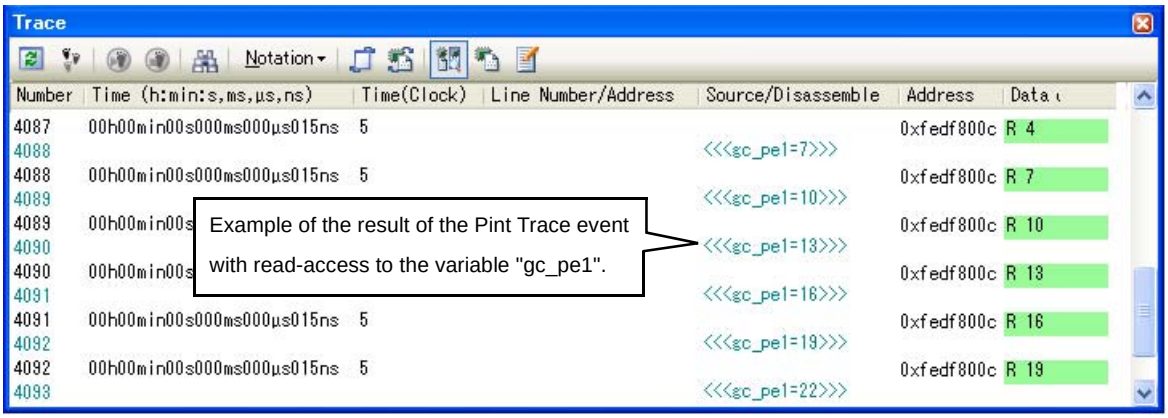
(2) Execute the program

Execute the program (see "2.8 Execute Programs").

If the conditions for a Point Trace event that you have set are met while the program is executing, that information is collected as trace data.

See "2.12.6 Display the collected execution history" for details on checking trace data.

Figure 2-109. Example of Point Trace Event Results View



(3) Delete a Point Trace event

To delete a Point Trace event you have set, select the Point Trace event you want to delete on the Events panel, and then click the button in the toolbar (see "2.16.4 Delete events").

2.12.5 Stop/restart collection of execution history

It is possible to temporarily stop or restart the collection of execution history during program execution.

(1) Stop collection of execution history temporarily
(2) Restart collection of execution history

(1) Stop collection of execution history temporarily

By clicking the button on the toolbar in the Trace panel during program execution, it is possible to temporarily stop collection of trace data without stopping program execution.
Use this function when you want to stop only the trace function without halting the program and check the trace data that has been collected until you stop it.

(2) Restart collection of execution history

If you have halted the trace function during program execution, you can start collection of trace data again by clicking the button on the toolbar in the Trace panel.
Note that the trace data that has been collected before you restart is cleared once.

2.12.6 Display the collected execution history

The collected trace data is displayed in the Trace panel below.
Select the [View] menu >> [Trace].
The trace data displays by mixing the disassembled text and source text by default, but it is also possible to display either one of these by selecting the Display mode.
For details on the contents and function in each area, see the section for the Trace panel.

Figure 2-110. Display Trace Data (Trace Panel [Full-spec emulator][E1][E20])

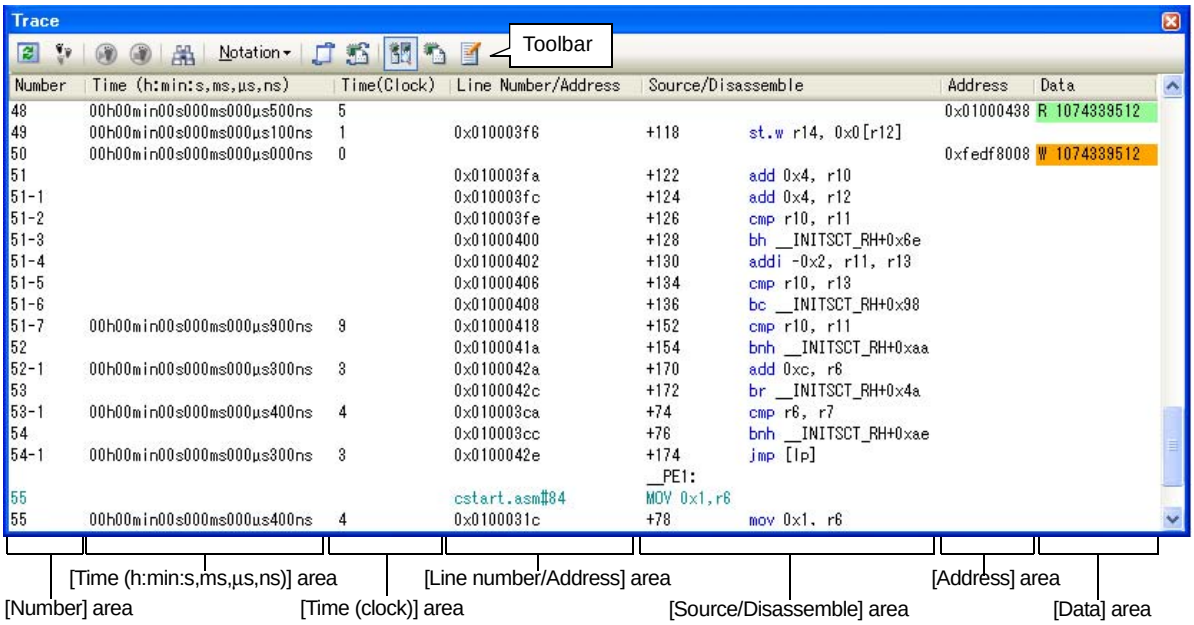
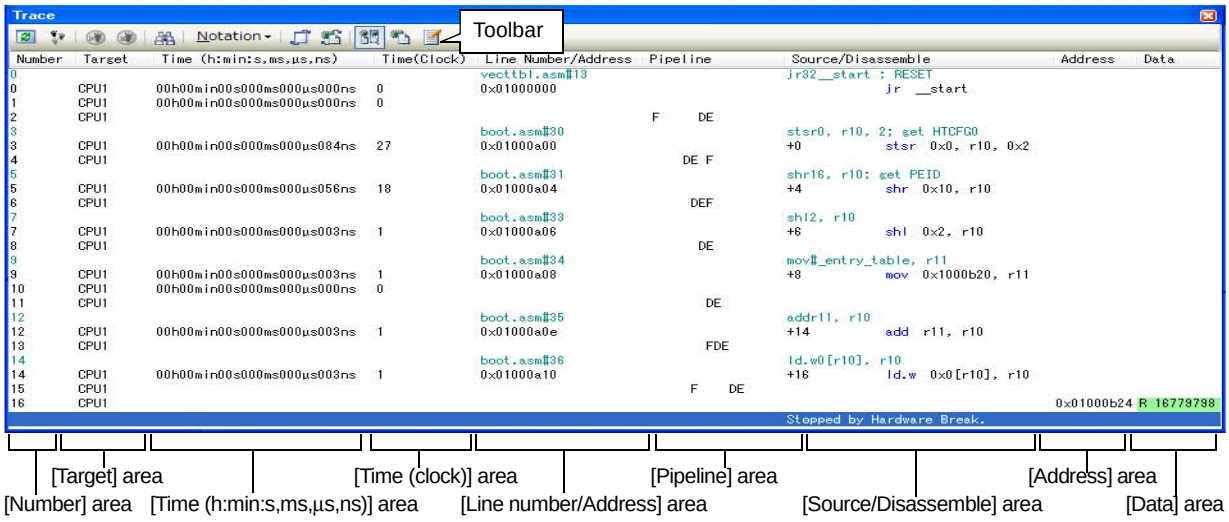


Figure 2-111. Display Trace Data (Trace Panel [Simulator])



This section describes the following.

(1) Change display mode

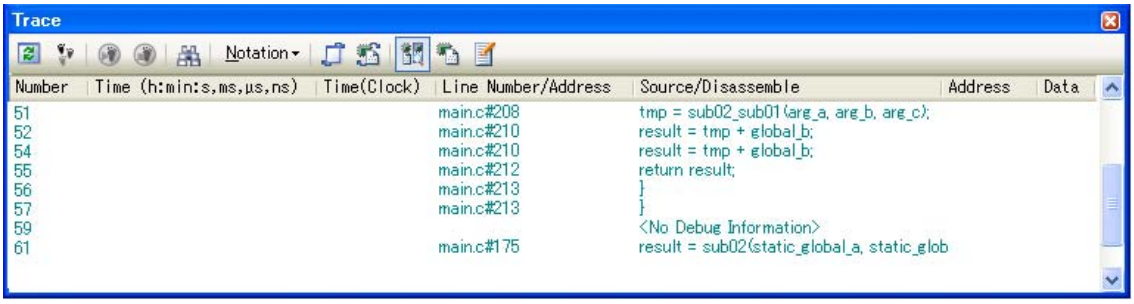
Display mode can be changed to the purpose when clicking the buttons below in the toolbar.

Note that these buttons are disabled while the tracer is running.

Table 2-13. Display Modes of Trace Panel

Button	Display Mode	Displayed Content
	Mixed display mode	Displays the instruction (disassemble results), labels, source text (corresponding source line), point trace results, and break causes (default).
	Disassemble display mode	Displays the instruction (disassemble results), labels, point trace results, and break causes.
	Source display mode	Displays the source text (corresponding source line), and break causes. However, when a place where no debugging information is present is executed, "<No Debug Information>" is displayed.

Figure 2-112. Example of Source Display Mode View (Trace Panel)



(2) Change display format of values

The display format of the [Line Number/Address], [Address] and [Data] area can be changed using buttons below on the toolbar.

Note that these buttons are disabled while the tracer is running.

Notation	The following buttons to change the notation of a data value are displayed.
	Displays values on this panel in hexadecimal number (default).
	Displays values on this panel in decimal number.
	Displays values on this panel in octal number.
	Displays values on this panel in binary number.

(3) Link with other panels

Items in the trace panel can be linked to other panels using the currently selected line address as a pointer (window focus will not move).

Click the  button on the toolbar to start linking to the Editor panel. Click the  button on the toolbar to start linking to the Disassemble panel.

If the button is clicked again, the link is disconnected.

Remark The Editor panel/Disassemble panel opens when selecting the [Jump to Source]/[Jump to Disassemble] from the context menu with moving the caret to the source line/address corresponding to the address of the currently selected line (focus is moved).

2.12.7 Clear the trace memory

To clear the collected trace data contents, click the button on the toolbar.
Note that this button is disabled while a tracer is running.

Remark When [Yes] is selected in the [Clear trace memory before running] property in the [Trace] category on the [Debug Tool Settings] tab of the Property panel, the trace memory is cleared each time a program is executed.

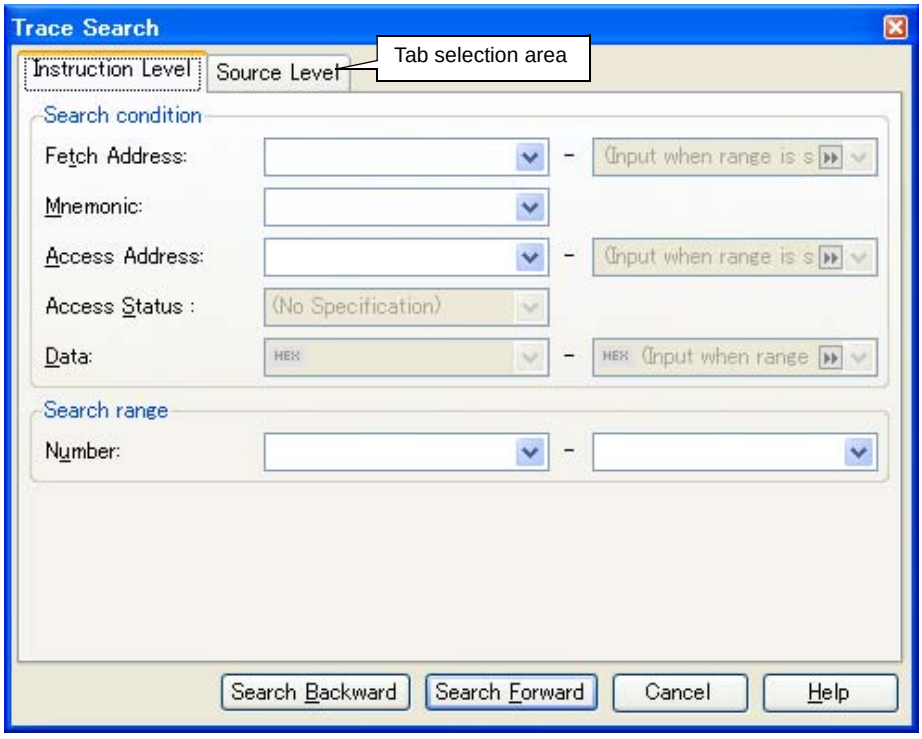
2.12.8 Search the trace data

To search for the collected trace data, click the button to open the Trace Search dialog box (note that the search is disabled during execution of a program).

In this dialog box, follow the steps below.

Note that by selecting the appropriate tab in this dialog box, you can choose to search for trace data at the instruction level or search at the source level.

Figure 2-113. Search Trace Data (Trace Search Dialog Box)



This section describes the following.

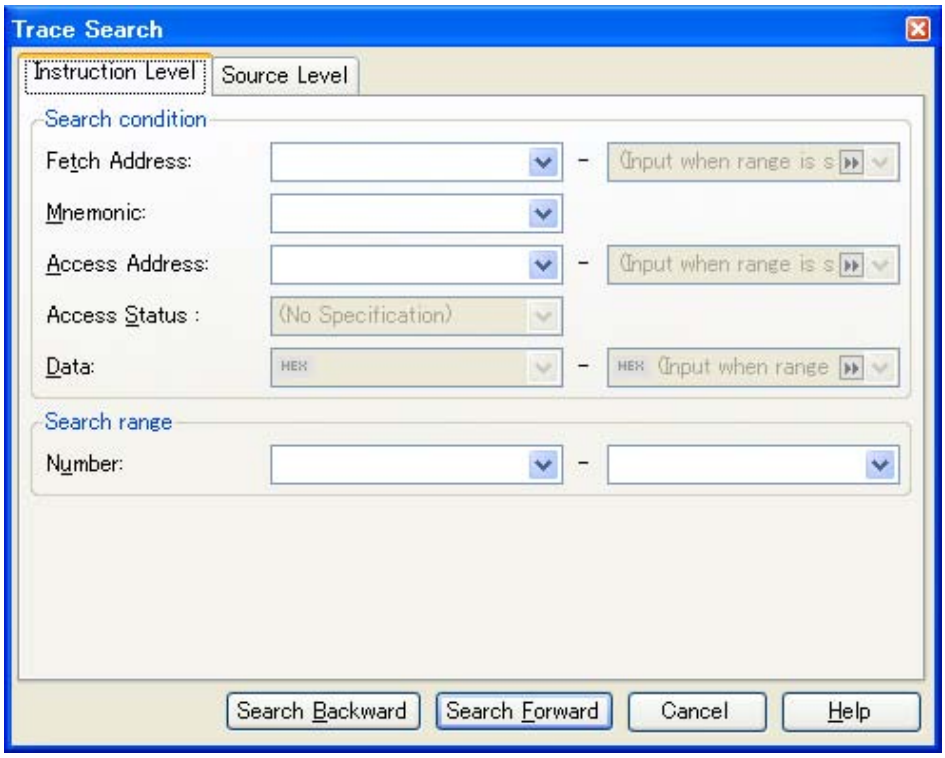
(1) Search in the instruction level
(2) Search in the source level

(1) Search in the instruction level

Search for the trace data at the instruction level.
Select the [Instruction Level] tab and then follow the steps below.

Caution When you search for the trace data at the instruction level, the display mode must be set in the Trace panel to the Mixed display mode or Disassemble display mode.

Figure 2-114. Search Trace Data in Instruction Level



(a) Specify [Fetch Address]

Specify the fetch address if it is a required search parameter.

You can either type address expressions directly into the text boxes, or select it from the input history via the drop-down list (up to 10 items).

The fetch address can also be specified as a range. In this case, specify a range by specifying address expressions in both the left and right text boxes.

If the right-hand text box is blank or contains the text [(Input when range is specified)], then the fixed address specified in the left-hand text box will be searched.

Note that if an address value greater than the microcontroller address space is specified, the upper address value is masked.

An address value greater than the value expressed within 32 bits cannot be specified.

(b) Specify [Mnemonic]

Specify the mnemonic if it is a required search parameter.

The specified character strings in this area is searched within the [Source/Disassemble] area of the Trace panel.

You can either type a mnemonic directly into the text boxes, or select one from the input history via the drop-down list (up to 10 items).

Searches are case-insensitive, and partial matches are also allowed.

(c) Specify [Access Address]

Specify the access address if it is a required search parameter.

You can either type the address value directly into the text boxes (in hexadecimal number), or select it from the input history via the drop-down list (up to 10 items).

The access address can also be specified as a range. In this case, specify a range by specifying address expressions in both the left and right text boxes.

If the right-hand text box is blank or contains the text [(Input when range is specified)], then the fixed address specified in the left-hand text box will be searched.
Note that if an address value greater than the microcontroller address space is specified, the upper address value is masked.
An address value greater than the value expressed within 32 bits cannot be specified.

(d) Specify [Access Status]

This item is only enable if a value for Specify [Access Address] is specified.
Select the access type (Read/Write, Read, Write, Vector Read and DMA) from drop-down list.
Select [(No Specification)] if you do not wish to limit access types.

(e) Specify [Data]

This item is only enable if a value for Specify [Access Address] is specified.
Specify the access data.
You can either type the data directly into the text boxes (in hexadecimal number), or select it from the input history via the drop-down list (up to 10 items).
The data can also be specified as a range. In this case, specify a range by specifying data in both the left and right text boxes.
If the right-hand text box is blank or contains the text [(Input when range is specified)], then the fixed data specified in the left-hand text box will be searched.

(f) Specify [Number]

Specify the range within the trace data to search via the number displayed in the [Number] area of the Trace panel.
Specify the starting number in the left text box, and the ending number in the right text box ("0" to "*last number*" are specified by default).
You can either type the numbers directly into the text boxes (in base-10 format), or select them from the input history via the drop-down list (up to 10 items).
If the left-hand text box is left blank, it is treated as if "0" were specified.
If the right-hand text box is left blank, it is treated as if the last number were specified.

(g) Click the [Search Backward]/[Search Forward] button

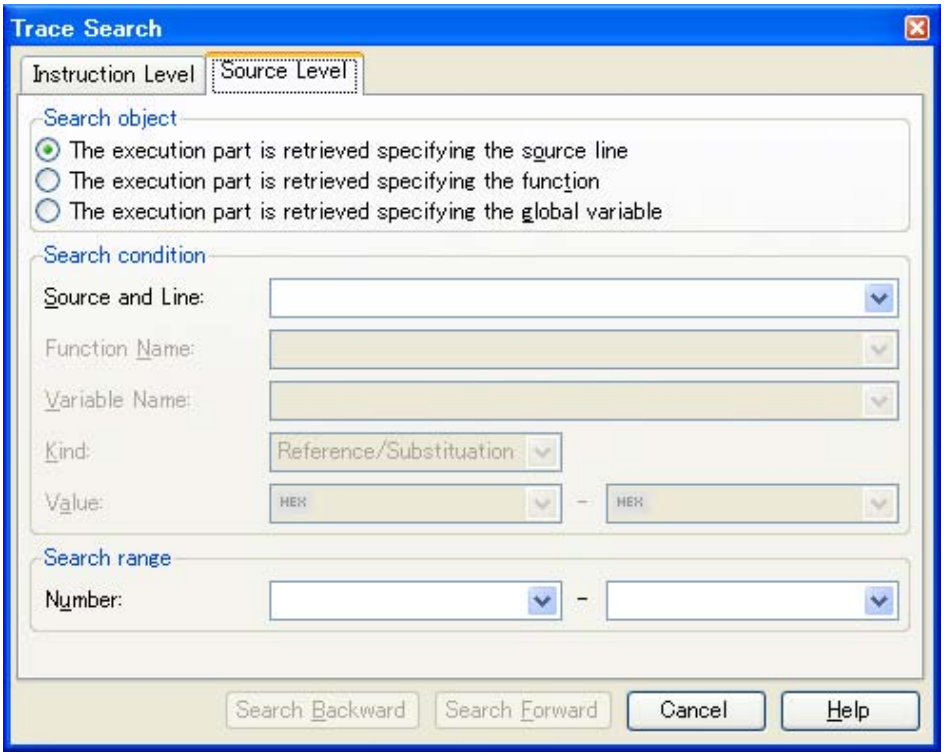
When the [Search Backward] button is clicked, search is taken place in the order from the large number to small and the search results are shown selected in the Trace panel.
When the [Search Backward] button is clicked, search is taken place in the order from the small number to large and the search results are shown selected in the Trace panel.

(2) Search in the source level

Search for the trace data at the source level.
Select the [Source Level] tab.

Caution When you search for the trace data at the source level, the display mode must be set in the Trace panel to the Mixed display mode or Source display mode.

Figure 2-115. Search Trace Data in Source Level



(a) Search with specifying the source line (default)

Select the [The execution part is retrieved specifying the source line] item in the [Search object] area and then follow the operation below.

<1> Specify [Source and Line]

The specified character strings in this area is searched within the [Line Number/Address] area of the Trace panel.

You can either type the character strings of the source line to be find directly into the text box, or select them from the input history via the drop-down list (up to 10 items).

Searches are case-insensitive, and partial matches are also allowed.

Examples 1. main.c#40
2. main.c
3. main

<2> Specify [Number]

Specify the range within the trace data to search via the number displayed in the [Number] area of the Trace panel.

Specify the starting number in the left text box, and the ending number in the right text box ("0" to *"last number*" are specified by default).

You can either type the numbers directly into the text boxes (in base-10 format), or select them from the input history via the drop-down list (up to 10 items).

If the left-hand text box is left blank, it is treated as if "0" were specified.

If the right-hand text box is left blank, it is treated as if the last number were specified.

<3> Click the [Search Backward]/[Search Forward] button

When the [Search Backward] button is clicked, search is taken place in the order from the large number to small and the search results are shown selected in the Trace panel.

When the [Search Backward] button is clicked, search is taken place in the order from the small number to large and the search results are shown selected in the Trace panel.

(b) Search with specifying the function name

Select the [The execution part is retrieved specifying the function] item in the [Search object] area and then follow the operation below.

<1> Specify [Function Name]

You can either type the function name to be find directly into the text box, or select it from the input history via the drop-down list (up to 10 items).

Searches are case-insensitive, and only complete matches are retrieved.

<2> Specify [Number]

Specify the range within the trace data to search via the number displayed in the [Number] area of the Trace panel.

Specify the starting number in the left text box, and the ending number in the right text box ("0" to *"last number*" are specified by default).

You can either type the numbers directly into the text boxes (in base-10 format), or select them from the input history via the drop-down list (up to 10 items).

If the left-hand text box is left blank, it is treated as if "0" were specified.

If the right-hand text box is left blank, it is treated as if the last number were specified.

<3> Click the [Search Backward]/[Search Forward] button

When the [Search Backward] button is clicked, search is taken place in the order from the large number to small and the search results are shown selected in the Trace panel.

When the [Search Backward] button is clicked, search is taken place in the order from the small number to large and the search results are shown selected in the Trace panel.

(c) Search with specifying the global variable

Select the [The execution part is retrieved specifying the global variable] item in the [Search object] area and then follow the operation below.

<1> Specify [Variable Name]

You can either type the variable name to be find directly into the text box, or select it from the input history via the drop-down list (up to 10 items).

Searches are case-insensitive, and only complete matches are retrieved.

<2> Specify [Kind]

Select the access type ([Reference/Substitution], [Reference], or [Substitution]) from the drop-down list.

<3> Specify [Value]

You can either type the accessed variable value directly into the text box, or select one from the input history via the drop-down list (up to 10 items).

The variable value can also be specified as a range. In this case, specify a range by specifying variable values in both the left and right text boxes.

If the right-hand text box is blank, then access locations with the fixed variable values specified in the left-hand text box will be searched for.

<4> Specify [Number]

Specify the range within the trace data to search via the number displayed in the [Number] area of the Trace panel.

Specify the starting number in the left text box, and the ending number in the right text box ("0" to "*last number*" are specified by default).

You can either type the numbers directly into the text boxes (in base-10 format), or select them from the input history via the drop-down list (up to 10 items).

If the left-hand text box is left blank, it is treated as if "0" were specified.

If the right-hand text box is left blank, it is treated as if the last number were specified.

<5> Click the [Search Backward]/[Search Forward] button

When the [Search Backward] button is clicked, search is taken place in the order from the large number to small and the search results are shown selected in the Trace panel.

When the [Search Backward] button is clicked, search is taken place in the order from the small number to large and the search results are shown selected in the Trace panel.

2.12.9 Save the contents of execution history

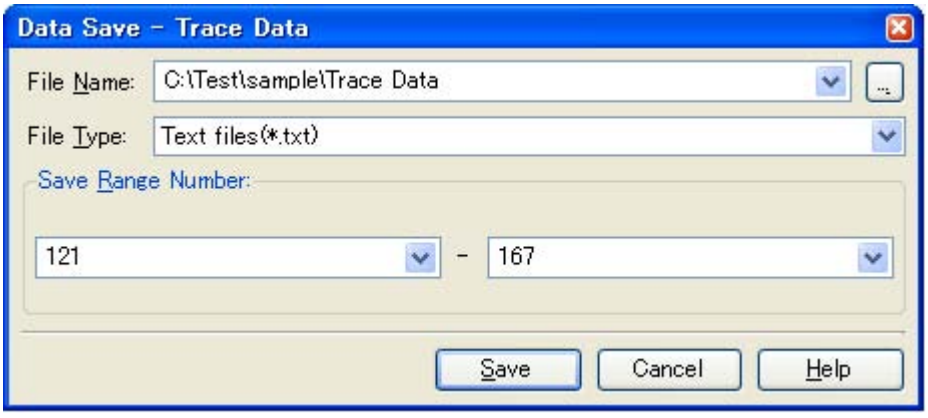
Contents of the collected trace data can be saved with range selection in text files (*.txt)/CSV files (*.csv).

When saving to the file, the latest information is acquired from the debug tool, and it is saved in accordance with the display format on this panel.

The following Data Save dialog box can be opened by selecting the [File] menu >> [Save Trace Data As...].

In this dialog box, follow the steps below.

Figure 2-116. Save Execution History (Data Save Dialog Box)



(1) Specify [File Name]

Specify the name of the file to save.

You can either type a filename directly into the text box (up to 259 characters), or select one from the input history via the drop-down list (up to 10 items).

You can also specify the file by clicking the [...] button, and selecting a file via the Select Data Save File dialog box.

(2) Specify [File Type]

Select the format in which to save the file from the following drop-down list.

The following file formats can be selected.

List Item	Format
Text files (*.txt)	Text format (default)
CSV (Comma-Separated Variables)(*.csv)	CSV format[Note]

Note The data is saved with entries separated by commas (,).
If the data contains commas, each entry is surrounded by double quotes "" in order to avoid illegal formatting.

(3) Specify [Save Range Number]

Specify the range of the number to save via "start number" and "end number".
Directly enter decimal number in each text box or select from the input history displayed in the drop-down list (up to 10 items).
When saving all the trace data, select the [All Trace Data] item in the drop-down list at the left (the right text box becomes invalid).
If a range is selected in the panel, that range is specified as the default. If there is no selection, then the range currently visible in the panel is specified.

(4) Click the [Save] button

Trace data is saved in the specified file with the specified format.

Figure 2-117. Output Example of Trace Data

```
Number  Target  Time  Clock  Line/Address  Pipeline  Source/Disassemble  Address  Data
-------------------------------------------------------------------------------------
Number  Target  Time  Clock  Line/Address  Pipeline  Source/Disassemble  Address  Data
   :       :     :      :            :         :                  :        :
```

Remark Items of trace data output differ depending on the debug tool used.

2.13 Measure Execution Time of Programs

This section describes how to measure the execution time of the program.

Remark For "measurement of execution time of programs" for a microcontroller that supports multi-core, see also to "2.7 Select a Core (PE)".

2.13.1 Measure execution time until stop of the execution

In the debug tool, there is a function to measure the program execution time (Run-Break time) from the start to the stop. Therefore, when the program starts its execution, the execution time is automatically measured.
You can check the result of the measurement by either one of the following.

Cautions 1. The Run-Break time for a step execution cannot be measured.

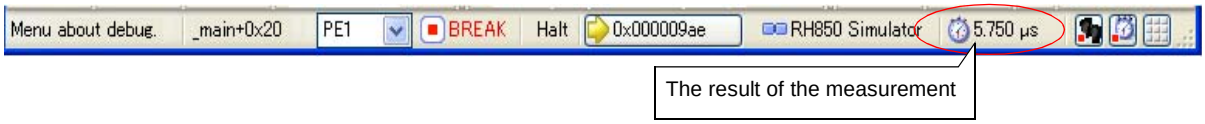
2. [Simulator]
To measure the Run-Break time, [Yes] must be specified with the [Use timer function] property in the [Timer] category oncategory on the [Debug Tool Settings] tab of the Property panel.

Remark This function is operated by a Run-Break Timer event, which is one of the built-in events set by default in the debug tool.

(1) Check in the status bar

After the program is stopped, the result of the measurement is displayed in the status bar on the Main window (when measurements have not been performed yet, "Not measured" is displayed).

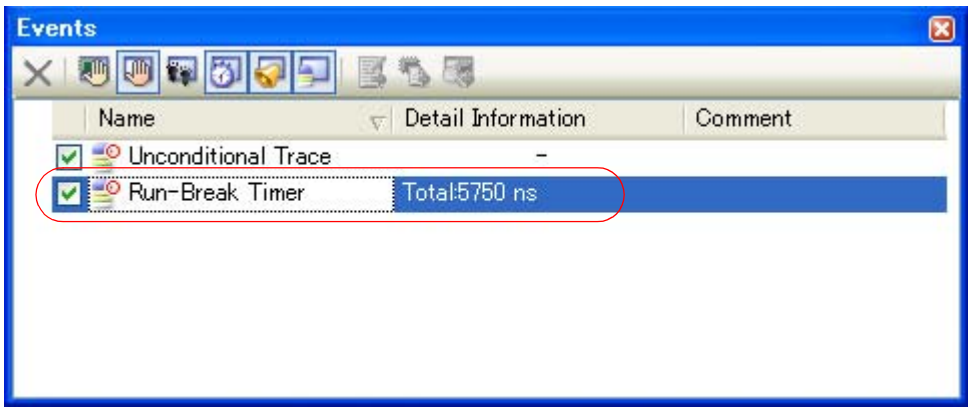
Figure 2-118. Example of Result of Run-Break Timer Event (Status Bar)



(2) Check on the Events panel

After the program is stopped, the result of the measurement is displayed in the Events panel opened by selecting the [View] menu >> [Event], in event type as "Run-Break Timer".

Figure 2-119. Example of Result of Run-Break Timer Event (Events Panel)



2.13.2 Measure execution time in the arbitrary section [Simulator]

In the program execution process, the execution time in the arbitrary section can be measured by setting Timer Result event. This Timer Result event consists of a timer start event and a timer end event.

To use this function, follow the procedure described below.

(1) Set a Timer Result event
(2) Execute the program
(3) Delete a Timer Result event

Cautions 1. Also see "2.16.6 Notes for setting events" for details on Timer Result events (e.g. limits on the number of enabled events).

2. To use this function, [Yes] must be specified with the [Use timer function] property in the [Timer] category on the [Debug Tool Settings] tab of the Property panel.

(1) Set a Timer Result event

To set a Timer Result event, set a timer start event and a timer end event that starts/stops a timer measurement. Use one of the following methods to set a timer start event and a timer end event.

(a) For execution-related events

Perform this operation in the Editor panel/Disassemble panel in which the source text/disassembled text is displayed.

Follow the operation listed below from the context menu, in accordance with your desired event type, after moving the caret to the target line that has a valid address.

Event Type	Operation
Timer start	Select [Timer Settings] >> [Start Timer] >> [Set Timer *n*[Note]]
Timer end	Select [Timer Settings] >> [Stop Timer] >> [Set Timer *n*[Note]]

Note Select the channel number (*n*: 1 to 8) in which a Timer Result event is set.

Caution Simulator will not include the time for a timer end event in the measurement results. For this reason, set a timer end event to one line below the range for which you wish to measure the run time.

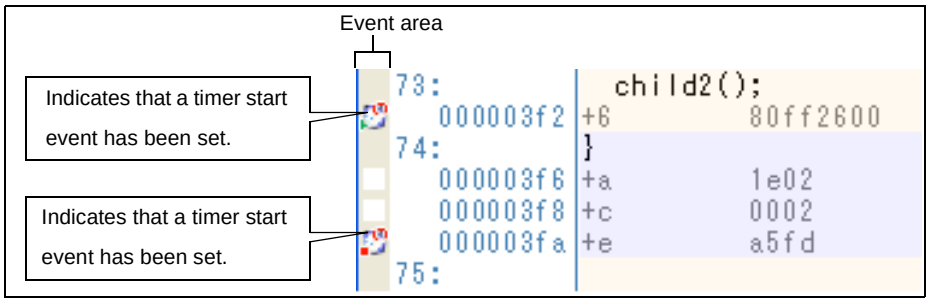
A timer start event or a timer end event is set to the instruction at the start address corresponding to the line of the caret position.

Once a timer start event or a timer end event is set, the following event mark is displayed in the event area of the line/address that an event is set.

Table 2-14. Event Marks of Timer Start Event/Timer End Event

Event Type	Event Mark
Timer start	
Timer end	

Figure 2-120. Timer Start and Timer End Events Setting Example (Disassemble Panel)

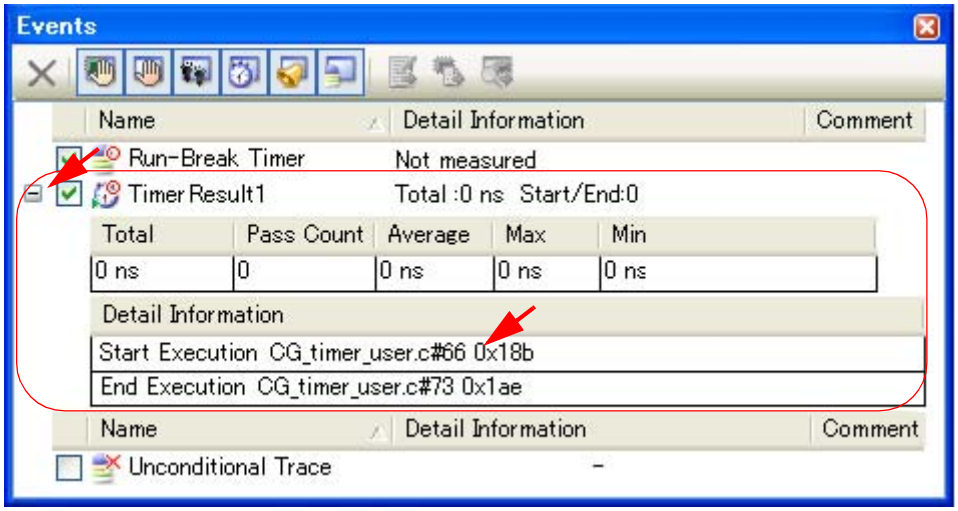


(b) For access-related events

In this product version, this function is not supported.

When a timer start event and a timer end event are set, they are managed collectively on the Events panel as one instance of a Timer Result event (see "2.16 Manage Events"). When you click the "+" mark at a Timer Result event item, detailed information on the timer start event and the timer end event you have set is displayed.

Figure 2-121. Example of Timer Start and Timer End Events (Execution Type) in Events Panel



Remark Event marks differ depending on the event state (see "2.16.1 Change the state of set events (valid/invalid)").

When an event is set at the point which other event is already set, the event mark ( ) is displayed meaning more than one event is set at the point.

(2) Execute the program

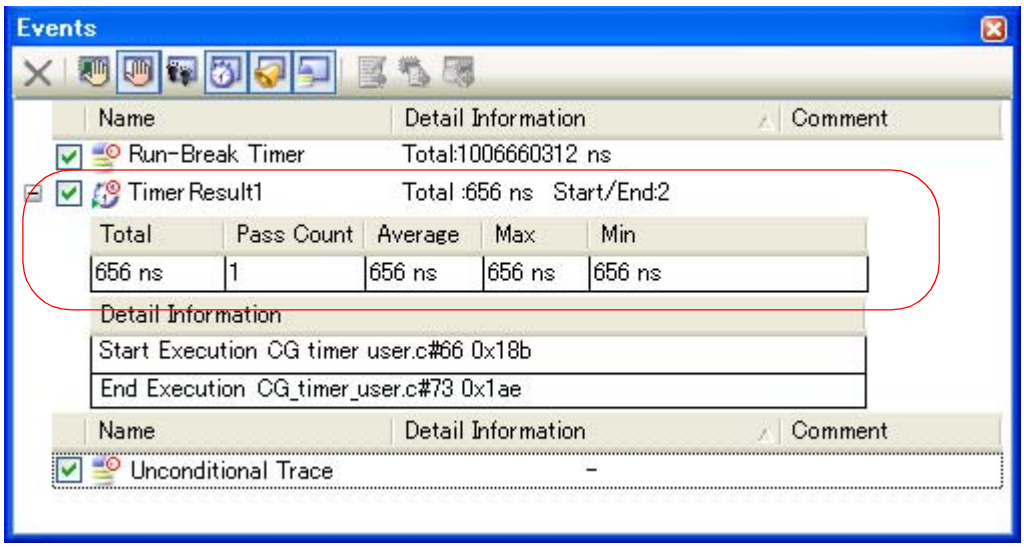
Execute the program (see "2.8 Execute Programs").

When an instruction for which a timer start event or a timer end event has been set is executed, a timer measurement is started or finished.

After the program is stopped, the result of the measurement is displayed in the Events panel opened by selecting the [View] menu >> [Event], in event type as "Timer Result".

This Timer Result is a particular type of event that is displayed on only the Events panel when either a timer start event or a timer end event has been set.

Figure 2-122. Example of Result of Timer Result Event (Timer Start Event/Timer End Event)



(3) Delete a Timer Result event

To delete a Timer Result event you have set, on the Editor panel/Disassemble panel, right-click the event mark in the event area and select [Delete Event] from the context menu that is displayed.

Also, there is another way to delete a set event. Select the Timer Result event you want to delete on the Events panel, and then click the button in the toolbar (see "2.16.4 Delete events").

Caution If either a timer start or timer end event is deleted from the event marks on the event area, all of the corresponding event marks are deleted.

2.13.3 Measurable time ranges

The scope of time that can be measured via timers using Run-Break Timer events (see "2.13.1 Measure execution time until stop of the execution" for details) or Timer Result events (see "2.13.2 Measure execution time in the arbitrary section [Simulator]" for details) is shown below.

Table 2-15. Measurable Time Ranges

Debug Tool	Run-Break Timer Event		Timer Result Event
Full-spec emulator	Min.	60 ns	-
	Max.	Approx. 4 min. 20 s (LPD: 4-pin) Overflow detection included	
E1/E20	Min.	60 ns	-
	Max.	Approx. 4 min. 20 s (LPD: 4-pin) Overflow detection included	
Simulator	Depends on the clock frequency for timer/trace		Depends on the clock frequency for timer/trace

2.14 Measure Coverage [Simulator]

This section describes coverage measurements that are conducted using the coverage function.

There are several kinds of coverage measurement methods. Of these, CubeSuite+ performs, in areas designated below, a code coverage measurement of fetch-related operations on source lines and functions (C0 coverage) and a data coverage measurement of access-related operations on variables.

The areas in which CubeSuite+ performs coverage measurements are as follows:

- 1 MByte space of addresses 0x000000 to 0x0FFFFF in the internal ROM area(fixed measurement area)
- Any 1 MByte space other than the fixed measurement area above (see "2.14.1 Configure the coverage measurement")

Remarks 1. C0 coverage: Instruction coverage (statement coverage)
For example, if all instructions (statements) in code are executed at least once, then C0 = 100%.

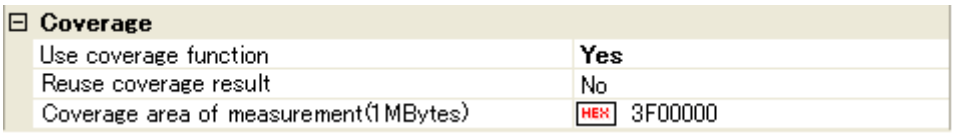
2. For "coverage measurements" for a microcontroller that supports multi-core, see also to "2.7 Select a Core (PE)".

2.14.1 Configure the coverage measurement

You need to configure the code coverage measurement before using the coverage function.

You can configure the coverage measurement function in the [Coverage] category on the [Debug Tool Settings] tab of the Property panel as follows:

Figure 2-123. [Coverage] Category



(1) [Use coverage function]

Select whether to use the coverage function.

Select [Yes] to use the coverage function (default: [No]).

(2) [Reuse coverage result]

This property appears only when the [Use coverage function] property is set to [Yes].

The currently obtained results of code coverage measurements are automatically saved when CubeSuite+ is disconnected from the debug tool. The next time it is connected to the debug tool, specify whether or not you want to reproduce the contents of saved measurement results.

Select [Yes] to reproduce the contents of previously obtained code coverage measurement results (default: [No]).

Caution This function applies to only the internal ROM area.

(3) [Coverage area of measurement(1MBytes)]

This property appears only when the [Use coverage function] property is set to [Yes].

Specify the code coverage measurement area.

Directly enter the start address of any 1 Mbyte space other than the internal ROM area (0x000000 - 0x0FFFFF) in hexadecimal number (default: [100000]).

2.14.2 Display the coverage measurement result

When the program starts running, a coverage measurement is automatically begun, and when the program stops running, the coverage measurement is terminated at the same time.

(1) Code coverage ratio

(a) Display of code coverage ratios for source text lines and disassembled text lines

This display is on the Editor panel/Disassemble panel that displays the target program.
On each panel, the source text lines and disassembled lines for which coverage was measured have their backgrounds displayed, as in "Table 2-17.", in classified colors based on the code coverage ratios calculated following the calculation methods in "Table 2-16.".
However, when disconnected from the debug tool, or while the program is under execution, the results are not displayed.
Note that the obtained code coverage measurement results can be reset in whole by selecting [Clear Coverage Information] from the context menu in the Editor panel/Disassemble panel (the color-coded display on each panel are also cleared).

Caution When the selected microcontroller supports multi-core, in the Local RAM self area, note that the measurement results will be displayed only for the access in the currently selected PE*n* (see "2.7 Select a Core (PE)").

Table 2-16. Method for Calculating Code Coverage Ratio for Source Lines and Disassemble Lines

Panel	Calculation Method
Editor panel	"Number of bytes of code executed in the address range corresponding to the source text line" / "Total number of bytes of code in the address range corresponding to the source text line"
Disassemble panel	"Number of bytes of code executed in the address range corresponding to the disassembled text line" / "Total number of bytes of code in the address range corresponding to the disassembled text line"

Table 2-17. View of Code Coverage Measurement Result (Default)

Code Coverage	Background Color
100 %	*Source text/disassembled text*
1 to 99 %	*Source text/disassembled text*
0 % (not yet executed)	*Source text/disassembled text*

Remarks 1. Code coverage measurement results are automatically updated at a break in each panel.

2. The above background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box.

3. The above background colors do not apply to the lines that are outside of the subject area.

4. If the downloaded lode module file is older than the source file currently being open, the displaying of the code coverage measurement result is not performed in the Editor panel.

Figure 2-124. View of Code Coverage Measurement Result (Editor Panel)

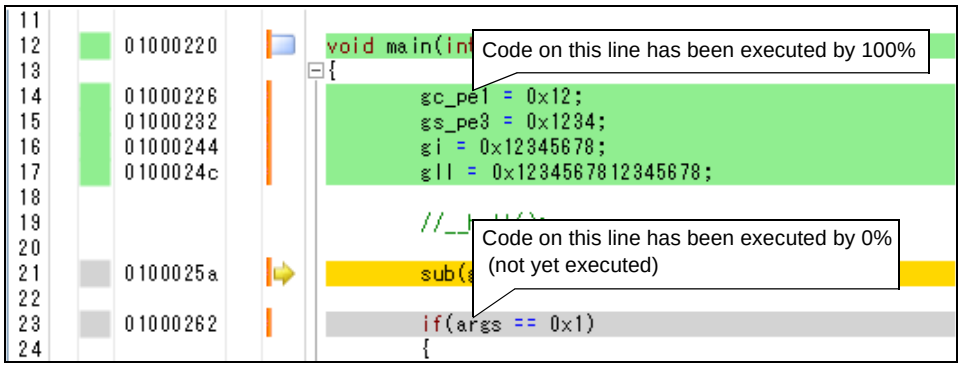
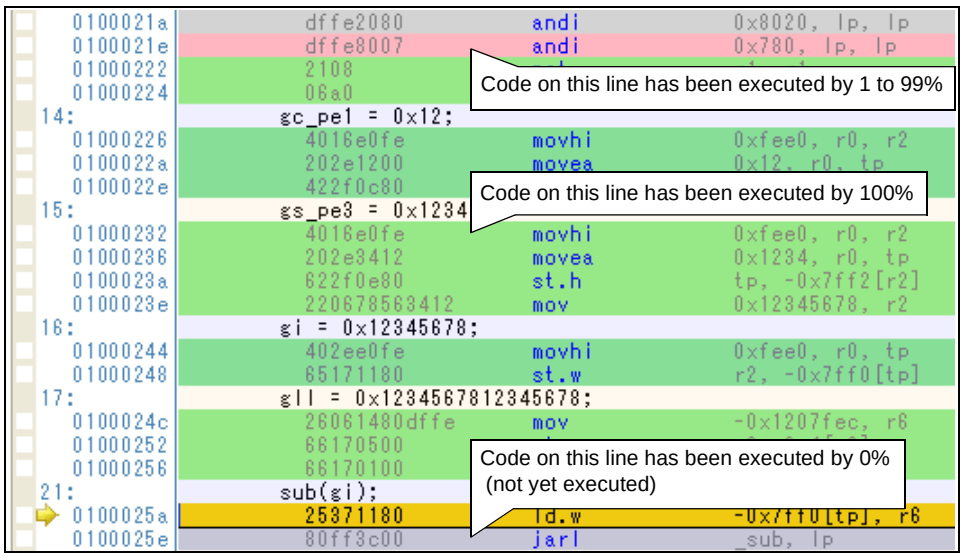


Figure 2-125. View of Code Coverage Measurement Result (Disassemble Panel)



(b) Display of code coverage ratios for each function

Code coverage ratios for each function can be checked via the [Code Coverage[%]] item in the Function List panel of the analyze tool. For details on "the code coverage ratio of the function", see "CubeSuite+ Integrated Development Environment User's Manual: Analysis".

(2) Data coverage ratio

Data coverage ratios for each variable can be checked via the [Data Coverage[%]] item in the Variable List panel of the analyze tool. For details on "the data coverage ratio of the variable", see "CubeSuite+ Integrated Development Environment User's Manual: Analysis".

2.15 Set an Action into Programs

This section describes how to set the specified action into the program.

2.15.1 Inset printf

By setting a Printf event that is one of "action events", the value of the specified variable expression can be output to the Output panel by executing a printf command after temporarily stopping the program in execution at an arbitrary position.

To use this function, follow the steps below.

Cautions 1. [Full-spec emulator][E1][E20]
Printf event is implemented using the Software break function [Full-spec emulator][E1][E20]. Therefore, you need to select [Yes] in the [Use software break] property in the [Break] category on the [Debug Tool Settings] tab of the Property panel before setting a Printf event.

2. Also see "2.16.6 Notes for setting events" for details on a Printf event (e.g. limits on the number of enabled events).

3. No action events occur during step execution (/ /) or program execution ignoring break-related events ( ).

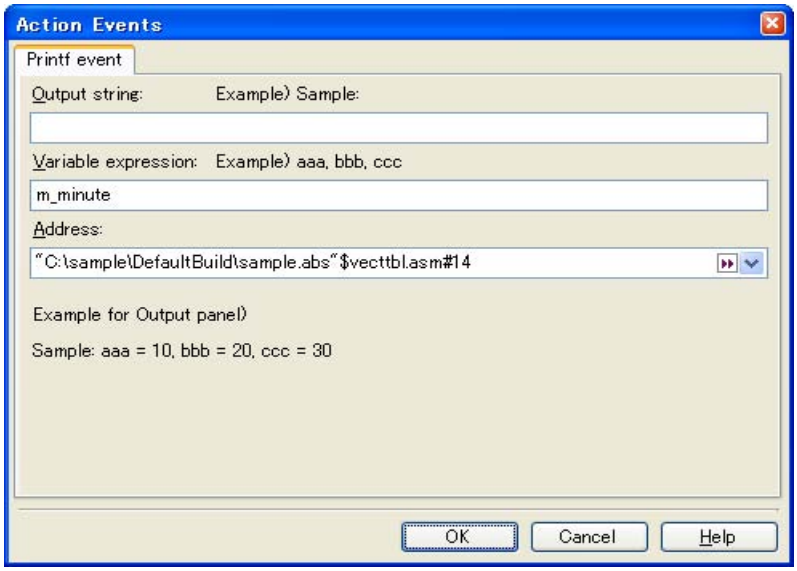
(1) Set a Printf event

Set a Printf event to the position where you want to execute the printf command in the Editor panel/Disassemble panel.

On each panel, select [Register Action Event...] from the context menu after moving the caret to the line that has a valid address to open the Action Events dialog box below.

In this dialog box, follow the steps below.

Figure 2-126. Set Printf Event (Action Events Dialog Box: [Printf event] tab)



(a) Specify [Output string]

Directly enter from the keyboard the characters to add when output to the Output panel.

Characters must be in one line (spaces allowed).

(b) Specify [Variable expression]

Specify the variable expression for the Printf event to take place.

Type a variable expression directly into the text box (up to 1024 characters).
You can specify up to 10 variable expressions for a single Printf event by separating them with commas ",".
If this dialog box opens with a variable expression selected in the Editor panel/Disassemble panel, the selected variable expression appears as the default.
For the basic input format that can be specified as variable expressions and the values output by Printf event, see "Table A-11. Relationship between Variable Expressions and Output Value (Printf Event)".

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

(c) Specify [Address]

Specify the address at which to set the Printf event.
The address of the location currently being specified is displayed by default.
If you want to edit this area, you can either type an address expression directly into the text box (up to 1024 characters), or select them from the input history via the drop-down list (up to 10 items).

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

(d) Click the [OK] button

Set the Printf event to the line/address at the caret position in the Editor panel/Disassemble panel.
When the Printf event is set, the mark is displayed in the event area on the Editor panel/Disassemble panel, and the set Printf event is managed in the Events panel (see "2.16 Manage Events").

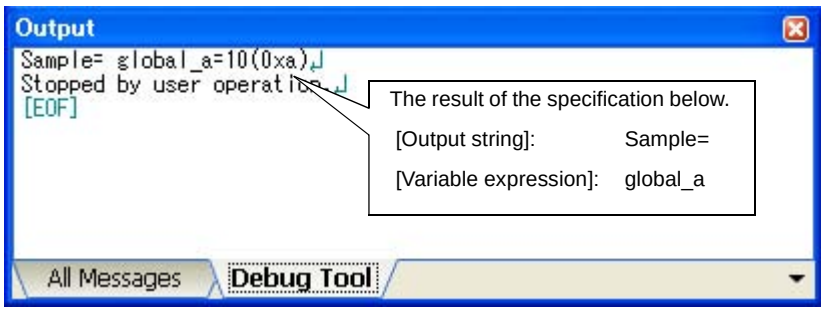
(2) Execute the program

Execute the program (see "2.8 Execute Programs").
By executing the program, the program momentarily stops immediately before executing the instruction at the location where this event is set, and the value of the variable expression specified in this dialog box is output to the Output panel.

(3) Check the output result

The output result format from the Printf event in the [Debug Tool] tab of the Output panel are as follows (see "Figure A-47. Output Result Format of Printf Event"):

Figure 2-127. Example of Output Result of Printf Event



(4) Edit the Printf event

You can edit the Print event that has been set once.
To do this, on the Events panel, select [Edit Condition...] from the context menu after selecting the Printf event to be edited. On the Action Events dialog box opened automatically, edit the items, and then click the [OK] button.

2.16 Manage Events

An event represents a certain status of the target system when debugging such as "Address 0x1000 is fetched" and "Data is written to address 0x2000".

In CubeSuite+, these events are used as the action trigger of the debug function such as breakpoint, start/stop the tracing, and start/stop the timer.

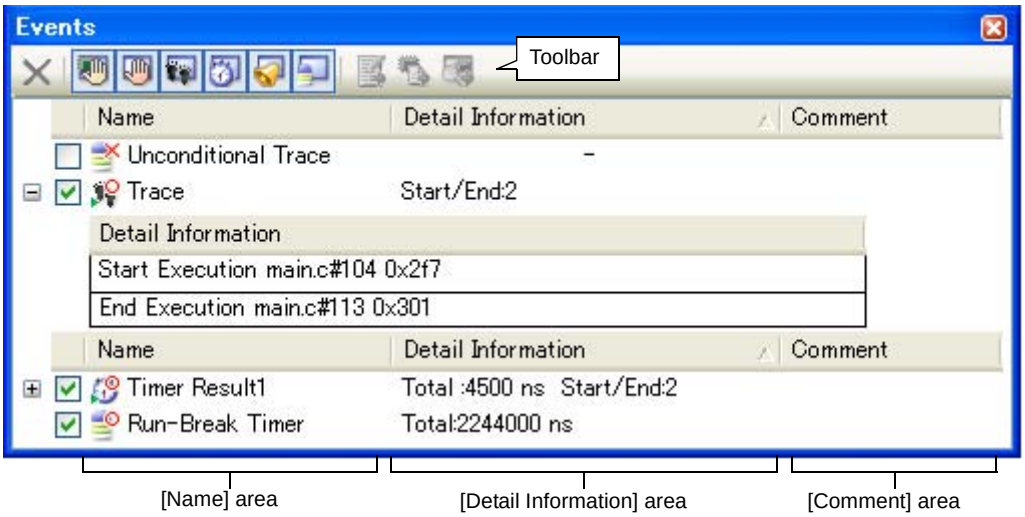
This section describes how to manage those events.

Select the [View] menu >> [Event].

Events are all managed in the Events panel. In the Events panel, you can confirm the details of the currently set events in a list, and they can be deleted and changed enable/disable status.

For details on the contents and function in each area, see the section for the Events panel.

Figure 2-128. Manage Events (Events Panel)



Remarks 1. For "event occurrence" for a microcontroller that supports multi-core, see also to "2.7 Select a Core (PE)".

2. For details on how to set various events, see the section below:

- "2.9.3 Stop the program at the arbitrary position (breakpoint)"
- "2.9.4 Stop the program at the arbitrary position (break event)"
- "2.9.5 Stop the program with the access to variables/I/O registers"
- "2.12.3 Collect execution history in the arbitrary section"
- "2.12.4 Collect execution history only when the condition is met [Simulator]"
- "2.13.2 Measure execution time in the arbitrary section [Simulator]"
- "2.15.1 Inset printf"

2.16.1 Change the state of set events (valid/invalid)

By changing the check on the check box of the event name, the setting state of the event can be changed (the Event mark is changed depending on the setting state of the event).

The following are types of the setting state of the event.

Figure 2-129. Event Name Check Box

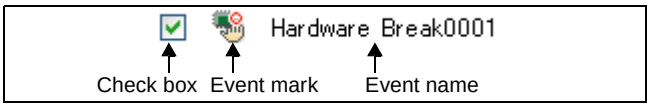


Table 2-18. Setting State of Event

☑	Valid state	Event occurs when the specified condition is met. It is possible to set the event to an invalid state by removing the check.
☐	Invalid state	Event does not occur when the specified condition is met. It is possible to set the event to a valid state by removing the check.
■	Suspended State	A specified condition cannot be set in the program to be debugged. It is not possible to operate the check box.

Remarks 1. Both of the timer start event and the timer end event is must be set for the Timer Result event. Therefore, it is not possible to set a particular event to a valid state by only the setting of one of these (at the same time as both events are set, they are treated as grouped events as a Timer Result).

2. It is not possible to set the Run-Break Timer event to an invalid/suspended state.

3. The setting state of the event can be changed from the menu displayed by right clicking on the Event mark in the Editor panel/Disassemble panel.

4. The setting of the Unconditional Trace event and the Trace event to valid or invalid state is exclusively controlled. Therefore, the Unconditional Trace event, which is a built-in event, is valid state by default, but if either a trace start event/trace end event is set, it automatically becomes invalid state, and the Trace event, which is a event name that is collectively called with a trace start event and a trace end event, becomes valid state. Conversely, if the set Trace event is invalid state, the Unconditional Trace event automatically becomes valid state.

2.16.2 Display only particular event types

Click on the toolbar button to display only the particular event type.

	Displays events related to the Hardware Break.
[Full-spec emulator][E1][E20]	Displays events related to the Software Break.
	Displays events related to the trace.
	Displays events related to the timer.
	Displays events related to the action event (Printf event).
	Displays the built-in events (Unconditional Trace event and Run-Break Timer event).

2.16.3 Jump to the event address

Clicking the following buttons jumps to each panel which selected events address exist.

Note however, that when a Trace event/Timer Result event/ Unconditional Trace event/ Run-Break Timer event is selected, these buttons are disabled.

	Opens the Editor panel and jumps to the source line corresponding to the address where the selected event is being set.
	Opens the Disassemble panel and jumps to the disassemble results corresponding to the address where the selected event is being set.
	Opens the Memory panel and jumps to the source line corresponding to the address where the selected event is being set.

2.16.4 Delete events

To delete any event and event condition you have set, select the event and click the  button on the toolbar.
Note that it is not possible to delete the built-in events (Unconditional Trace event and Run-Break Timer event).

Remarks 1. For the Break event of execution type, it is possible to delete the set event to click the event mark displayed in the Editor panel/Disassemble panel.

2. To delete all of the events and event conditions you have set at a time, select [Select All] from the context menu, then click the  button (note, however, that it is not possible to delete the built-in events).

2.16.5 Write comment to events

The user can write comments for each event that has been set.

To input comments, click the [Comment] area after selecting the event to input comments, then input directly the desired text from the keyboard (the edit mode is cancelled by pressing the [Esc] key).

After editing the comments, complete the editing by pressing the [Enter] key or moving the focus to outside the edit region.

Up to 256 characters can be inputted for the comments, and this is saved as the settings of the user during use.

2.16.6 Notes for setting events

This section describes notes for setting each type of event.

(1) Maximum number of enabled events
(2) Event types that can be set and deleted during execution
(3) Other notes

(1) Maximum number of enabled events

The number of events that can be set to Valid state at the same time are subject to the following limitations.
Consequently, if enabling a new event would exceed the limit, you must first set some other event to Invalid state.

Table 2-19. Maximum Number of Enabled Events

Event Type	Debug Tool		
	Full-spec emulator	E1/E20	Simulator
Hardware Break (execution type: after execution)	-		-
Hardware Break (execution type: before execution)	12[Note 1]		64
Hardware Break (access type)	4[Note 1]		
Software Break	2000 (only for the code flash)		-
Trace (trace start/trace end)	8 + 7		64
Point Trace	-		64
Timer Result (timer start/timer end)	-		1[Note 2]
Action (Printf)	100		64[Note 3]

"*x* + *y*": "Hardware Break (execution type): *x*" + "Hardware Break (access type): *y*"

Notes 1. Among Hardware Break (execution type: before execution) events, four events are shared with Hardware Break (access type) events (e.g. when four Hardware Break (access type) events are used, only eight hardware break (execution type: before execution) events can be used).

2. Only a single timer start event and a single timer end event can be specified.

3. Combination with Hardware Break events

(2) Event types that can be set and deleted during execution

The following types of events can be set or deleted during program execution, or during tracer/timer execution.

Table 2-20. Event Types That Can be Set and Deleted during Execution

Event Type	Debug Tool		
	Full-spec emulator	E1/E20	Simulator
Hardware Break (execution type: after execution)	-		-
Hardware Break (execution type: before execution)	△		▲
Hardware Break (access type)	△		▲
Software Break	NG		-
Trace (trace start/trace end)	▲		▲
Point Trace	-		▲
Timer Result (timer start/timer end)	-		NG
Action (Printf)	NG		▲

△ : Possible, if the program execution is allowed to pause for events[Note]

▲ : Impossible while tracer or timer is executing

NG : Impossible

\- : Not supported

Note To enable this, specify [Yes] with the [Set event by stopping execution momentarily] property in the [Set Event While Running] category on the [Debug Tool Settings] tab of the Property panel.

(3) Other notes

- No events can be set to local variables.

- Events do not occur during step execution (including return execution) and program execution by selecting [Go to Here] from the context menu.

- If the location set for an existing event changes to midway in an instruction because the program to debug has been downloaded again, re-set the event using the following method:
  - When debugging information is available:
    The location setting of events is always moved to the beginning of the source text line.
  - When debugging information is not available:
    Depends on the [Automatic change method of event setting position] property in the [Download] category on the [Download File Settings] tab of the Property panel.

- If a change to internal ROM/RAM changes the location the event is set to a non-mapped area, then set events will not occur (they will also not change to Invalid state /Suspended State on the Events panel).

- If you differentiate function or variable names by leading underscores, then CubeSuite+ may misrecognize them, and convert symbols or make break event settings invalid. This applies for cases like when you have two functions, one named "_reset" and the other named "__reset".

2.17 Use Hook Function

This section describes how to set hooks in the debug tool by using the hook function.

By setting a hook transaction, you can automatically change the values of the I/O register/CPU register before and after downloading a load module or after resetting the CPU.

Configure the hook transaction in the [Hook Transaction Settings] category on the [Hook Transaction Settings] tab of the Property panel.

Remark By setting a I/O register by using the [Before download] property, for example, downloading can be executed at high speeds. Downloading to the external RAM is also facilitated by using this function.

Figure 2-130. [Hook Transaction Settings] Category

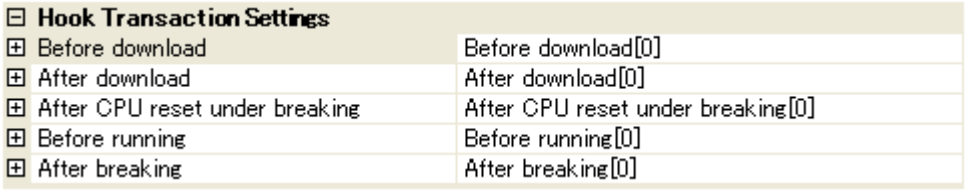


Table 2-21. Properties in [Hook Transaction Settings] Category

Property	Description
Before download	Perform the specified process immediately before downloading the load module file.
After download	Perform the specified process immediately after downloading the load module file.
After CPU reset under breaking	Perform the specified process immediately after resetting the CPU under breaking.
Before running	Perform the specified process immediately before starting program execution.
After breaking	Perform the specified process immediately after breaking program execution.

The properties in the [Hook Transaction Settings] category indicate the timing with which the hook process will be performed. "[]" indicates the current number of specified processes (no hook processes are configured by default).

Specify the target process in the property for which you want the hook process to be performed.

To specify a process, select the target property, then open the Text Edit dialog box by clicking the [...] button that appears on the right edge of the field.

Figure 2-131. Opening Text Edit Dialog Box

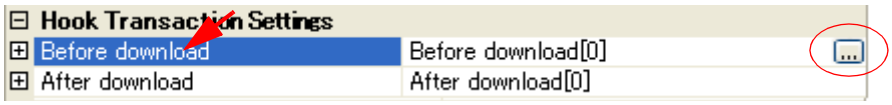
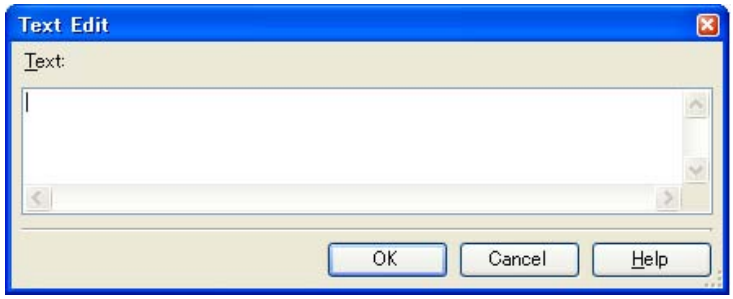


Figure 2-132. Use Hook Function (Text Edit Dialog Box)



In this dialog box, directly enter the desired process from the key board.
The format for specifying processes is as follows:

[Process 1]:

Automatically overwrites the value of *I/O register* with *Value*.
Specification format:

```
I/O-register-name Value
```

[Process 2]:

Automatically overwrites the value of *CPU register* with *Value*.
Specification format:

```
CPU-register-name Value
```

[Process 3]:

Automatically executes a script file which is specified with *Python script path* (absolute path or relative path from the project folder).
Specification format:

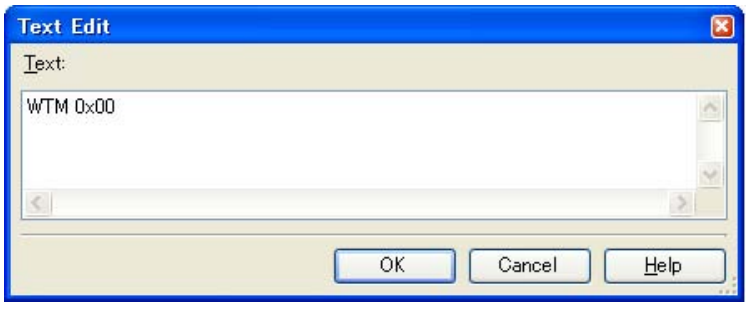
```
Source Python-script-path
```

Remarks 1. When specifying hook processes, lines starting with a hash mark "#" will be treated as comments.

2. A tab character can be used instead of the space character.

Up to 64 characters for one process, and up to 128 processes for each property can be set (one line in the [Text] area in the Text Edit dialog box is equivalent to one processing).
After the specification of the process is complete, click the [OK] button to set the process to the Property panel.

Figure 2-133. Example of Hook Transaction



2.18 About Input Value

This section describes consideration to take when inputting values in each panel and dialog box.

2.18.1 Input rule

Following is the rules for input to each panel/dialog box.

(1) Character set

Character sets that are allowed to input are as follows:

Table 2-22. List of Character Set

Character Set	Outline
ASCII	1- byte alphabets, numbers, symbols
Shift-JIS	2-byte alphabet, number, symbol, Hiragana, Katakana, Kanji and 1-byte Katakana.
EUC-JP	2-byte alphabet, number, symbol, Hiragana, Katakana, Kanji and 1-byte Katakana.
UTF-8	2-byte alphabet, number, symbol, Hiragana, Katakana, Kanji (include Chinese characters) and 1-byte Katakana.
UTF-16	2-byte alphabet, number, symbol, Hiragana, Katakana, Kanji (include Chinese characters) and 1-byte Katakana.

(2) Escape sequence

Escape sequences that are allowed to input are as follows:

Table 2-23. Escape Sequence List

Escape Sequence	Value	Outline
\0	0x00	null character
\a	0x07	Alert
\b	0x08	Backspace
\t	0x09	Horizontal tab
\n	0x0A	New line
\v	0x0B	Vertical tab
\f	0x0C	Form feed
\r	0x0D	Carriage return
\"	0x22	Double-quotation mark
\'	0x27	Single-quotation mark
\?	0x3F	Question mark handled as a question mark if ? is entered.
\\	0x5C	Backslash

(3) Number

Notations allowed when entering numbers are as follows:

Table 2-24. Notation List

Notation	Outline
Binary number	Start with 0b and continues with the numbers from 0 to 1. (Case insensitive for alphabets)
Octal number	Start with 0 and continues with the numbers from 0 to 7.
Decimal	Start without 0 and continues with the numbers from 0 to 9.
Hexadecimal number	Start with 0x and continues with the numbers from 0 to 9 and alphabets a to f. (Case insensitive for alphabets) In the input area with the HEX mark, prefix 0x is not needed.

(4) Expression and operator

Expression represents constants, CPU register name, I/O register name and symbols and those connected with operators.

An expression comes in two types: an address expression and a watch-expression. The expression that requires the address of a symbol is referred to as an address expression, and the one that requires the value of a symbol is referred to as a watch-expression.

(a) An address expression and operators

With an address expression, the address of a symbol is used to perform operations. Only when a CPU register name is written, the value of the symbol is used to perform operations.

The basic input formats of address expressions are as follows:

Table 2-25. Basic Input Format of Address Expressions

Expression	Description
Name of a C language variable[Note 1]	Address of a C language variable
Expression [*Expression*[Note 2]]	Address of an array
Expression.Member name	Address of a structure/union/class member
Expression->Member name	Address of a structure/union/class member that is pointed to
Name of a CPU register	Value of the CPU register
Name of an I/O register	Address of the I/O register
Label name[Note 3], EQU symbol name[Note 3] and [immediate value]	Address of a label, a value of an EQU symbol, and an immediate address
Integer constant	Address

Notes 1. If the register is assigned the value of a C variable, an error results.

2. The expression that is input as an index to an array is parsed as a watch-expression.

3. If the label name or EQU symbol name includes a "$", be sure to enclose the name in "{ }" (e.g. {$Label}).
When you specify the CPU register name "I", add ":REG" (e.g. I:REG) to distinguish it from the keyword "I" that indicates an imaginary number.

From "Table 2-25. Basic Input Format of Address Expressions", the following expressions with operator can be constructed.

Table 2-26. Construction of Expressions with Operators

Expression	Description
(*Expression*)	Value of the parenthetical watch-expression
! *Expression*	Inverts symbol
- *Expression*	Logical negation
~ *Expression*	Bit inversion
Expression * *Expression*Note	Multiplication
Expression / *Expression*Note	Division
Expression % *Expression*Note	Remainder calculation
Expression + *Expression*Note	Addition
Expression . *Expression*Note	Subtraction
Expression & *Expression*Note	Logical multiplication by bits
Expression ^ *Expression*Note	Exclusive disjunction by bits
Expression | *Expression*Note	Logical sum by bits

Note Variables and functions can be combined by an operator only with variables, functions and integer constants.

Example C variable name + I/O register name

(b) Watch-expression and operator

With watch-expression, the value of a symbol is used to perform operations. Only when the value does not exist, the address of the symbol is used to perform operations (e.g. main( ) + 1).

The basic input formats of watch-expressions are as follows:

Table 2-27. Basic Input Format of Watch-expressions

Expression	Description
Name of a C language variable	Address of a C language variable
Expression [*Expression*	Element of an array
Expression.Member name	Value of a structure/union/class member
Expression->Member name	Value of a structure/union/class member that is pointed to
**Expression*	Value of pointer variable
&*Expression*	Location address
Name of a CPU register	Value of the CPU register
Name of an I/O register	Value of the I/O register
Label nameNote, EQU symbol nameNote and and [immediate value]	Values of a label, a value of an EQU symbol, and an immediate address
Integer constant	Integer constant value
Floating constant	Floating point constant value
Character constant	Character constant value

Note If the label name or EQU symbol name includes a "$", be sure to enclose the name in "{ }" (e.g. {$Label}).
Any imaginary number must be multiplied by an uppercase "I" (e.g. 1.0 + 2.0*I). When you specify the CPU register name "I", add ":REG" (e.g. I:REG) to distinguish it from the keyword "I" that indicates an imaginary number.

From "Table 2-27. Basic Input Format of Watch-expressions", the following watch-expressions with operator can be constructed. For the operators listed in the table below, the expression is parsed according to C language specifications.

Table 2-28. Construction of Expressions with Operators

Expression	Description
(*Expression*)	Specifies the order in which operations are performed
! *Expression*	Inverts symbol
- *Expression*	Logical negation
Expression * *Expression*[Note]	Multiplication
Expression / *Expression*[Note]	Division
Expression % *Expression*[Note]	Remainder calculation
Expression + *Expression*[Note]	Addition
Expression . *Expression*[Note]	Subtraction
Expression & *Expression*[Note]	Logical multiplication by bits
Expression ^ *Expression*[Note]	Exclusive disjunction by bits
Expression | *Expression*[Note]	Logical sum by bits

Note Variables and functions can be combined by an operator only with variables, functions and integer constants.
Example C variable name + I/O register name

2.18.2 Symbol name completion function

This function helps users input data by selecting one of the listed symbol names that exist in the program, when specifying an address expression and so on.

The list of symbol names appears by pressing the [Ctrl] + [Space] keys when a part of the target symbol name is being input in the text box that supports this function. In this list, double-click the target symbol name (or press the [Space]/[Enter] key after selecting it by using the [Up]/[Down] key) to complement the symbol name currently being input.

At this time, if a key other than the [Space]/[Enter] key is pressed or the focus moves to outside the panel/dialog box currently being operated, then the list of symbol names will disappear (the symbol name completion will not be performed).

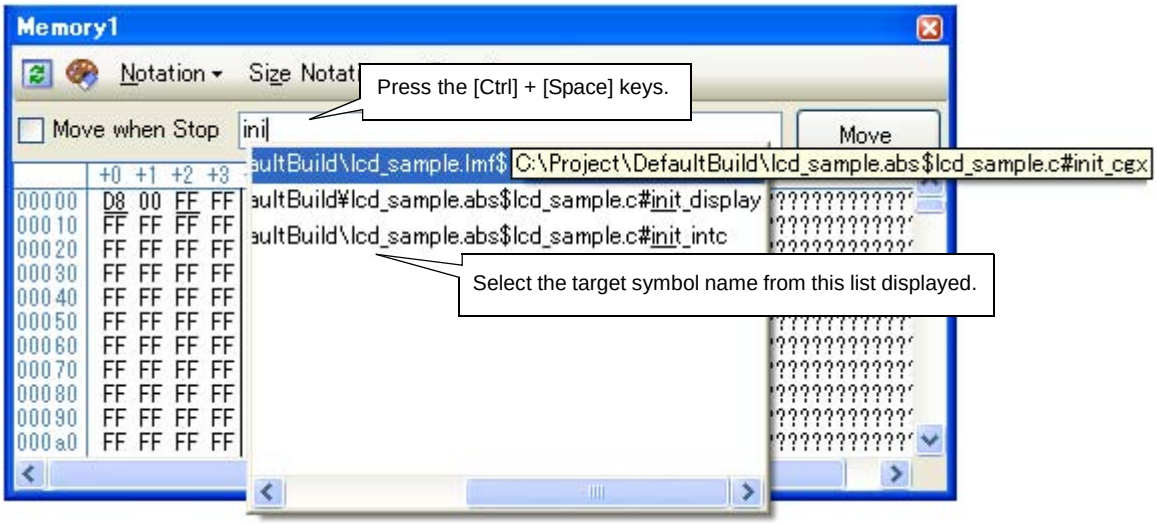
Cautions 1. If there are no character strings in the text box or there are no candidates of the symbol, then the list of symbol names will not appear.

2. Since the information for use by the symbol name completion function is generated while symbols are being downloaded, the time taken for downloading and the memory usage on the host machine will increase when this function is enabled. Therefore, if you do not intend to use the symbol name completion function, we recommend invalidating this function by selecting

[No] in the [Generate the information for input completion] item in the Download Files dialog box ([Yes] is selected by default).
Note, however, that if GHS compiler is used, it is not possible to invalidate this function (a specification of the [Generate the information for input completion] item will be ignored).

Remark See the explanation of the corresponding panel/dialog box as to whether this function can be used or not when inputting a symbol name.

Figure 2-134. Symbol Name Completion Function



2.18.3 Icons for invalid input

In some of the dialog boxes in CubeSuite+, the icon will appear at a point where incorrect characters are entered as a warning sign.

Remark Placing the cursor over the icon will pop up the information that indicates the characters to be entered.

APPENDIX A WINDOW REFERENCE

Appendix A provides detailed explanations of windows/panels/dialog boxes used for debugging with CubeSuite+.

A.1 Description

Windows/panels/dialog boxes for debugging are listed below.

Table A-1. Window/Panel/Dialog Box List

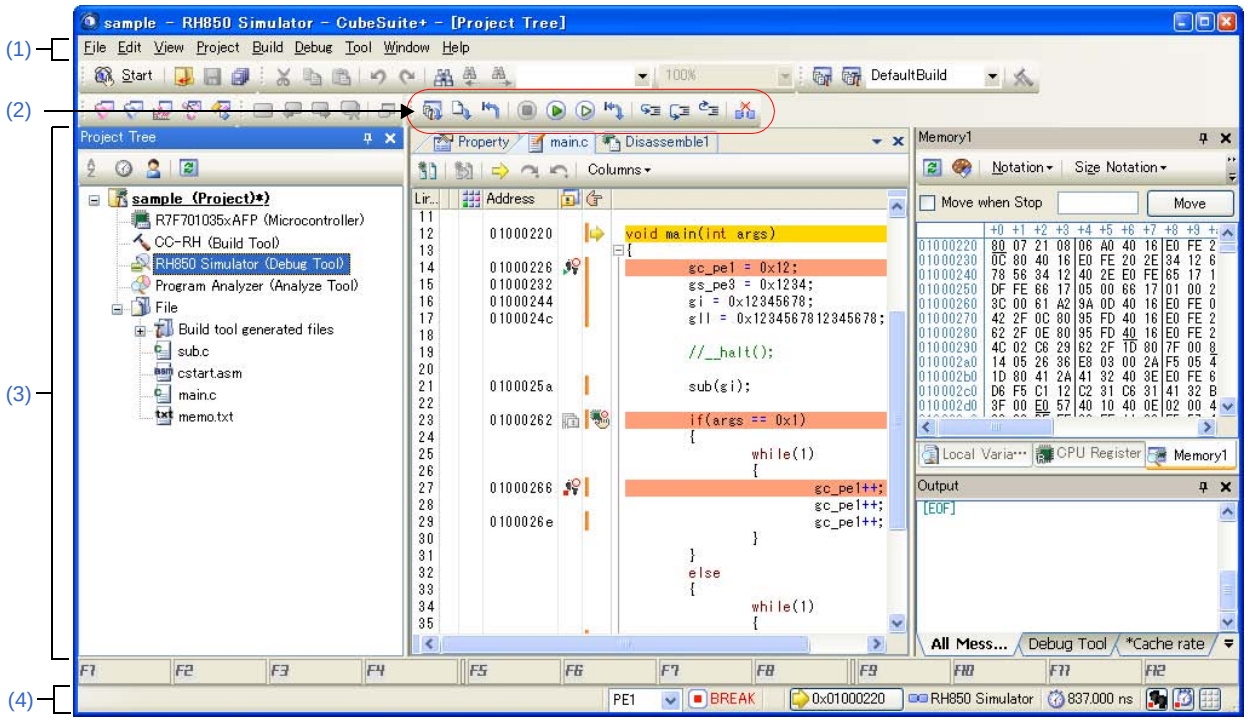
Window/Panel/Dialog Box Name	Description
Main window	Controls the program execution. Various windows, panels and dialog boxes can be opened from this window.
Debug Manager panel	Selects a core (PE*n*) to be debugged and displays the core status.
Project Tree panel	Selects the debug tool to use.
Property panel	Displays detailed information on the debug tool currently selected in the Project Tree panel, and enables the settings of the tool to be changed.
Editor panel	Enables text files to be viewed and edited, and is used to execute source level debug.
Memory panel	Displays and modifies memory values.
Disassemble panel	Displays the results of memory value disassemble and is used to execute line assemble and instruction level debug.
CPU Register panel	Displays the contents of CPU registers, and modifies register values.
IOR panel	Displays and modifies I/O register values.
Local Variables panel	Displays and modifies local variables.
Watch panel	Displays and modifies registered watch-expression values.
Call Stack panel	Displays call stack information on function calls.
Trace panel	Displays trace data acquired from the debug tool.
Events panel	Displays detailed information on set events, switches the events between enabled and disabled, or deletes them.
Output panel	Displays messages output from the build tool/debug tool/plug-ins, or the results of batch searches carried out using the Find and Replace dialog box.
Memory Mapping dialog box	Displays the memory mapping.
Download Files dialog box	Selects files to be downloaded and sets the download conditions.
Flash Options Setting dialog box [Full-spec emulator][E1][E20]	Configures options for the flash memory.
Select Priority Boot Loader Project dialog box	Selects the one that you wish to take precedence over other projects in stand-along debugging.
Text Edit dialog box	Inputs and modifies character strings.
Action Events dialog box	Sets action events.
Encoding dialog box	Selects a file-encoding.
Save Settings dialog box	Specifies the encoding and the new line code of the file being edited.
Bookmarks dialog box	Displays and deletes bookmarks.

Window/Panel/Dialog Box Name	Description
Column Number Settings dialog box	Specifies the number of view columns of memory values on the Memory panel.
Address Offset Settings dialog box	Specifies an offset value for the address display on the Memory panel.
Memory Initialize dialog box	Initializes memory.
Memory Search dialog box	Searches memory.
Print Address Range Settings dialog box	Sets the address range to print the contents of the Disassemble panel.
Print Preview window	Previews the source file before printing.
Trace Search dialog box	Searches trace data.
Scroll Range Settings dialog box	Sets the scroll range for the Memory panel/Disassemble panel.
Go to Line dialog box	Moves the caret to the specified line.
Jump to Function dialog box	Selects a function to which the caret moves.
Go to the Location dialog box	Moves the caret to the specified position.
Data Save dialog box	Saves the settings and other data displayed in the respective windows/panels/dialog boxes or saves upload data.
Progress Status dialog box	Displays the progress of the processing being executed.
Option dialog box	Makes settings for various environments.
Select Download File dialog box	Selects files to be downloaded.
Open Watch Expression Data File dialog box	Selects a file for importing watch-expressions.
Open File dialog box	Selects files to be opened.
Save As dialog box	Saves the contents of the panel into a specified file.
Select Data Save File dialog box	Selects the file to save data.
Open Option Setting File dialog box	Selects the option setting file to import to the Option dialog box.
Save Option Setting File dialog box	Saves the set contents of the Option dialog box to a option setting file.

Main window

This window is automatically opened when CubeSuite+ is started up.
In this window, you can control the program execution and open panels for the debugging process.

Figure A-1. Main Window



This section describes the following.

- [How to open]
- [Description of each area]

[How to open]

- From the Windows [Start] menu, select [All Programs] >> [Renesas Electronics CubeSuite+] >> [CubeSuite+].

[Description of each area]

(1) Menubar

Menu items related to the debugging are described below.

Remark The items that can be selected in each menu can be customized using the User Setting dialog box.

(a) [View]

The [View] menu provides the following items and functions (default).

Debug Manager	Opens the Debug Manager panel This item is disabled when the selected microcontroller version does not support multi-core or when disconnected from the debug tool.

Watch	The following cascade menus are displayed to open the Watch panel. These items are disabled when disconnected from the debug tool.
Watch1	Opens the Watch panel (Watch1).
Watch2	Opens the Watch panel (Watch2).
Watch3	Opens the Watch panel (Watch3).
Watch4	Opens the Watch panel (Watch4).
Local Variable	Opens the Local Variables panel.
Call Stack	Opens the Call Stack panel.
Memory	The following cascade menus are displayed to open the Memory panel. These items are disabled when disconnected from the debug tool.
Memory1	Opens the Memory panel (Memory1).
Memory2	Opens the Memory panel (Memory2).
Memory3	Opens the Memory panel (Memory3).
Memory4	Opens the Memory panel (Memory4).
IOR	Opens the IOR panel. This item is disabled when disconnected from the debug tool.
CPU Register	Opens the CPU Register panel. This item is disabled when disconnected from the debug tool.
Trace	Opens the Trace panel. This item is disabled when disconnected from the debug tool.
Disassemble	The following cascade menus are displayed to open the Disassemble panel. These items are disabled when disconnected from the debug tool.
Disassemble1	Opens the Disassemble panel (Disassemble1).
Disassemble2	Opens the Disassemble panel (Disassemble2).
Disassemble3	Opens the Disassemble panel (Disassemble3).
Disassemble4	Opens the Disassemble panel (Disassemble4).
Event	Opens the Events panel. This item is disabled when disconnected from the debug tool.
Show Current PC Location	Displays the current PC position in the Editor panel. This item is disabled when disconnected from the debug tool.
Back to Last Cursor Position	Goes back to the position before jumping (see "(7) Jump to functions"/"(4) Move to the symbol defined location") to the defined location. This item is disabled when disconnected from the debug tool.
Forward to Next Cursor Position	Forwards to the position before operating [Back to Last Cursor Position].
Tag Jump	Jumps to the corresponding line/column in the corresponding file if the information of a file name/line number/column number exists in the line at the caret position on the Editor panel/Output panel (see "(8) Jump to a desired line (tag jump)").

(b) [Debug]

The [Debug] menu provides the following items and functions (default).

Download	Downloads the specified file(s) into the debug tool currently selected in the active project. If CubeSuite+ is disconnected from the debug tool at this time, it is automatically connected to the debug tool before a download is executed. This item is disabled during execution of a program or when [Build & Download] is under execution.
Build & Download	Executes the build of a project and downloads the file into the debug tool currently selected in the active project. If CubeSuite+ is disconnected from the debug tool at this time, it is automatically connected to the debug tool before a download is executed. However, when the build has failed, download will not be executed.
Connect to Debug Tool	Connects to the debug tool currently selected in the active project. This item is disabled while connected to the debug tool or if the version of compiler being used is not supported by CubeSuite+.
Upload...	Opens the Data Save dialog box to save the memory contents. This item is disabled during execution of a program, when [Build & Download] is under execution or when disconnected from the debug tool.
Disconnect from Debug Tool	Disconnects from the currently connected debug tool. This item is disabled when [Build & Download] is under execution or when disconnected from the debug tool.
Stop	Forcibly stops the program currently being executed. This item is disabled during non-execution of a program or when disconnected from the debug tool.
Go	Executes the program from the current PC position. Execution of the program will be stopped when the condition of a set break event is met. This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool.
Ignore break and go	Executes the program from the current PC position. Execution of the program continues, ignoring set break events and action events. This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool.
Step In	Executes the program step by step[Note] from the current PC position (Step in execution). However, in the case of a function call, the program is stopped at the beginning of the function having been called. This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool.
Step Over	Executes the program step by step[Note] from the current PC position (Step over execution). In the case of a function call by the jarl instruction, all the source lines/instructions in the function are treated as one step and executed until the position where execution returns from the function (step-by-step execution will continue until the same nest is formed as when the jarl instruction has been executed). In the case of an instruction other than jarl, operation is the same as when [Step In] is selected. This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool.
Return Out	Executes the program until execution returns from the current function (or returns to the calling function)[Note](Return out execution). This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool.

CPU Reset	Resets the CPU (does not execute a program) This item is disabled when [Build & Download] is under execution or when disconnected from the debug tool.
Restart	Resets the CPU and then executes the program from the reset address. This item is disabled when [Build & Download] is under execution or when disconnected from the debug tool.

Note Step execution can be carried out either in units of source lines or in units of instructions.
For details, see "2.8.3 Execute programs in steps".

(2) Debug toolbar

The debug toolbar includes the buttons that control the execution of programs.
The debug toolbar provides the following buttons and functions (default).

Remarks 1. The buttons on the toolbar can be customized using the User Setting dialog box. Furthermore, a new toolbar can be created using the same dialog box.

2. A Group of toolbar displayed can be selected with the context menu that is displayed by right-clicking on the toolbar.

	Executes the build of a project and downloads the file into the debug tool currently selected in the active project. If CubeSuite+ is disconnected from the debug tool at this time, it is automatically connected to the debug tool before a download is executed. However, when the build has failed, download will not be executed. The function of this item is the same as that of [Build & Download] in the [Debug] menu.
	Downloads the specified file(s) into the debug tool currently selected in the active project. If CubeSuite+ is disconnected from the debug tool at this time, it is automatically connected to the debug tool before a download is executed. This item is disabled during execution of a program or when [Build & Download] is under execution. The function of this item is the same as that of [Download] in the [Debug] menu.
	Resets the CPU (does not execute a program) This item is disabled when [Build & Download] is under execution or when disconnected from the debug tool. The function of this item is the same as that of [CPU Reset] in the [Debug] menu.
	Forcibly stops the program currently being executed. This item is disabled during non-execution of a program or when disconnected from the debug tool. The function of this item is the same as that of [Stop] in the [Debug] menu.
	Executes the program from the current PC position. Execution of the program will be stopped when the condition of a set break event is met. This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool. The function of this item is the same as that of [Go] in the [Debug] menu.
	Executes the program from the current PC position. Execution of the program continues, ignoring set break events and action events. This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool. The function of this item is the same as that of [Ignore break and go] in the [Debug] menu.

	Resets the CPU and then executes the program from the reset address. This item is disabled when [Build & Download] is under execution or when disconnected from the debug tool. The function of this item is the same as that of [Restart] in the [Debug] menu.
	Executes the program step by step[Note] from the current PC position (Step in execution). However, in the case of a function call, the program is stopped at the beginning of the function having been called. This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool. The function of this item is the same as that of [Step In] in the [Debug] menu.
	Executes the program step by step[Note] from the current PC position (Step over execution). In the case of a function call by the jarl instruction, all the source lines/instructions in the function are treated as one step and executed until the position where execution returns from the function (step-by-step execution will continue until the same nest is formed as when the jarl instruction has been executed). In the case of an instruction other than jarl, operation is the same as when the button is clicked. This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool. The function of this item is the same as that of [Step Over] in the [Debug] menu.
	Executes the program until execution returns from the current function (or returns to the calling function)[Note] (Return out execution). This item is disabled during execution of a program, when [Build & Download] is under execution, or when disconnected from the debug tool. The function of this item is the same as that of [Return Out] in the [Debug] menu.
	Disconnects from the currently connected debug tool. This item is disabled when [Build & Download] is under execution or when disconnected from the debug tool.

Note Step execution can be carried out either in units of source lines or in units of instructions.
For details, see "2.8.3 Execute programs in steps".

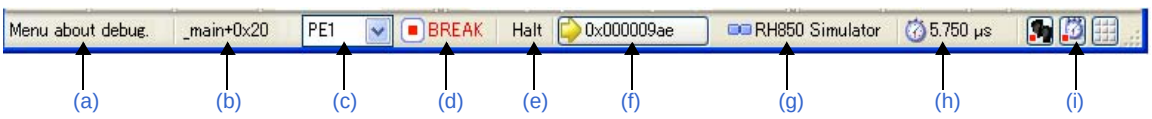
(3) Panel display area

This area displays the various panels.
For details on the display content, see the sections describing the individual panels.

(4) Statusbar

Statusbar displays the following items of information.

Figure A-2. Statusbar



(a) Status message

This area displays the following messages and other information.

- A brief explanation of the selected menu item
- A message reporting that an invalid value has been input in the panel/dialog
- A message reporting that the specified character string has not been found as a result of a search using the Find and Replace dialog box

- A statement of the cause of the break when a break has occurred (see "2.9 Stop Programs (Break)")

(b) Focus panel status information

This area displays status information on the panel currently having the focus.
Note that nothing is displayed here for a panel that has no status information.

(c) Selection of debug target core

This area is used to select a core (PE*n*) to be debugged (see "2.7 Select a Core (PE)").
Note that nothing is displayed here when the selected microcontroller version does not support multi-core or when disconnected from the debug tool.

(d) Running state

This area displays the state of the program with the following icons and character strings.
Note that nothing is displayed here when the debug tool is not connected.

State of Program	Displayed Content
Under execution	RUN
Now halted	BREAK
Step execution in progress	STEP

(e) CPU status

This area displays the current CPU status of the debug tool. When there is the possibility that the CPU is in two or more statuses, the corresponding display contents are displayed separated by "&".
Note that nothing is displayed here when the debug tool is not connected.

Debug Tool	Displayed Content	CPU Status
Full-spec emulator E1/E20	Halt	In HALT mode
	Stop	In STOP mode
	Reset	In reset state
	Pow Off	Power not supplied to the target
Simulator	Halt	In HALT mode
	StopIdle	In STOP mode or IDLE mode
	Reset	In reset state

(f) Current PC position

This area displays the current PC position with a hexadecimal value. When this area is clicked, the caret moves to the current PC position on the Editor panel.
In addition, when the mouse pointer is placed over this area, a pop-up window appears to display the following information: "Current PC: 0x *current PC value* (*source name*#*line count*[Note])".
Note that nothing is displayed here when the debug tool is not connected.

Note "*symbol name+offset value*" is displayed when acquisition of information is impossible.

Remark "Running" is displayed in this area during execution of a program.

(g) Connection state

This area displays the current state of connection with the debug tool using the following icons and character strings.

Connection State	Displayed Content
Connected	*Debug tool name*
Disconnected	DISCONNECTED

(h) Run-Break Timer measurement result

This area displays the result of measurement by the Run-Break Timer event (the unit of value used differs depending on the measurement amount). See "2.13.1 Measure execution time until stop of the execution". Note that nothing is displayed here when the debug tool is not connected.

Condition	Displayed Content
Un-measuring	Not measured
Under measurement	Measuring
When a timer measurement overflow has occurred	OVERFLOW

(i) Debug tool state

This area displays the current state of debug tool's functions using the following icons and character strings. Note that nothing is displayed here when the debug tool is not connected.

Function	Use		Not Use
	Being Executed	Stopped	
Trace			
Timer			
Coverage			

Remark [Simulator]

When the program is halted, clicking the appropriate icon enables the state to be switched between "Use" and "Not use". The result of switching will be reflected in the setting of the [Use trace function]/[Use timer function]/[Use coverage function] property in the [Trace]/[Timer]/[Coverage] category on the [Debug Tool Settings] tab of the Property panel.

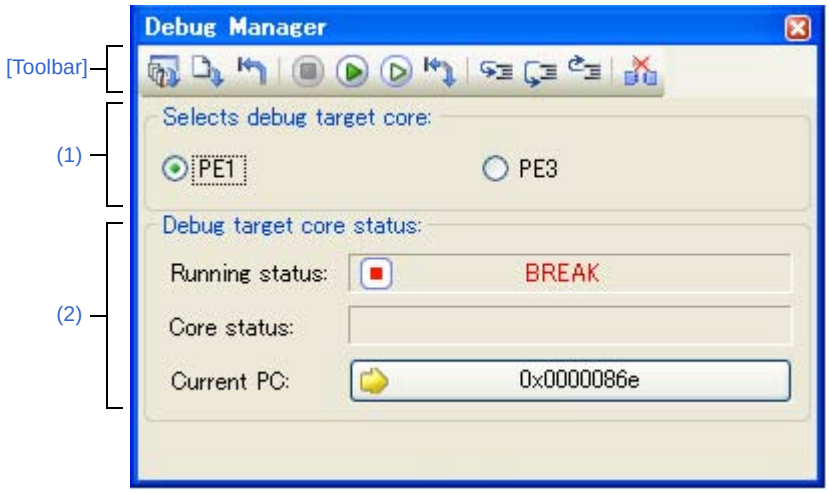
Debug Manager panel

When the selected microcontroller is the multi-core product, this panel is used to select a core (PE: Processer Element) to be debugged and display the core status (see "2.7 Select a Core (PE)").

This panel appears only when connected to the debug tool.

Caution This panel cannot be opened when the selected microcontroller is the single-core product.

Figure A-3. Debug Manager Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]

[How to open]

- From the [View] menu, select [Debug Manager].

[Description of each area]

(1) [Selects debug target core] area

Select a core (PE*n*) to be debugged with a option button.

Note that this area becomes invalid during execution of a program.

Remark You can also select a core to be debugged on the statusbar in the Main window.

(2) [Debug target core status] area

This area displays the status of the core currently being selected.

Remark You can also confirm the information displayed in this area on the statusbar in the Main window.

(a) [Running status]

Displays the current state of the program with the following icons and character strings.

State of Program	Displayed Content
Running	RUN
Stopped	BREAK
In step execution	STEP

(b) [Core status]

Displays the current core statuses of the debug tool. When there is the possibility that the core is in two or more statuses, the corresponding display contents are displayed separated by "&".

Debug Tool	Displayed Content	Core Status
Full-spec emulator E1/E20	Halt	In HALT mode
	StopIdle	In Hardware STOP/Software STOP/IDLE mode
	Hold	In bus hold
	Wait	In wait state
	Reset	In reset state
	Pow Off	Power not supplied to the target
Simulator	Halt	In HALT mode
	StopIdle	In STOP/IDLE mode
	Reset	In reset state

(c) [Current PC]

Displays the current PC position with a hexadecimal value. When this button is clicked, the caret moves to the current PC position on the Editor panel.

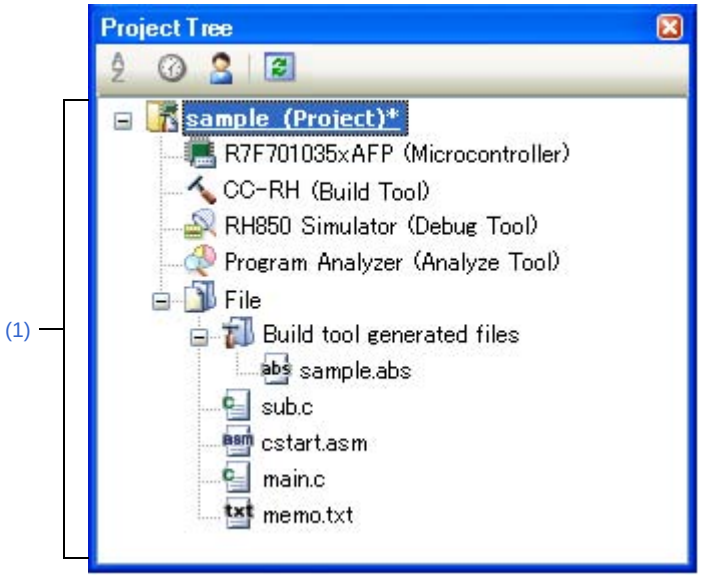
[Toolbar]

The function of this toolbar is the same as that of the Debug toolbar on the Main window. For details on the function of each button, see "(2) Debug toolbar".

Project Tree panel

This panel is used to display the project components (Microcontroller, Build Tool, Debug Tool, etc.) in a tree structure. On this panel, you can select or change the debug tool to use.

Figure A-4. Project Tree Panel



The following items are explained here.

- [How to open]
- [Description of each area]
- [Context menu]

[How to open]

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

[Description of each area]

(1) Project tree area

Project components are displayed in tree view with the following given node.

Node	Description
Microcontroller type Debug tool name (Debug tool)	- *Microcontroller type*: The selected microcontroller type is displayed. - *Debug tool name*: The debug tool (Full-spec emulator, E1, E20, or Simulator) currently being used in the project is displayed[Note]. Simulator is selected when a new project is created.

Note The selectable debug tools differ depending on the microcontroller selected in the project.

Select the debug tool node to configure with the Property panel. If the Property panel is not being opened, double-click the node to open the corresponding Property panel.

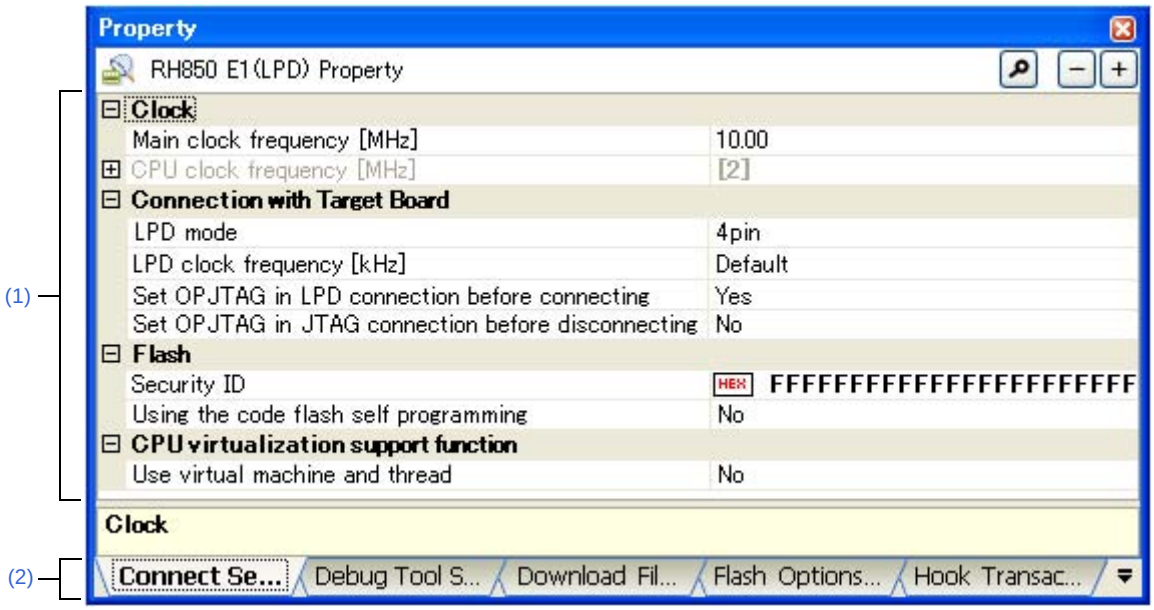
[Context menu]

Using Debug Tool	The following cascade menus are displayed to select the debug tool to use. Note that the debug tools displayed in this menu differ depending on the microcontroller selected in the project.
Device name Full-spec emulator	Uses Full-spec emulator as the debug tool.
Device name E1(LPD)	Uses E1 in LPD communication mode as the debug tool.
Device name E20(LPD)	Uses E20 in LPD communication mode as the debug tool.
Device name Simulator	Uses Simulator as the debug tool.
Property	Displays the selected category node's property in the Property panel.

Property panel

This panel is used to display and set the debug tool operation environment that is selected in the Project Tree panel.

Figure A-5. Property Panel (When E1 Is Selected)



This section describes the following.

- [How to open]
- [Description of each area]
- [[Edit] menu (Property panel-dedicated items)]
- [Context menu]

[How to open]

- On the Project Tree panel, select the [*Microcontroller type Debug tool name* (Debug Tool)] node to use, and then select [Property] from the [View] menu or the context menu.
- On the Project Tree panel, double-click the [*Microcontroller type Debug tool name* (Debug Tool)] node to use.

Remark If this panel has been opened, the detailed information on the debug tool is displayed by selecting the [*Microcontroller type Debug tool name* (Debug Tool)] node on the Project Tree panel.

[Description of each area]

(1) Detailed information display/change area

In this area, the detailed information on the debug tool that is selected in Project Tree panel is displayed by category in the list. Also, you can directly change its settings.

The ⊟ mark is indicates all the items in the category are expanded. The ⊞ mark indicates all the items are collapsed. You can expand/collapse the items by clicking these marks or double-clicking the category name.

Note that only the hexadecimal number is allowed in the text box if the HEX mark is displayed in the property configuration area.

For details on the information/how to setup in the category and property items contained in it, see the section explaining the corresponding tab.

(2) Tab selection area

Categories for the display of the detailed information are changed when each tab is selected.
In this panel, following tabs are contained (see the section explaining each tab for details on the display/setting on the tab).

- [Connect Settings] tab
- [Debug Tool Settings] tab
- [Download File Settings] tab
- [Flash Options Settings] tab [Full-spec emulator][E1][E20]
- [Hook Transaction Settings] tab

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

Undo	Undoes the latest property value editing being done.
Cut	Deletes the selected character string(s) and copies them to the clipboard while editing the property value.
Copy	Copies the contents of the selected range to the clipboard as character string(s).
Paste	Pastes the contents of the clipboard to the property value while editing the property value.
Delete	Deletes the selected character string(s) while editing the property value.
Select All	Selects all the character strings in the selected property while editing the property value.
Find...	Opens the Find and Replace dialog box with selecting [Quick Find] tab.

[Context menu]

[While not editing the property value]

Reset to Default	Restores the selected setting of the property item to default value.
Reset All to Default	Restores all the selected settings of the property items on the tab to default value.

[While editing the property value]

Undo	Undoes the latest property value editing being done.
Cut	Deletes the selected character string(s) and copies them to the clipboard while editing the property value.
Copy	Copies the contents of the selected range to the clipboard as character string(s).
Paste	Pastes the contents of the clipboard to the property value while editing the property value.
Delete	Deletes the selected character string(s) while editing the property value.
Select All	Selects all the character strings in the selected property while editing the property value.

[Connect Settings] tab

This tab is used to display the detailed information categorized by the following and the configuration can be changed.

(1) [Clock]
(2) [Connection with Target Board] [Full-spec emulator][E1][E20]
(3) [Flash] [Full-spec emulator][E1][E20]
(4) [Configuration] [Simulator]
(5) [CPU virtualization support function]

Figure A-6. Property Panel: [Connect Settings] Tab [Full-spec emulator]

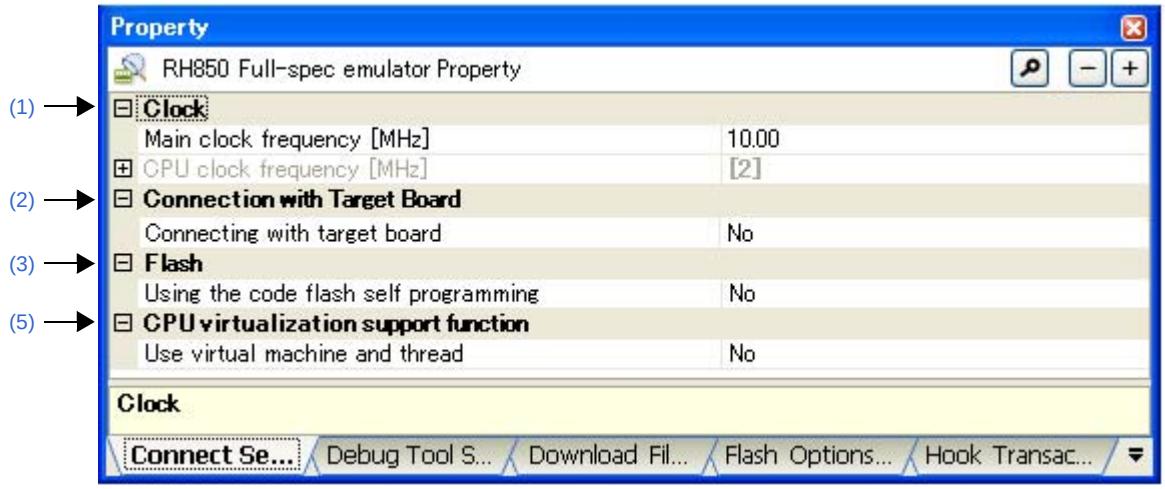


Figure A-7. Property Panel: [Connect Settings] Tab [E1][E20]

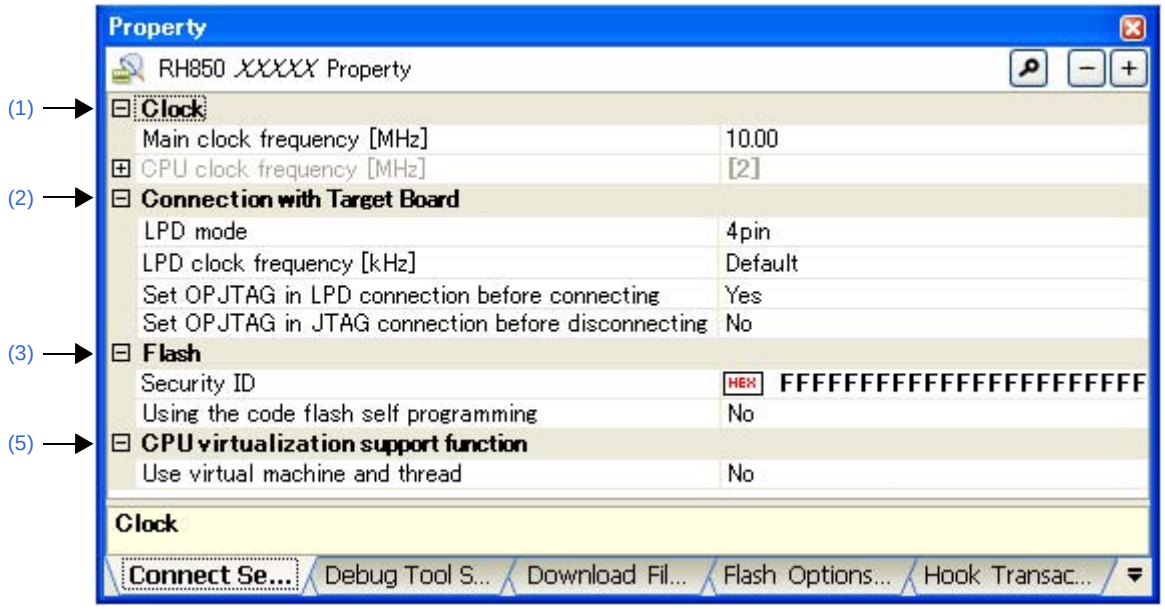
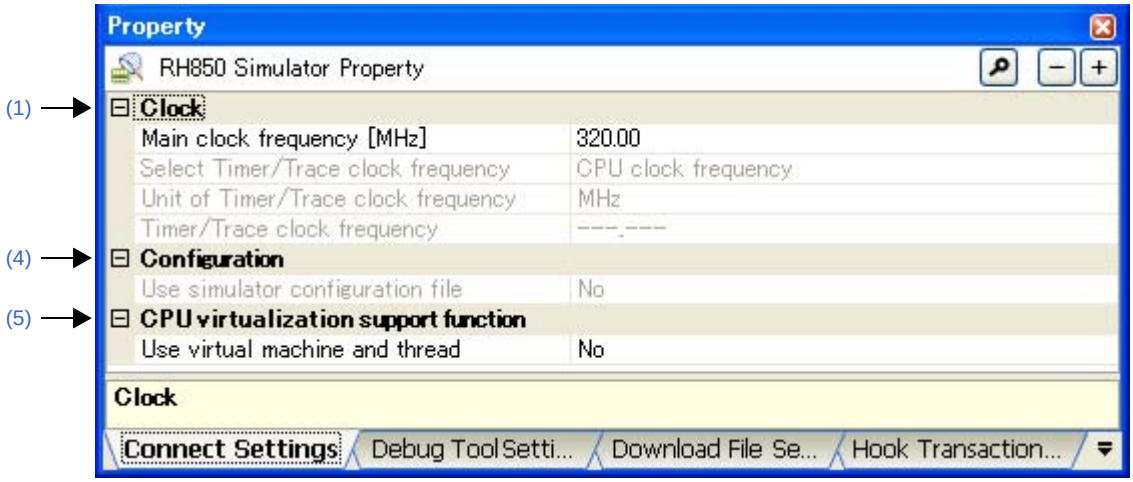


Figure A-8. Property Panel: [Connect Settings] Tab [Simulator]



[Description of each category]

(1) [Clock]

The detailed information on clocks is displayed and its configuration can be changed.

Main clock frequency [MHz]	Specify the main clock frequency (before multiplication) in MHz unit.	
	Default	**[Full-spec emulator][E1][E20]** 10.00 **[Simulator]** 320.00
	Modifying	Select from the drop-down list or directly enter from the keyboard.
	Available values	- Either one of the following from the drop-down list **[Full-spec emulator][E1][E20]** 10.00, 20.00 (unit: MHz) **[Simulator]** 1.00, 2.00, 3.00, 3.57, 4.00, 4.19, 4.91, 5.00, 6.00, 7.20, 8.00, 8.38, 9.60, 10.00, 12.00, 16.00, 20.00, 25.00, 30.00, 32.00, 33.33, 34.00, 40.00, 48.00, 50.00, 64.00, 80.00, 160.00, 240.00, 320.00 (unit: MHz) - Directly enter the numbers ranged below 0.001 to 999.999 (unit: MHz)
CPU clock frequency[MHz] **[Full-spec emulator][E1][E20]**	Specify the CPU clock frequency for each core. The CPU clock frequency for each core can be specified with subproperties of this property. The CPU clock frequency is used to convert the time stamp information for a trace to an actual time. The number of subproperties displayed differs with the selected microcontroller.	
Core name (Subproperty) **[Full-spec emulator][E1][E20]**	Displays the name of the core incorporated in the selected microcontroller.	
	Default	Depends on the selected microcontroller
	Modifying	Changes not allowed

CPU clock frequency (Subproperty) **[Full-spec emulator] [E1][E20]**	Specify the CPU clock frequency of the *core name*.	
	Default	Depends on the selected microcontroller
	Modifying	Directly enter from the keyboard.
	Available values	0.001 to 999.999 (unit: MHz)
Select Timer/Trace clock frequency **[Simulator]**	Displays the clock frequency for using timer/trace function.	
	Default	CPU clock frequency
	Modifying	Changes not allowed
Unit of Timer/Trace clock frequency **[Simulator]**	Displays the unit of the clock frequency for using timer/trace function.	
	Default	MHz
	Modifying	Changes not allowed
Timer/Trace clock frequency **[Simulator]**	Displays the value of the clock frequency for using timer/trace function. Note, however, that "---_---" is displayed while disconnected from the debug tool.	
	Default	320.00
	Modifying	Changes not allowed

(2) [Connection with Target Board] [Full-spec emulator][E1][E20]

The detailed information on the connection to the target board is displayed and its configuration can be changed.

Caution Properties in this category cannot be changed when CubeSuite+ is connected to the debug tool.

Connecting with target board **[Full-spec emulator]**	Select whether the target board is connected to Full-spec emulator.		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Target board is connected.
		No	Target board is not connected.
LPD mode **[E1][E20]**	Select LPD communication mode to be used.		
	Default	Depends on the selected microcontroller.	
	Modifying	Select from the drop-down list.	
	Available values	Depends on the selected microcontroller.	
Baud rate[kbps] **[E1][E20]**	Select the baud rate for LPD communication. This property appears only when the [LPD mode] property is set to [1pin].		
	Default	500	
	Modifying	Select from the drop-down list.	
	Available values	500, 1000, 2000 (unit: Kbps)	

LPD clock frequency [kHz] **[E1][E20]**	Specify the clock frequency for LPD communication. When [Default] is selected, the default value specific to the microcontroller is used in connection to the target board. This property appears only when the [LPD mode] property is set to [4pin].		
	Default	Blank	
	Modifying	Select from the drop-down list.	
	Available values	Default, 5500, 11000 (unit: kHz)	
Set OPJTAG in LPD connection before connecting **[E1][E20]**	Select whether to start up the microcontroller in serial programming mode upon connection to the debug tool and change the option byte settings to select LPD connection.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Starts up the microcontroller in serial programming mode upon its connection to CubeSuite+. The debug tool then checks the OPJTAG byte and, if LPD is not selected, changes the setting to select LPD. After that, the microcontroller enters debugging mode (default).
		No	Starts up the microcontroller in debugging mode upon its connection to CubeSuite+. The debug tool then checks OPJTAG and, if LPD is not selected, shows a message dialog box.
Set OPJTAG in JTAG connection before disconnecting **[E1][E20]**	Select whether to change the option byte settings to select JTAG connection before disconnection of the debug tool.		
	Default	No	
	Modifying	Select from the drop-down list. Note that changes can be made only when the [Set OPJTAG in LPD connection before connecting] property is set to [Yes].	
	Available values	Yes	Changes the option byte settings to select JTAG connection before disconnection of the debug tool.
		No	Does not change the option byte settings before disconnection of the debug tool.

(3) [Flash] [Full-spec emulator][E1][E20]

The detailed information on the flash memory writing is displayed and its configuration can be changed.

Caution Properties in this category cannot be changed when CubeSuite+ is connected to the debug tool.

Security ID **[E1][E20]**	Specify the key code for ID authentication when reading the code from the internal ROM or internal flash memory[Note].	
	Default	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
	Modifying	Directly enter from the keyboard.
	Available values	32 digits hexadecimal number (16 bytes)

Using the code flash self programming	Select whether to rewrite the code flash by using the flash self library of the flash self programming function.		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Rewrites the code flash. If [Yes] is selected, the code flash will not be cashed.
		No	Does not rewrite the code flash.

Note For details on a key code for ID authentication, see User's Manual of the emulator used.

(4) [Configuration] [Simulator]

The property in this category is always disabled.

(5) [CPU virtualization support function]

The property in this category is always disabled.

[Debug Tool Settings] tab

This tab is used to display the detailed information categorized by the following and the configuration can be changed.

(1) [Memory]
(2) [Access Memory While Running]
(3) [Set Event While Running] [Full-spec emulator][E1][E20]
(4) [Break] [Full-spec emulator][E1][E20]
(5) [Trace]
(6) [Timer] [Simulator]
(7) [Mask for Input Signal] [Full-spec emulator][E1][E20]
(8) [Coverage] [Simulator]
(9) [Simulator GUI] [Simulator]

Figure A-9. Property Panel: [Debug Tool Settings] Tab [Full-spec emulator]

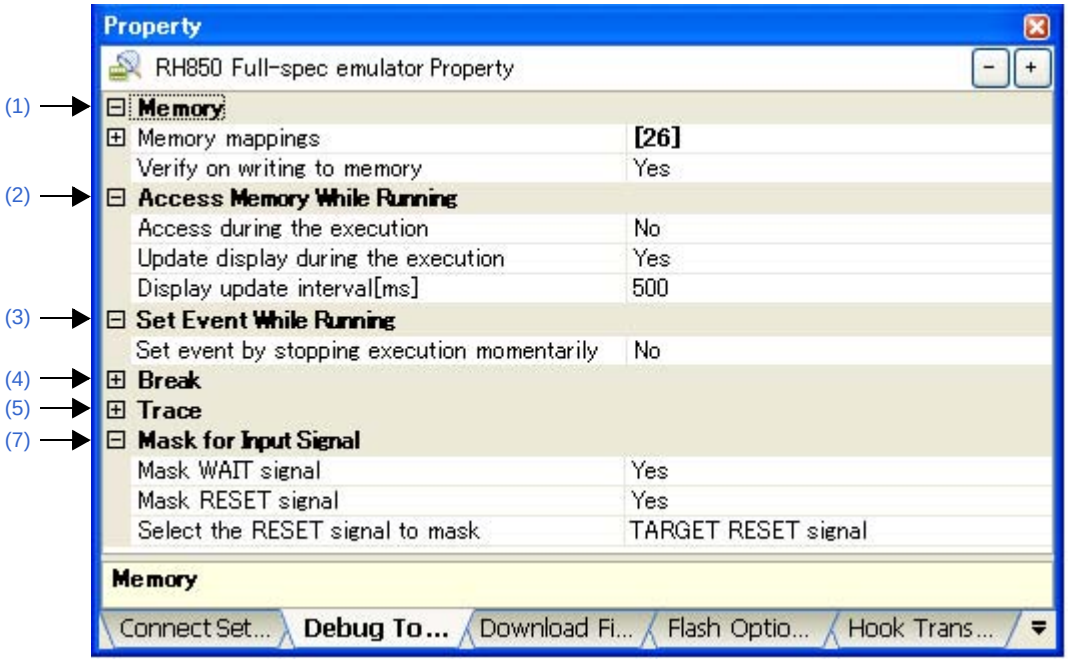


Figure A-10. Property Panel: [Debug Tool Settings] Tab [E1][E20]

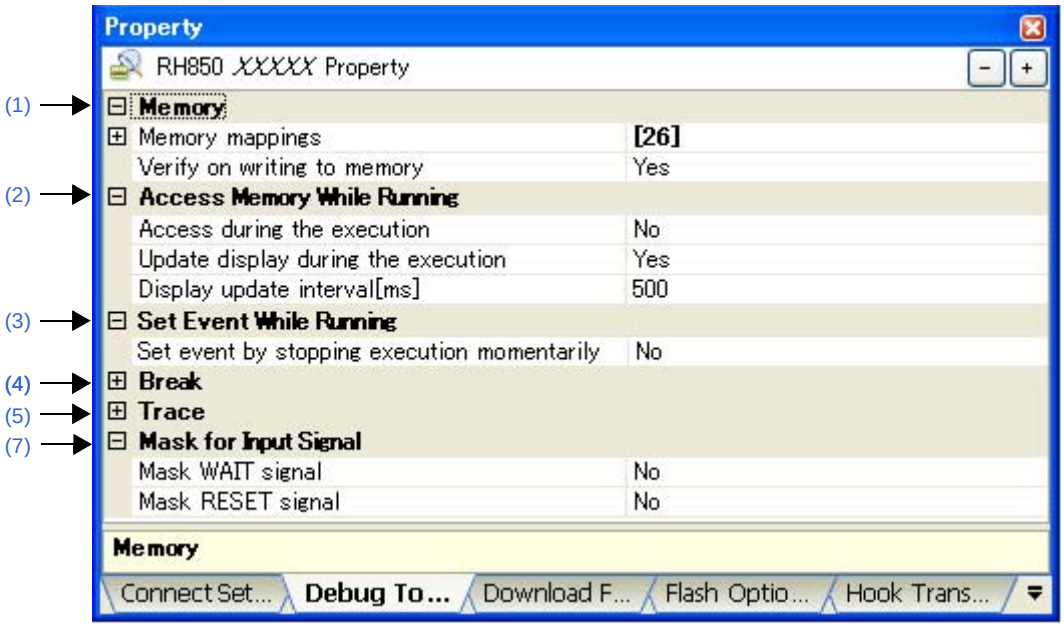
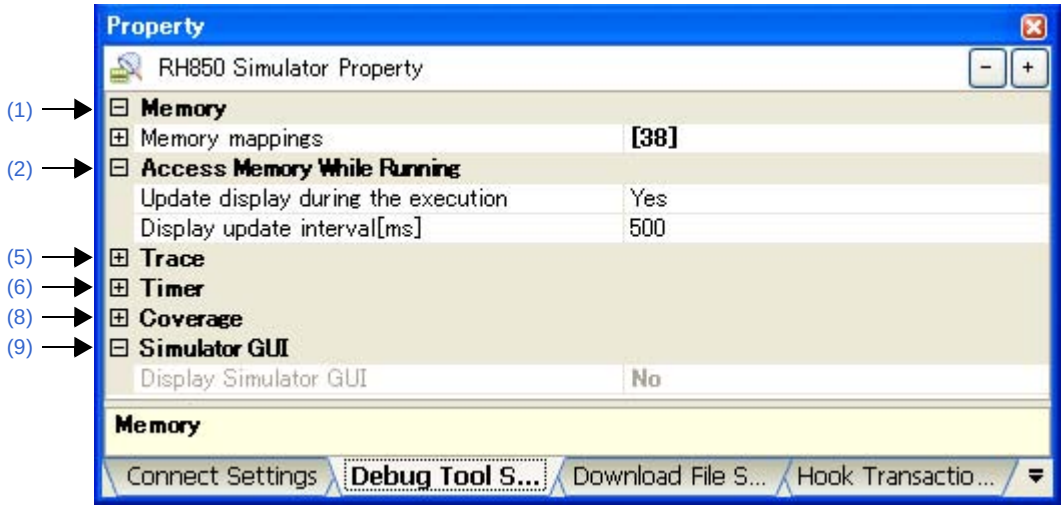


Figure A-11. Property Panel: [Debug Tool Settings] Tab [Simulator]



[Description of each category]

(1) [Memory]

The detailed information on memories is displayed and its configuration can be changed.

Memory mappings	The state of memory mapping is displayed for each type of memory area[Note].	
	Default	[*Sum total by microcontroller's inherent type of memory mapped area*]
	Modifying	Changes not allowed
	Displayed Content	Displays the memory mapping status for each type of memory area. The detailed information of the memory type, start address and end address is displayed by clicking the "+" mark of each memory type.

Verify on writing to memory **[Full-spec emulator]** **[E1][E20]**	Select whether to perform a verify check when the memory value is initialized.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Executes the verify check.
		No	Does not execute the verify check.

Note The type is of the memory mapping area registered in the device file.

(2) [Access Memory While Running]

The detailed information on memory accesses while executing a program (real-time display update function: see "(4) Display/modify the memory contents during program execution") is displayed and its configuration can be changed.

Access during the execution **[Full-spec emulator]** **[E1][E20]**	Select whether to allow access to the internal RAM area during execution of a program.		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Accesses to the internal RAM area during execution of a program.
		No	Does not access to the internal RAM area during execution of a program.
Update display during the execution	Select whether to update the display in the Watch panel/Memory panel during a program execution.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Updates the display during program execution.
		No	Does not update the display during program execution.
Display update interval[ms]	Specify the interval in 100ms unit to update the contents in the Watch panel/Memory panel display while executing a program. This property appears only when the [Update display during the execution] property is set to [Yes].		
	Default	500	
	Modifying	Directly enter from the keyboard.	
	Available values	Integer number between 100 and 65500 (rounding up the fractions less than 100 ms).	

(3) [Set Event While Running] [Full-spec emulator][E1][E20]

The detailed information on the function of the event setting during program execution is displayed and its configuration can be changed.

Set event by stopping execution momentarily	Select whether to forcibly pause the execution for events that cannot be set while executing the program or operating the tracer/timer. For details on the event types that are affected by this property, see "(2) Event types that can be set and deleted during execution".		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Sets these events by stopping the program execution or the tracer/timer operation momentarily.
		No	Does not allow to set these events during program execution or the tracer/timer operation.

(4) [Break] [Full-spec emulator][E1][E20]

The detailed information on break functions is displayed and its configuration can be changed.

Use software break	Select whether to use the Software break function [Full-spec emulator][E1][E20]Note.		
	Default	No	
	Modifying	Select from the drop-down list. Note that changes can be made only when program execution is halted.	
	Available values	Yes	Uses the software break function.
		No	Does not use the software break function.
First using type of breakpoint	Select the type of the breakpoint to use with priority when setting it at the source line or the execution address with a one click operation of the mouse in the Editor panel/Disassemble panel.		
	Default	Software break	
	Modifying	Select from the drop-down list.	
	Available values	Software break	Sets software breakpoint with priority.
		Hardware break	Sets hardware breakpoint with priority.
Stop emulation of peripherals when stopping	Select whether to terminate the peripheral emulation while stopping the program execution (Peripheral Break).		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Terminates the peripheral emulation.
		No	Does not terminate the peripheral emulation.

Note If this property is set to [No] after you have used the software break function, all software break events and Printf events that have been set will be disabled. Selecting [Yes] in this state does not automatically restore the events, so you will need to manually enable them.

(5) [Trace]

The detailed information on trace functions is displayed and its configuration can be changed (see "2.12.1 Configure the trace operation").

Cautions 1. [Full-spec emulator][E1][E20]
Properties in this category cannot be changed during program execution.

2. [E1][E20]

If the trace function is not mounted on the microcontroller used, all properties in this category become unchangeable after connecting to the debug tool (the trace function cannot be used).

Select trace data **[Full-spec emulator]** **[E1][E20]**	Select the type of the trace data to be acquired[Note 1].		
	Default	Branch PC + Data Access	
	Modifying	Select from the drop-down list.	
	Available values	Branch PC	PC values for source/destination instructions of branching during program execution are collected as trace data.
		Data Access	Data information on access-related events that occurred during program execution are collected as trace data.
		PC + Data Access	PC values for source/destination instructions of branching during program execution, as well as data information on access-related events that occurred are collected as trace data.
Trace priority **[Full-spec emulator]** **[E1][E20]**	Select which item should be given priority when collecting the trace data[Note 1].		
	Default	Speed priority	
	Modifying	Select from the drop-down list.	
	Available values	Speed priority	Traces giving priority to the real-time performance.
		Data priority	Traces after stopping the execution pipeline of the CPU temporarily so that no data is missed.
Use trace function **[Simulator]**	Select whether to use the trace function[Note 2].		
	Default	No	
	Modifying	Select from the drop-down list. Note that changes can be made only when program execution is halted.	
	Available values	Yes	Uses trace functions.
		No	Does not use trace functions.
Clear trace memory before running	Select whether to clear the trace memory before executing.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Clears the trace memory.
		No	Does not clear the trace memory.
Operation after trace memory is full	Select the operation after the trace memory is full with the collected trace data[Note 1].		
	Default	Non stop and overwrite to trace memory	
	Modifying	Select from the drop-down list.	
	Available values	Non stop and overwrite to trace memory	Continues overwriting trace data even after trace memory is used up.
		Stop trace[Note 3]	Stops overwriting trace data when trace memory is used up (the program execution will not be stopped).
		Stop	Stops running the program and overwriting trace data when trace memory is used up.

Accumulate trace time **[Simulator]**	Select whether to display the accumulated tracing time in the Trace panel.		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Displays the accumulated tracing time.
		No	Displays the trace time with differential value.
Rate of frequency division of trace time tag **[Simulator]**	Select the frequency division ratio of the counter to be used for time tag display (the [Time] item in the Trace panel). Changing the frequency division ratio here changes the number of clocks necessary to count up a counter value which is displayed in the time tag.		
	Default	1/1	
	Modifying	Select from the drop-down list.	
	Available values	1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1K, 1/4K, 1/8K, 1/16K, 1/64K, 1/256K, 1/1M	
Trace range setting **[Full-spec emulator]** **[E1][E20]**	Select the range of trace data to be collected.		
	Default	Traces section	
	Modifying	Select from the drop-down list.	
	Available values	Traces section	Collects the execution history as trace data within the section specified with a trace start event and a trace end event.
		Traces out of range	Collects the execution history as trace data outside the range specified with a trace start event and a trace end event.
Trace memory size[frames] **[Full-spec emulator]** **[Simulator]**	Select the memory size for storing the trace data by the trace frame numbers[Note 1, 4].		
	Default	**[Full-spec emulator]** 8K **[Simulator]** 4K	
	Modifying	Select from the drop-down list.	
	Available values	**[Full-spec emulator]** 8K, 32K, 64K, 128K, 256K, 512K **[Simulator]** 4K, 8K, 12K, 16K, 20K, 24K, 28K, 32K, 36K, 40K, 44K, 48K, 52K, 56K, 60K, 64K, 128K, 192K, 256K, 320K, 384K, 448K, 512K, 576K, 640K, 704K, 768K, 832K, 896K, 960K, 1M, 2M, 3M	
Enable trace data complement **[Full-spec emulator]**	Select whether to enable complement display when displaying the collected trace data. By enabling complement display, instructions between branch instructions that cannot be traced by hardware can be displayed. This setting will be applied from the next acquisition of trace data.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Performs complementary display of trace data.
		No	Does not perform complementary display of trace data.

Trace target setting **[Full-spec emulator]** **[E1][E20]**	Select the core to be traced.		
	Default	Debug core only	
	Modifying	Select from the drop-down list.	
	Available values	Debug core only	Collects the trace data regarding the currently selected PE*n* (default).
		All core	Collects the trace data for all PEs.

Notes 1. [Full-spec emulator][E1][E20]
The trace memory is cleared when you change the setting of this property.

2. This property is automatically set to [Yes] when selecting [Start Tracing]/[Stop Tracing] from the context menu in the Editor panel/Disassemble panel.

3. This item is not displayed when the [Trace priority] property is set to [Data priority].

4. The trace frame is a unit of trace data. Each fetch/write/read uses one trace frame.

(6) [Timer] [Simulator]

The detailed information on timer functions is displayed and its configuration can be changed.

Use timer function	Select whether to use the timer function.		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Uses timer functions.
		No	Does not use timer functions.

(7) [Mask for Input Signal] [Full-spec emulator][E1][E20]

The detailed information on the masking input signal is displayed and its configuration can be changed.

Mask WAIT signal	Select whether to mask WAIT signal to prevent the signal input to emulators.		
	Default	**[Full-spec emulator]** No **[E1][E20]** Yes	
	Modifying	Select from the drop-down list[Note].	
	Available values	Yes	Masks WAIT signal.
		No	Does not mask WAIT signal.
Mask RESET signal	Select whether to mask RESET signal to prevent the signal input to emulators.		
	Default	**[Full-spec emulator]** No **[E1][E20]** Yes	
	Modifying	Select from the drop-down list[Note].	
	Available values	Yes	Masks RESET signal.
		No	Does not mask RESET signal.

Select the RESET signal to mask	Select a RESET signal to be masked. This property appears only when the [Mask RESET signal] property is set to [Yes].		
	Default	**[Full-spec emulator]** TARGET RESET signal **[E1][E20]** TARGET RESET signal and INTERNAL RESET signal	
	Modifying	**[Full-spec emulator]** Select from the drop-down list. **[E1][E20]** Changes not allowed	
	Available values	TARGET RESET signal	Masks TARGET RESET signal.
		TARGET RESET signal and INTERNAL RESET signal	Masks TARGET RESET signal and INTERNAL RESET signal.

Note [Full-spec emulator]

When the [Connecting with target board] property in the [Connection with Target Board] [Full-spec emulator][E1][E20] category on the [Connect Settings] tab is set to [No], this property is fixed to [Yes] automatically after connecting to the debug tool (changes not allowed).

(8) [Coverage] [Simulator]

The detailed information on coverage functions is displayed and its configuration can be changed.

Use coverage function	Select whether to use the coverage function.		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Uses coverage functions
		No	Does not use coverage functions
Reuse coverage result	Select whether to load/save the coverage measurement result when connecting to or disconnecting from the debug tool. This property appears only when the [Use coverage function] property is set to [Yes].		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Loads/saves the coverage measurement result.
		No	Does not load/save the coverage measurement result.
Coverage area of measurement(1MBytes)	Specify the area that performs coverage measurement. Specify the start address of any 1 Mbyte space other than the internal ROM area. This property appears only when the [Use coverage function] property is set to [Yes].		
	Default	100000	
	Modifying	Directly enter from the keyboard.	
	Available values	Address without the address range of the internal ROM area (symbols cannot be used).	

(9) [Simulator GUI] [Simulator]

The detailed information on the Simulator GUI function is displayed and its configuration can be changed.

Caution If a microcontroller whose Simulator does not support peripheral function simulations is selected (i.e. the selected microcontroller supports only a instruction simulator), all properties in this category become invalid.

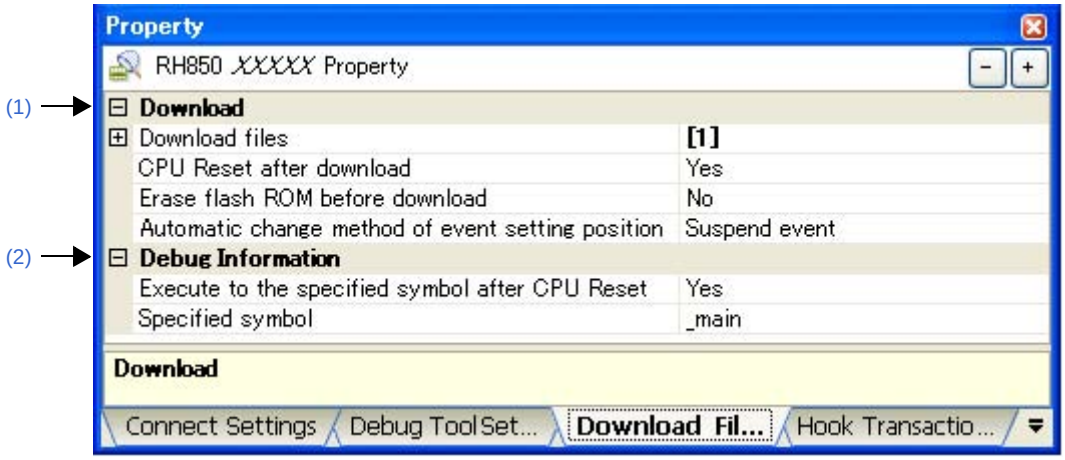
Display Simulator GUI	Select whether to display the Simulator GUI window to use the Simulator GUI function.		
	Default	Yes	
	Modifying	Select from the drop-down list. Note that changes can be made only when program execution is halted.	
	Available values	Yes	Displays the Simulator GUI window.
		No	Does not displays the Simulator GUI window.
Display Simulator GUI on top of other windows	Select whether to display the Simulator GUI window in the forefront when program execution starts. This property appears only when the [Display Simulator GUI] property is set to [Yes].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Displays it in the forefront.
		No	Does not display it in the forefront.

[Download File Settings] tab

This tab is used to display the detailed information categorized by the following and the configuration can be changed.
For details on the download function, see "2.5 Download/Upload Programs".

(1) [Download]
(2) [Debug Information]

Figure A-12. Property Panel: [Download File Settings] Tab



[Description of each category]

(1) [Download]

The detailed information on download is displayed and its configuration can be changed.

Download files	Specify the file to download[Note 1]. The names of files to be downloaded and the download conditions are listed in the lower area.		
	Default	[*Number of files to download*]	
	Modifying	Specify with the Download Files dialog box. The Download Files dialog box is opened by clicking the [...] button that appears at right edge of this field when you select this property (you cannot specify the file to download on this panel).	
CPU Reset after download	Select whether to reset the CPU after downloading.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Resets the CPU after downloading.
		No	Does not reset the CPU after downloading.
Erase flash ROM before download **[Full-spec emulator]** **[E1][E20]**	Select whether to erase the flash ROM before downloading.		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Erases the flash ROM before downloading.
		No	Does not erase the flash ROM before downloading.

Automatic change method of event setting position	Select how to perform the setting again if the file is downloaded again, and the location (address) set for the currently set event changes to midway in the instruction[Note 2].		
	Default	Suspend event	
	Modifying	Select from the drop-down list.	
	Available values	Move to the head of instruction	Sets the event to the top address of the instruction.
		Suspend event	Disables the event (suspended state).

Notes 1. Files specified as build targets in a main project or sub-project cannot be deleted from the target files to download (These files are automatically registered as download files by default).
See "Table 2-1. Downloadable File Formats" for downloadable file format.

2. This property setting works only for the location setting of events without the debug information. The location setting of events with the debug information is always moved to the beginning of the source text line.

(2) [Debug Information]

The detailed information on debugging is displayed and its configuration can be changed.

Execute to the specified symbol after CPU Reset	Select whether to execute the program to the specified symbol position after CPU reset.		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Executes the program to the specified symbol position after CPU reset.
		No	Does not execute the program after CPU reset.
Specified symbol	Specify the position at which the program is stop after CPU reset. This property appears only when the [Execute to the specified symbol after CPU Reset] property is set to [Yes].		
	Default	_main	
	Modifying	Directly enter from the keyboard.	
	Available values	Address expression from 0 to the *"end address of the address space"*.	

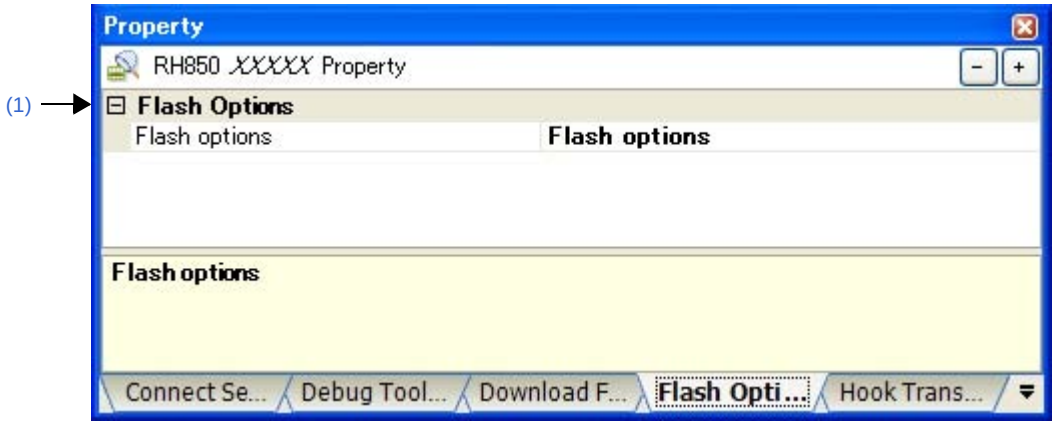
[Flash Options Settings] tab [Full-spec emulator][E1][E20]

This tab is used to configure options for the flash memory incorporated in the microcontroller.
Note that this tab appears only when the selected microcontroller supports the flash options.

(1) [Flash Options]

Cautions 1. You can configure options only while connected to the debug tool.
2. CPU reset may be generated automatically depending on the selected microcontroller when you change the configuration on this tab.

Figure A-13. Property Panel: [Flash Options Settings] Tab



[Description of each category]

(1) [Flash Options]

The detailed information on the flash options is displayed and its configuration can be changed.

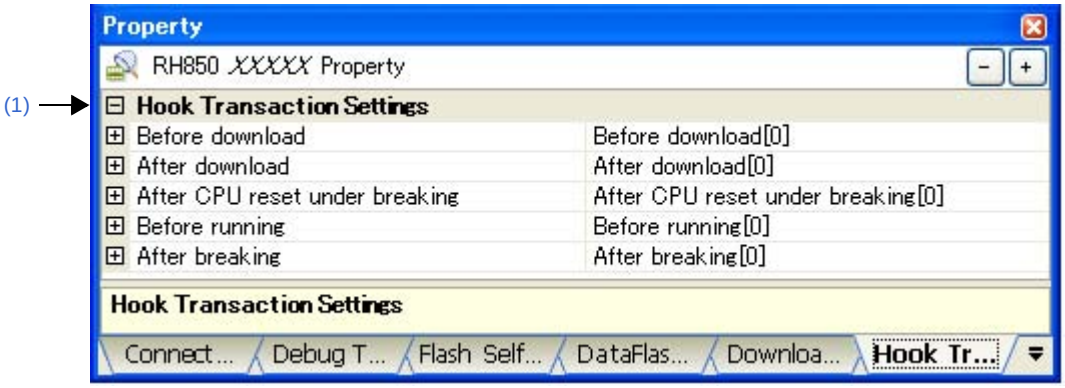
Flash options	Specify options for the flash memory.	
	Default	Flash options
	Modifying	Specify with the Flash Options Setting dialog box [Full-spec emulator][E1][E20]. The Flash Options Setting dialog box is opened by clicking the [...] button that appears at right edge of this field when you select this property (you cannot specify options for the flash memory on this panel). Note that the contents of options for the flash memory that have been specified are not displayed on this panel.

[Hook Transaction Settings] tab

This tab is used to display the detailed information categorized by the following and the configuration can be changed.
For details on the hook transaction, see "2.17 Use Hook Function".

(1) [Hook Transaction Settings]

Figure A-14. Property Panel: [Hook Transaction Settings] Tab



[Description of each category]

(1) [Hook Transaction Settings]

The detailed information on the hook transaction is displayed and its configuration can be changed.

Before download	Specify the process[Note] to proceed right before downloading the load module file.	
	Default	Before download[0] ("[]" is the current number of specified processes.)
	Modifying	Specify with the Text Edit dialog box. The Text Edit dialog box is opened by clicking the [...] button that appears at right edge of this field when you select this property (you cannot specify options for the flash memory on this panel).
	Available values	Up to 128 processes (one line in the Text Edit dialog box is equivalent to one processing) Note that up to 64 characters for one process can be specified.
After download	Specify the process[Note] to proceed right after downloading the load module file.	
	Default	Before download[0] ("[]" is the current number of specified processes.)
	Modifying	Specify with the Text Edit dialog box. The Text Edit dialog box is opened by clicking the [...] button that appears at right edge of this field when you select this property (you cannot specify options for the flash memory on this panel).
	Available values	Up to 128 processes (one line in the Text Edit dialog box is equivalent to one processing) Note that up to 64 characters for one process can be specified.

After CPU reset under breaking	Specify the process[Note] to proceed right after CPU reset during break.	
	Default	Before download[0] ("[]" is the current number of specified processes.)
	Modifying	Specify with the Text Edit dialog box. The Text Edit dialog box is opened by clicking the [...] button that appears at right edge of this field when you select this property (you cannot specify options for the flash memory on this panel).
	Available values	Up to 128 processes (one line in the Text Edit dialog box is equivalent to one processing) Note that up to 64 characters for one process can be specified.
Before running	Specify the process[Note] to proceed right before the execution of the program.	
	Default	Before download[0] ("[]" is the current number of specified processes.)
	Modifying	Specify with the Text Edit dialog box. The Text Edit dialog box is opened by clicking the [...] button that appears at right edge of this field when you select this property (you cannot specify options for the flash memory on this panel).
	Available values	Up to 128 processes (one line in the Text Edit dialog box is equivalent to one processing) Note that up to 64 characters for one process can be specified.
After breaking	Specify the process[Note] to proceed right after the program break.	
	Default	Before download[0] ("[]" is the current number of specified processes.)
	Modifying	Specify with the Text Edit dialog box. The Text Edit dialog box is opened by clicking the [...] button that appears at right edge of this field when you select this property (you cannot specify options for the flash memory on this panel).
	Available values	Up to 128 processes (one line in the Text Edit dialog box is equivalent to one processing) Note that up to 64 characters for one process can be specified.

Note From the following three processes, input the specification format of the desired process to the Text Edit dialog box.

[Process 1]:

Automatically overwrites the value of *I/O register* with *Value*.

Specification format:

```
I/O-register-name Value
```

[Process 2]:

Automatically overwrites the value of *CPU register* with *Value*.

Specification format:

```
CPU-register-name Value
```

[Process 3]:

Automatically executes a script file which is specified with *Python script path* (absolute path or relative path from the project folder).

Specification format:

```
Source Python-script-path
```

Editor panel

This panel is used to display and edit files.

Furthermore, the source level debugging/instruction level debugging (see "2.8.3 Execute programs in steps") and the code coverage measurement result display **[Simulator]** (see "2.14 Measure Coverage [Simulator]") can be performed when connected to the debug tool and the downloaded source file is opened in this panel.

The code data, label and disassembled text can be displayed combined with the source text by selecting the Mixed display mode (see "(1) Change display mode").

When opened the file encoding and newline code is automatically detected and retained when it is saved. You can open a file with a specific encoding selected in the Encoding dialog box. If the encoding and newline code is specified in the Save Settings dialog box then the file is saved with those settings.

This panel can be opened multiple times (up to 100 panels).

Cautions 1. When a project is closed, all of the Editor panels displaying a file being registered in the project are closed.

2. When a file is excluded from a project, the Editor panel displaying the file is closed.

Remark This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

Figure A-15. Editor Panel

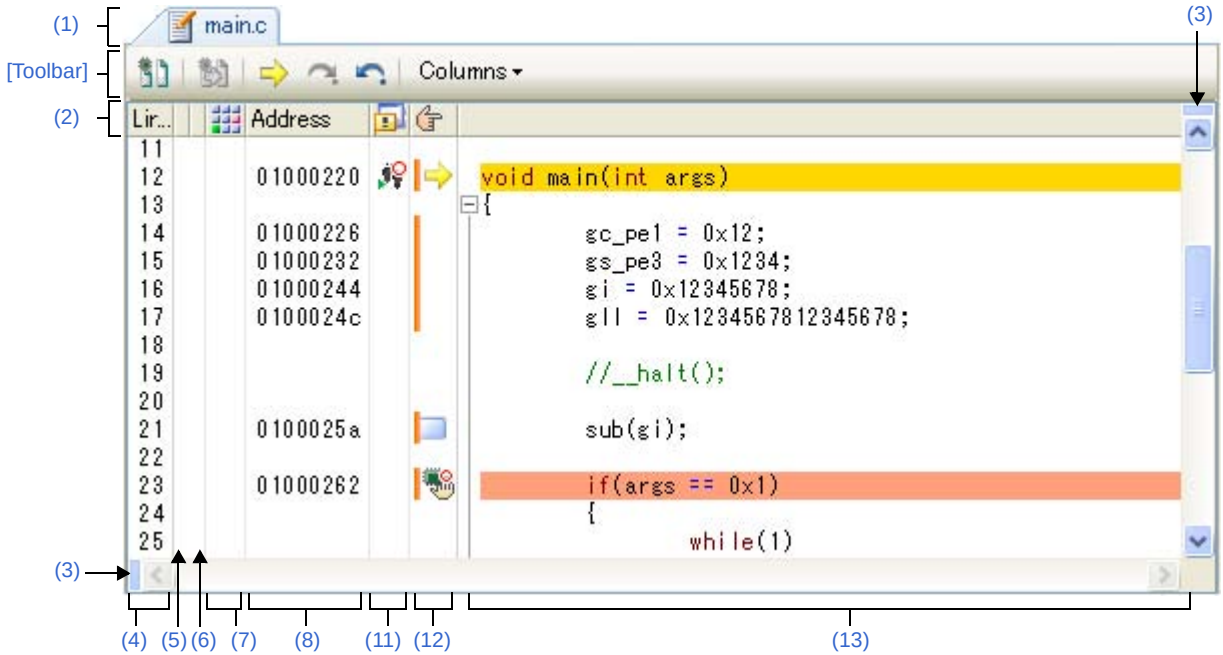


Figure A-16. Editor Panel (When Code Coverage Measurement Result Is Displayed) [Simulator]

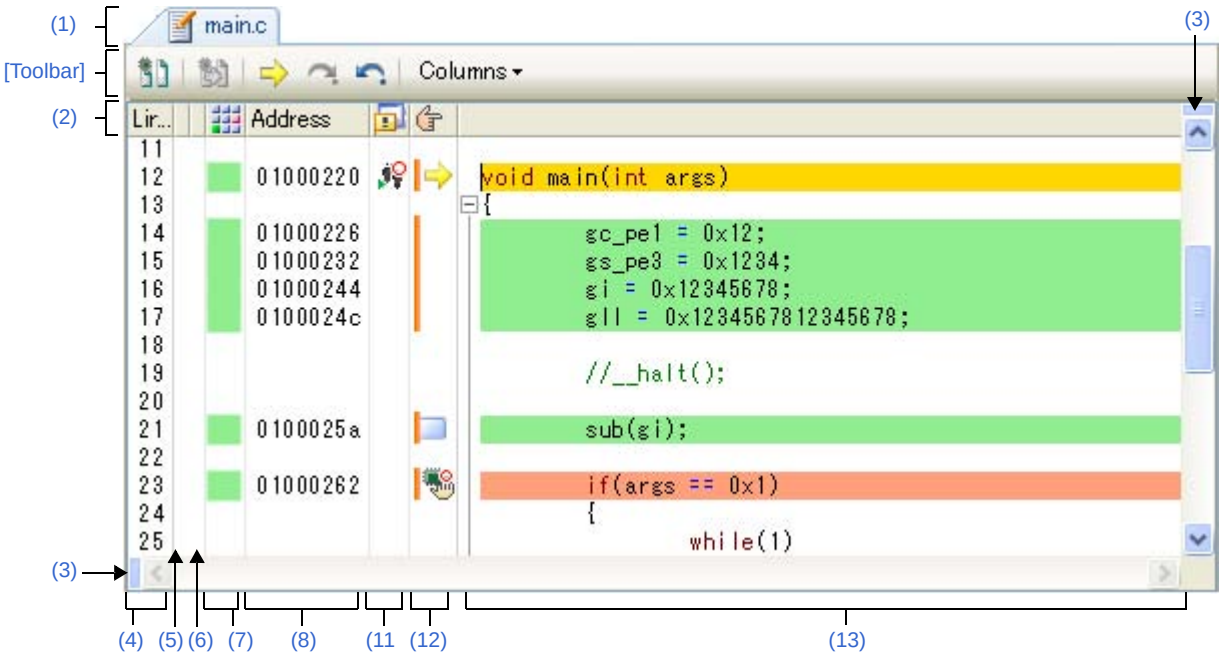
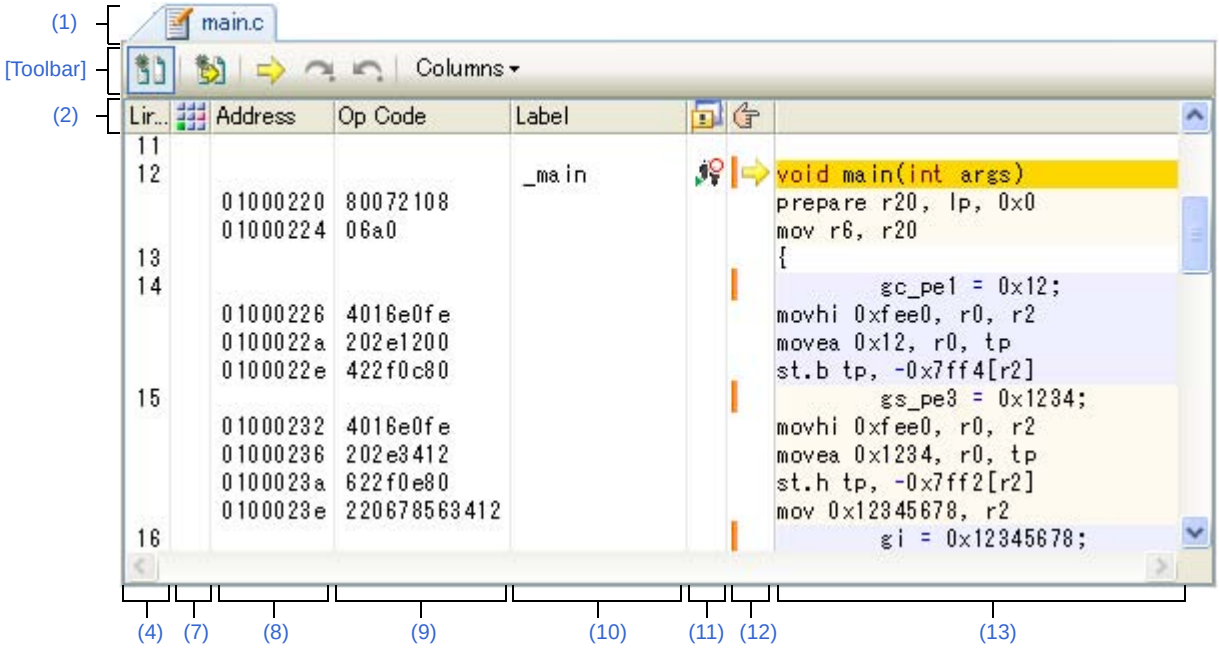


Figure A-17. Editor Panel (When Mixed Display Mode Is Selected)



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Editor panel-dedicated items)]
- [[Edit] menu (Editor panel-dedicated items)]
- [[Window] menu (Editor panel-dedicated items)]
- [Context menu]

[How to open]

- Automatically opens after downloading the load module file with debug information.
- On the Project Tree panel, double click a file.
- On the Project Tree panel, select a file, and then select [Open] from the context menu.
- On the Project Tree panel, select a file, and then select [Open with Internal Editor...] from the context menu.
- On the Project Tree panel, select [Add] >> [Add New File...] from the context menu, and then create a text file or source file.
- On the Disassemble panel, Call Stack panel, Trace panel, or Events panel, select [Jump to Source] from the context menu.
- Automatically opens if there is a source text line corresponding to the current PC value when the current PC value is forcibly changed or the program stops executing.

[Description of each area]

(1) Title bar

The name of the opened file is displayed.

Marks displayed at the end of the file name indicate the following:

Mark	Description
*	The file has been modified since being opened.
!	Update time and date of the source file opened are later than the one of the downloaded load module file. Note that this mark is valid only when connected to the debug tool and the downloaded source file is opened.
[RECYCLE]	The recycle mode (see "(3) Display multiple source files in a single panel") is valid. Note that this mark is valid only when connected to the debug tool and the downloaded source file is opened.
(Read only)	The opened file is read only.

(2) Column header

The title of each column on the Editor panel is displayed.

Hovering the mouse cursor over this area displays the title name.

Display	Title Name	Description
Line	Line	Displays line numbers (see "(4) Line number area").
(No display)	Selection	The display is colored to reflect the state in terms of saving of the state of editing (see "(5) Selection area"). However, this column is not displayed in the Mixed display mode.
(No display)	Out of Date Module Indicator	The display is colored to reflect cases where a source file has been updated more recently than the corresponding load module file (see "(6) Out of date module Indicator area"). However, this column is not displayed when disconnected from the debug tool or in the Mixed display mode.
	Coverage	Displays the coverage information (see "(7) Coverage area"). However, this column is not displayed when disconnected from the debug tool.
Address	Address	Displays addresses (see "(8) Address area"). However, this column is not displayed when disconnected from the debug tool.

Display	Title Name	Description
Op code	Op code	Displays instruction codes (see "(9) Op code area"). However, this column is displayed only in the Mixed display mode.
Label	Label	Displays labels (see "(10) Label area"). However, this column is displayed only in the Mixed display mode.
	Event	Sets events (see "(11) Event area"). However, this column is not displayed when disconnected from the debug tool.
	Main	Displays bookmarks, address marks and the current PC mark. Furthermore, sets breakpoints (see "(12) Main area").

Remark Show/hide of the column header can be switched by the setting of the toolbar.

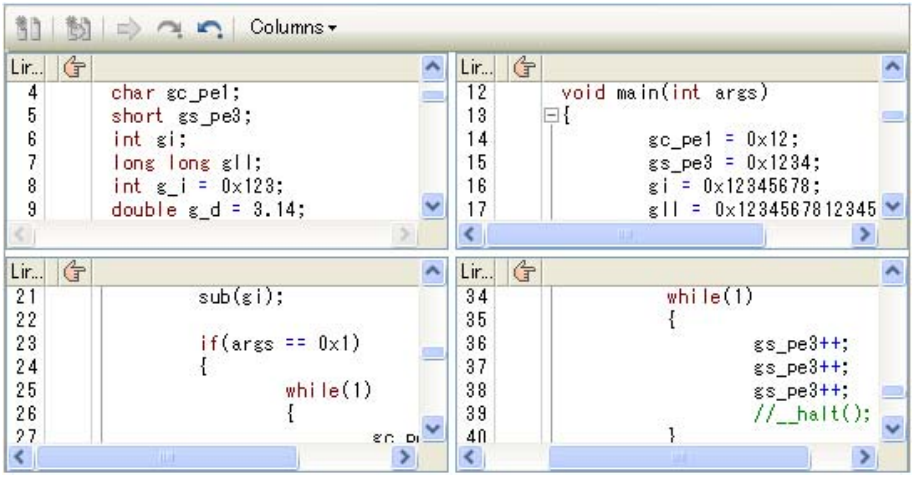
(3) Splitter bars

You can split the Editor panel by using the horizontal and vertical splitter bars within the view. This panel can be split up to two times vertically, and two times horizontally.

- To split this panel, drag the splitter bar down or to the right to the desired position, or double-click any part of the splitter bar.
- To remove the split, double-click any part of the splitter bar.

Caution The split is enabled only when this panel is in the normal display mode (setting to the Mixed display mode removes the split).

Figure A-18. Editor Panel (Vertical/Horizontal Two-way Split View)



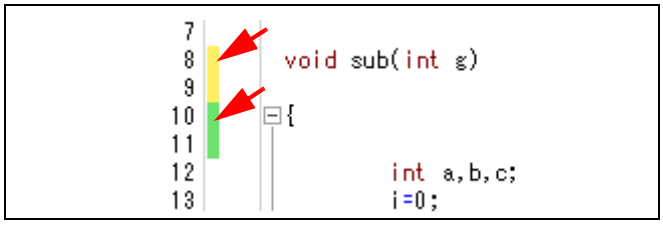
(4) Line number area

This area displays the line number of the opened file.

(5) Selection area

This area displays the following indicators that shows the line modification status (except in the Mixed display mode).

	This means new or modified line but unsaved.
	This means new or modified line and saved. To erase this mark, close the panel, and then open this source file again.

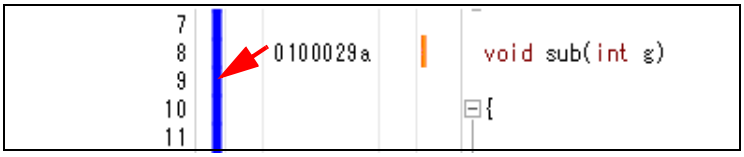


(6) Out of date module Indicator area

This area is valid only when connected to the debug tool and the downloaded source file is opened (except in the Mixed display mode).

If the update time and date of the source file opened are later than the one of the downloaded load module file, the following indicator is displayed (the color of the indicator depends on the "Warning" color of the [General - Font and Color] category of the Option dialog box).

To erase this mark, run a build and then download the load module file again.



(7) Coverage area

This area is valid only when connected to the debug tool and the downloaded source file is opened.

When the coverage function is valid, lines corresponding to the specified coverage measurement area are shown highlighted based on the code coverage measurement result that is acquired by executing the program (the color depends on the coverage color in the [General - Font and Color] category of the Option dialog box).

See "2.14 Measure Coverage [Simulator]" for details on the coverage measurement.

(8) Address area

This area is valid only when connected to the debug tool and the downloaded source file is opened.

This area shows the address corresponding to where the instruction is located in the memory space of the selected microcontroller.

The format of this area is fixed as hexadecimal number notation.

The address width corresponds to the one in memory space of the selected microcontroller in the project.

(9) Op code area

This area is valid only when connected to the debug tool and the downloaded source file is opened in the Mixed display mode.

This area shows the instruction code corresponding to the source text.

(10) Label area

This area is valid only when connected to the debug tool and the downloaded source file is opened in the Mixed display mode.

This area shows the label name when a label is defined for the address.

(11) Event area

This area is valid only when connected to the debug tool and the downloaded source file is opened.

This area is provided with the following functions.

(a) Setting/deleting of various events

By selecting a item from the context menu on the line that has the address mark ( ), a Timer event, Trace event or action event (Printf event) can be set/deleted.
Once an event is set, the Event mark corresponding to the event is displayed at the line that is set. In addition, the detailed information about the set event is reflected in the Events panel.

(b) Pop-up display

By hovering the mouse cursor over the Event mark, the name of the event, the detailed information for the event and the comments added to the event are a pop-up displayed.
When multiple events have been set in the applicable place, information for each event, up to a maximum of three events, is listed and displayed.

(12) Main area

This area is provided with the following functions.

(a) Bookmarks display

Bookmarks ( ) that have been registered are displayed.
See "(9) Register a bookmark" for details on the bookmark.

Caution This function is disabled when the Mixed display mode is selected.

The following functions are also available when the debug tool is connected and a downloaded source file is open.

(b) Address marks display

Address marks ( ) are displayed at lines that have valid addresses.
Breakpoints or various events can be set at lines with the address mark.

(c) Current PC mark display

The current PC mark ( ) that corresponds to the current PC position (PC register value) is displayed.
Note that the current PC mark is only displayed if the current PC value corresponds to the source text line, when the state of the debug tool is changed from execution to stop.

Remark When the Mixed display mode is selected, if the unit of step execution is set to instruction level by selecting the button on the toolbar, then the current PC mark will be moved to a disassembled text line.

(d) Setting/deleting of breakpoints

By clicking the line that has the address mark ( ) with the mouse, the breakpoints can be set easily.
Once a breakpoint is set, an Event mark is displayed at the line that is set. In addition, the detailed information about the set breakpoint is reflected in the Events panel.
When this operation is performed at a place where a breakpoint is already set, that breakpoint is deleted and the setting of breakpoints cannot be done.
See "2.9.3 Stop the program at the arbitrary position (breakpoint)" for details on how to set the breakpoint.

Remark Setting a breakpoint and changing the state of a breakpoint can also be done from the context menu in this area.

(e) Pop-up display

By hovering the mouse cursor over the Event mark, the name of the event, the detailed information for the event and the comments added to the event are a pop-up displayed.

When multiple events have been set in the applicable place, information for each event, up to a maximum of three events, is listed and displayed.

(13) Characters area

This area displays character strings of files and you can edit it.
This area is provided with the following functions.

(a) Characters editing

Characters can be entered from the keyboard.
Various shortcut keys can be used to enhance the edit function.

Caution This function is disabled when the Mixed display mode is selected.

Remark The following items can be customized by setting the Option dialog box.

- Display fonts
- Tab interval
- Show or hide white space marks (blank symbols)
- Colors of reserved words and comments

(b) Code outlining

This allows you to expand and collapse source code blocks so that you can concentrate on the areas of code which you are currently modifying or debugging. This is only available for only C source file types.
This is achieved by clicking the plus and minus symbols to the left of the Characters area.
Types of source code blocks that can be expanded or collapsed are:

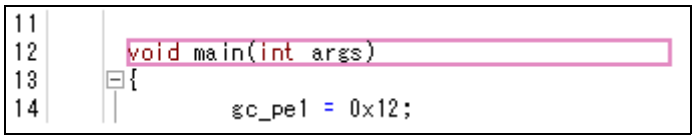
Open and close braces ('{' and '}')	⊞{ ... }
Multi-line comments ('/*' and '*/')	⊞/** */
Pre-processor statements ('if', 'elif', 'else', 'endif')	⊞#if [Preprocessor block] ⊞#elif [Preprocessor block] ⊞#else [Preprocessor block] #endif

Caution This function is disabled when the Mixed display mode is selected.

(c) Highlighting the current line

By selecting the [Enable line highlight for current] check box in the []General - Text Editor] category of the Option dialog box, the line at the current caret position can be displayed within a rectangle (the rectangle color depends on the highlight color in the [General - Font and Color] category of the same dialog box above).

Figure A-19. Highlighting Current Line



(d) Emphasizing brackets

The bracket that corresponds to a bracket at the caret position is shown emphasized.
Supported types of brackets vary with the file type.

File Type	Types of Brackets
C or Python	(and), { and }, [and]
HTML or XML	< and >

Remark When CubeSuite+ emphasizes the corresponding bracket, it does not consider those within comments, character constants, character strings, or string constants. For this reason, if the bracket at the position of the caret is within a comment, character constant, character string, or string constant, CubeSuite+ may emphasize a bracket that is not actually the corresponding bracket.

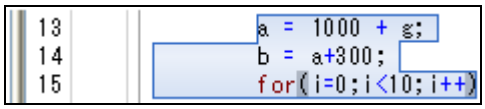
(e) Multiple lines selection and block selection

You can select multiple lines or a block that consists of multiple lines by any one of the following methods:

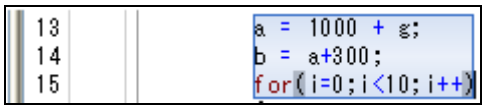
- Multiple lines selection
  - Drag the left-mouse button
  - Press the [Right], [Left], [Up] or [Down] key while holding down the [Shift] key
- Block selection
  - Drag the left-mouse button while holding down the [Alt] key
  - Press the [Right], [Left], [Up] or [Down] key while holding down the [Alt] + [Shift] key

Figure A-20. Multiple Lines Selection and Block Selection

[Multiple lines selection]



[Block selection]



Caution The information on bookmarks is not included in the selected contents.

Remark Editing of the selected contents can be done by using [Cut], [Copy], [Paste], or [Delete] from the [Edit] menu.

(f) Jump to functions

It automatically recognizes the currently selected characters or the word at the caret position as the function name and jumps to the target function.

See "Table 2-3. Operation of Tag Jump" for details on the jump to functions.

(g) Tag jump

If the information of a file name, a line number and a column number exists in the line at the caret position, selecting [Tag Jump] from the context menu opens the file in a new Editor panel and jumps to the corresponding line and the corresponding column (if the target file is already opened in the Editor panel, you can jump to the panel).

See "(8) Jump to a desired line (tag jump)" for details on the tag jump.

(h) Registration of bookmarks

By clicking the button on the bookmark toolbar or selecting [Bookmark] >> [Toggle Bookmark] from the context menu on this area, a bookmark can be registered to the line at the caret position.

See "(9) Register a bookmark" for details on the bookmark.

Caution This function is disabled when the Mixed display mode is selected.

(i) File monitor

If the contents of the currently displayed file is changed (including renaming or deleting) without using CubeSuite+, a message will appear asking you whether you wish to update the file or not.

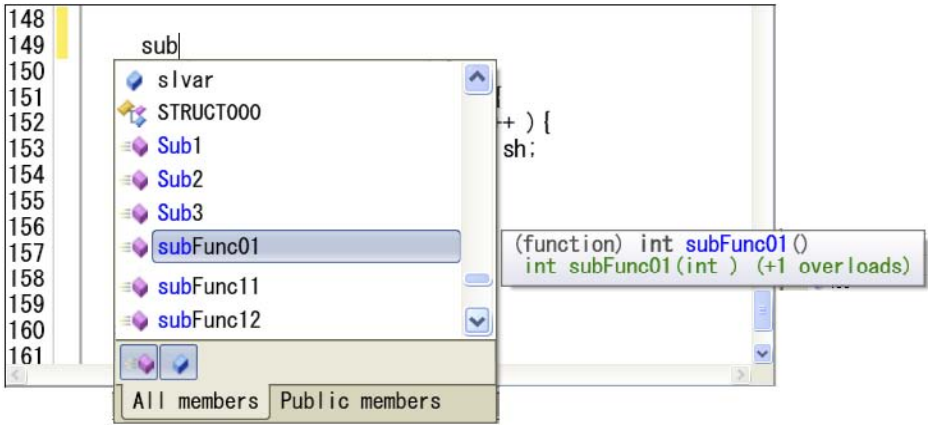
(j) Smart edit function

The smart edit function is used to complement the names of functions, variables and the arguments of functions during input and offer them as candidates.

The smart edit function operates with the items listed below.

- Global functions in the C language
- Global variables in the C language

Figure A-21. Display Example of Smart Edit Function (Candidates of Function and Variables)



Cautions 1. This function is supported only when CC-RH is used as the build tool (i.e. this function is disabled when GHS compiler is used).

2. This function is disabled when the Mixed display mode is selected.

Follow the procedure below to enable the smart edit function.

- Select the [Smart edit] check box in the [General - Text Editor] category of the Option dialog box (default).
- Candidates are displayed by using the cross reference information that is generated by the build tool. Therefore, set the build tool's Property panel[Note] so that the cross reference information is output, and then run and complete a build.
  If an error in building occurs, the cross reference information before the error occurred is used if any exists.

Note [Common Options] tab] >> [Output File Type and Path] category >> [Output cross reference information] property >> [Yes(-Xcref)]
If this setting is invalid, the smart edit function cannot be used since the output will be empty of the cross reference information.

<1> Display of candidates for functions and variables

- **How to display**

Candidates for functions and variables are displayed when:

- In the C language, "." or "->" is input if there is a relevant member for the left side.

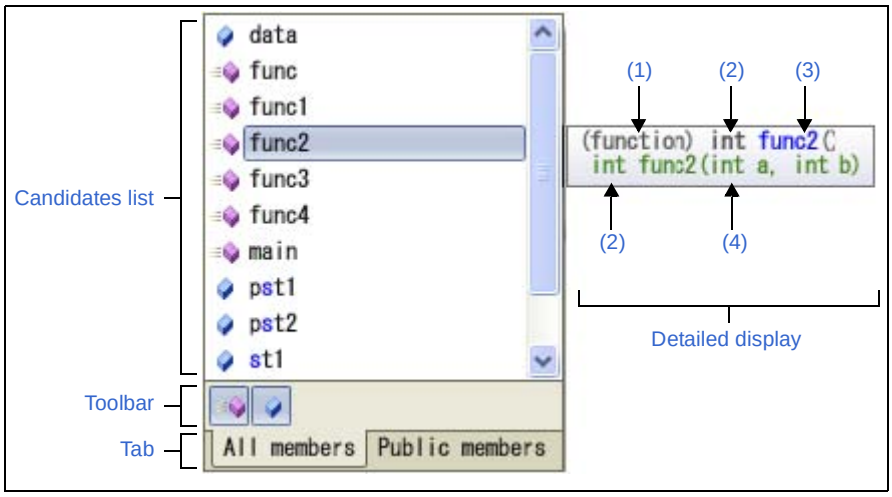
- The [Ctrl] + [Space] key on the keyboard is pressed (all candidates are displayed).
  However, if there is only one candidate, the relevant character string is inserted at this time without displaying the candidate.

- **How to insert character strings**

Select a character string from the candidates list by using the [Up]/[Down] key or the mouse, then press the [Enter] key or the [TAB] key.

- **Description of each area**

Figure A-22. Display of Candidates for Functions and Variables



- Candidates list

Displays candidates for functions and variables in alphabetical order.
If there are character strings that match to the character strings at the caret position, they are highlighted (case insensitive).
The following icons are displayed as labels for the list of candidates.

Icon	Description
	Shows that the candidate is for a typedef.
	Shows that the candidate is for a function.
	Shows that the candidate is for a variable.
	Shows that the candidate is for a structure type.
	Shows that the candidate is for an union type.

- Toolbar

Switches whether candidates for functions and variables are displayed or not.

Button	Description
	Displays candidates for functions.
	Displays candidates for variables.

- Tab

Switches the members to be displayed.

Tab Name	Description
All members	Displays all candidates.
Public members	Displays only the candidates with the public attribute.

- Detailed display

Displays details of candidates for functions or variables currently being selected.

Item	Description
(1) Kind	Shows whether the selected item is a function or a variable. (function): Shows the selected item is a function. (variable): Shows the selected item is a variable.
(2) Type	Shows the type of the function or the variable.
(3) Name	Shows the name of the function or the variable.
(4) Name and argument	Shows the name of the function or the variable. When the item is a function, its arguments are also shown.

<2> Display of candidates for arguments

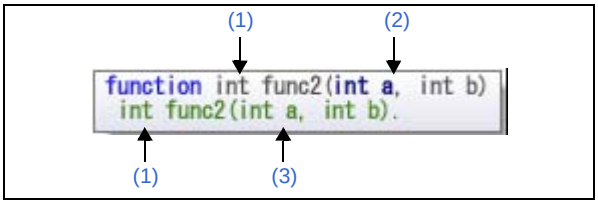
- **How to display**

Candidates for arguments are displayed when:

- In a function name, "(" is input if there is a relevant function on the left side of "(".
- The [Ctrl] + [Shift] + [Space] key on the keyboard is pressed while the text cursor is at the location of an argument for a function.

- Description of each area

Figure A-23. Display of Candidates for Arguments



Item	Description
(1) Type	Shows the type of the function or the variable.
(2) Name and argument	Shows the name of the function and its arguments. The argument at the current caret position is highlighted.
(3) Name and argument	Shows the name of the function and its arguments.

<3> Termination of the candidates display

The candidates display disappears by any one of the following methods:

- Press the [ESC] key

- Enter a key other than an alphanumeric character
  When nothing is selected from the candidates list: This operation has no effect.
  When an item is selected in the candidates list: The selected character strings are inserted.

<4> Notes for displaying of candidates list

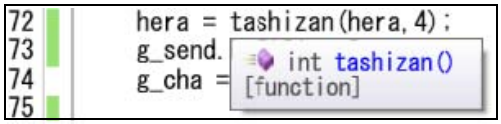
- The following items are not the subject of the candidates display.
  - Macro definitions
  - Local variables
  - Typedef statements
- When a structure or union is declared within a function, candidates are not displayed within the function after its own declaration.
- In some cases the type of variables to be displayed differs from that actually declared when a compiler option which affects the size of variables is set.

Remark When the mouse cursor is hovered over a function name or a variable name on the source text, the information about that function or variable appears in a pop-up.
Note the following, however, when using this function.
-This function cannot be used when connected to the debug tool.
-const, static, and volatile attributes cannot be displayed in a pop-up.
-If the target is a variable of class, structure, union, or enumeration type, its members are displayed as follows:
  - If the target is a class-, structure-, or union-type variable, the types and names of its members are displayed.
  - If the target is a class-type variable that includes methods (functions) among its members, the types of the return values and names of the methods (functions) are displayed. Also, '(' ')' is appended to the end of each method name.
  - If the target is an enumeration-type variable, only the names of the members are displayed.
  - Members are displayed in the same order as they are defined in the source file, and each is placed on a single line (up to 20 members can be displayed).

The meaning of each icon displayed in a pop-up is described below.

Icon	Description
	Shows that the target is for a typedef.
	Shows that the target is for a function.
	Shows that the target is for a variable.
	Shows that the target is for a structure type.
	Shows that the target is for an union type.
	Shows that the target is for an enumeration type.

Figure A-24. Pop-up Display of Smart Edit Function



The following functions are also available when the debug tool is connected and a downloaded source file is open.

(k) Highlighting the current PC line

When the current PC position (PC register value) corresponds to the source text lines, those lines are shown highlighted (the highlighting color depends on the current PC color in the [General - Font and Color] category of the Option dialog box).

(l) Highlighting lines with breakpoints

Lines where the breakpoints are set are shown highlighted (the highlighting color depends on the breakpoint color in the [General - Font and Color] category of the Option dialog box).

(m) Code coverage measurement result display [Simulator]

When the coverage function is valid, lines corresponding to the specified coverage measurement area are shown highlighted based on the code coverage measurement result that is acquired by executing the program (the highlighting color depends on the coverage color in the [General - Font and Color] category of the Option dialog box).

See "2.14 Measure Coverage [Simulator]" for details on the coverage measurement.

(n) Pop-up display of variables

When hovering the mouse cursor over a variable in the source text, a pop-up that shows the name and value of the variable is displayed (see "(4) Display variables").

(o) Setting of various events

By selecting a item from the context menu on the line that has the address mark ( ), various events can be set. Once an event is set, the Event mark corresponding to the event is displayed at the line that is set in the Event area or the Main area. In addition, the detailed information about the set event is reflected in the Events panel.

See the following for details on how to set events.

- "2.9.4 Stop the program at the arbitrary position (break event)"
- "2.9.5 Stop the program with the access to variables/I/O registers"
- "2.12.3 Collect execution history in the arbitrary section"
- "2.12.4 Collect execution history only when the condition is met [Simulator]"
- "2.13.2 Measure execution time in the arbitrary section [Simulator]"
- "2.15.1 Inset printf"

(p) Registration of watch expressions

C language variable, CPU registers, I/O registers, and assembler symbols can be registered in the Watch panel as watch expressions.

See "(1) Register a watch-expression" for details on how to operate it.

[Toolbar]

	Toggles between the normal display mode (default) and the mixed display mode, as the display mode of this panel (see "(1) Change display mode"). Note that this item is enabled only when connected to the debug tool and the downloaded source file is opened in this panel.
	Toggles between source (default) and instruction level, as the unit in which the program is step-executed (see "2.8.3 Execute programs in steps"). When the unit of a step execution is set to instruction level, then the current PC mark will be moved to a disassembled text line Note that this item is enabled only when connected to the debug tool and the downloaded source file is opened in this panel.

	Displays the current PC position. Note that this item is enabled only when connected to the debug tool.
	Forwards to the position before operating [Context menu] >> [Back To Last Cursor Position]. Note that this item is disabled when this panel is in the mixed display mode.
	Goes back to the position before operating [Context menu] >> [Jump to Function]. Note that this item is disabled when this panel is in the mixed display mode.
Columns	The following items are displayed to show or hide the columns or marks on all of the Editor panels. Remove the check to hide the items (all the items are checked by default). This setting is reflected in all the Editor panels.
Line Number	Shows the line number, in the line number area.
Selection	Shows the mark that indicates the line modification status, in the line number area.
Out of date module indicator	Shows the mark that indicates the update status of the downloaded load module file, in the line number area. Note that this item is enabled only when connected to the debug tool.
Coverage	Shows the coverage area. Note that this item is enabled only when connected to the debug tool.
Address	Shows the address area. Note that this item is enabled only when connected to the debug tool.
Op Code	Shows the code area. Note that this item is enabled only when connected to the debug tool and the mixed display mode is selected.
Label	Shows the label area. Note that this item is enabled only when connected to the debug tool and the mixed display mode is selected.
Event	Shows the event area. Note that this item is enabled only when connected to the debug tool.
Main	Shows the main area.
Column Header	Shows the column header.

[[File] menu (Editor panel-dedicated items)]

The following items are exclusive for the [File] menu in the Editor panel (other items are common to all the panels).

Close *file name*	Closes the currently editing Editor panel. When the contents of the panel have not been saved, a confirmation message is shown.
Save *file name*	Overwrites the contents of the currently editing Editor panel. When the file has never been saved or the file is read only, the same operation is applied as the selection in [Save *file name* As...]. Note that this item is disabled when this panel is in the mixed display mode.
Save *file name* As...	Opens the Save As dialog box to newly save the contents of the currently editing Editor panel. Note that if this panel is in the mixed display mode, then "*file name*" will be changed to "Source Mixed Data".
File name Save Settings...	Opens the Save Settings dialog box to change the encoding and newline code of the file being opened in the currently editing Editor panel.

Page Setup...	This item is always disabled.
Print...	Opens the Windows dialog box for printing the contents of the currently editing Editor panel.
Print Preview	Opens the Print Preview window to preview the file contents to be printed.

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

The following items are exclusive for [Edit] menu in the Editor panel (all other items are disabled).

Undo	Cancels the previous operation and restores the characters and the caret position (up to 100 times). Note that this item is disabled when this panel is in the mixed display mode.
Redo	Cancels the previous [Undo] operation and restores the characters and the caret position. Note that this item is disabled when this panel is in the mixed display mode.
Cut	Cuts the selected character string and copies it to the clipboard. If there is no selection, the entire line is cut. Note that this item is disabled when this panel is in the mixed display mode.
Copy	Copies the contents of the selected range to the clipboard as character string(s). If there is no selection, the entire line is copied.
Paste	Inserts (insert mode) or overwrites (overwrite mode) the characters that are copied on the clip board into the caret position. Note that this item is disabled when the contents of the clipboard are not recognized as characters or this panel is in the mixed display mode.
Delete	Deletes one character at the caret position. When there is a selection area, all the characters in the area are deleted. Note that this item is disabled when this panel is in the mixed display mode.
Select All	Selects all the characters from beginning to the end in the currently editing text file. Note that this item is disabled when this panel is in the mixed display mode.
Find...	Opens the Find and Replace dialog box with selecting [Quick Find] tab.
Replace...	Opens the Find and Replace dialog box with selecting [Quick Replace] tab. Note that this item is disabled when this panel is in the mixed display mode.
Go To...	Opens the Go to Line dialog box to move the caret to the specified line.
Bookmark	Displays a cascading menu for bookmarks (see "(9) Register a bookmark"). Note that these items are disabled when this panel is in the mixed display mode.
Toggle Bookmark	Registers a bookmark to the line at the current caret position. If a bookmark is already being registered to the line, then the bookmark will be deleted.
Next Bookmark	Moves the caret to the bookmark position that registered next, in the active Editor panel.
Previous Bookmark	Moves the caret to the bookmark position that registered previously, in the active Editor panel.
Clear All Bookmarks	Deletes all bookmarks currently being registered, in the active Editor panel.
List Bookmarks...	Opens the Bookmarks dialog box to list bookmarks currently being registered.
Outlining	Displays a cascading menu for controlling expand and collapse states of source file outlining (see "(b) Code outlining"). Note that these items are disabled when this panel is in the mixed display mode.

Collapse to Definitions	Collapses all nodes that are marked as implementation blocks (e.g. function definitions).
Toggle Outlining Expansion	Toggles the current state of the innermost outlining section in which the cursor lies when you are in a nested collapsed section.
Toggle All Outlining	Toggles the collapsed state of all outlining nodes, setting them all to the same expanded or collapsed state. If there is a mixture of collapsed and expanded nodes, all nodes will be expanded.
Stop Outlining	Stops code outlining and remove all outlining information from source files.
Start Automatic Outlining	Starts automatic code outlining and automatically displayed in supported source files.
Advanced	Displays a cascading menu for performing an advanced operation for the Editor panel. Note that these items are disabled when this panel is in the mixed display mode.
Increase Line Indent	Increases the indentation of the current cursor line by one tab.
Decrease Line Indent	Decreases the indentation of the current cursor line by one tab.
Uncomment Lines	Removes the first set of line-comment delimiters from the start of the current cursor line, appropriate to the current language. This operation will only be available when the language of the current source file has line-comment delimiters specified.
Comment Lines	Places line-comment delimiters at the start of the current cursor line, appropriate to the current language. This operation will only be available when the language of the current source file has line-comment delimiters specified.
Convert Tabs to Spaces	Converts all tabs on the current cursor line into spaces.
Convert Spaces to Tabs	Converts each set of consecutive space characters on the current line to tab characters, but only for those sets of spaces that are at least equal to one tab size.
Tabify Selected Lines	Tabifies the current line, causing all spaces at the start of the line (prior to any text) to be converted to tabs where possible.
Untabify Selected Lines	Untabifies the current line, causing all tabs at the start of the line (prior to any text) to be converted to spaces.
Make Uppercase	Converts all letters within the selection to uppercase.
Make Lowercase	Converts all letters within the selection to lowercase.
Toggle Character Casing	Toggles the character cases (uppercase / lowercase) of all letters within the selection.
Capitalize	Capitalizes the first character of every word within the selection.
Delete Horizontal Whitespace	Deletes any excess white space either side of the cursor position, leaving only one whitespace character remaining. If there the cursor is within a word or not surrounded by whitespace, this operation will have no effect.
Trim Trailing Whitespace	Deletes any trailing whitespace that appears after the last non-whitespace character on the cursor line.
Delete Line	Completely delete the current cursor line.
Duplicate Line	Duplicates the cursor line, inserting a copy of the line immediately after the cursor line.
Delete Blank Lines	Deletes the line at the cursor if it is empty or contains only whitespace.

[[Window] menu (Editor panel-dedicated items)]

The following items are exclusive for the [Window] menu in the Editor panel (other items are common to all the panels).

Split	Splits the active Editor panel horizontally. Only the active Editor panel can be split. Other panels will not be split. A panel can be split up to four times.

Remove Split	Removes the split view of the Editor panel.

[Context menu]

(1) Titlebar area

Close Panel	Closes the currently selected panel.
Close All but This	Closes all other panels being displayed in the same panel display area as the selected panel, except for the currently selected panel.
Save *file name*	Saves the contents of the opened text file.
Copy FUll Path	Copies the full path of the opened text file to the clipboard.
Open Containing Folder	Opens the folder where the text file is saved in Explorer.
New Horizontal Tab Group	The area for the display of active panels is evenly divided into two areas in the horizontal direction, and the panels are displayed as a new group of tabbed pages. Only one panel is active in the new group. The area may be divided into up to four panels. This item is not displayed in the following cases. - Only one panel is open. - The group has already been divided in the vertical direction. - The group has already been divided into four panels.
New Vertical Tab Group	The area for the display of active panels is evenly divided into two areas in the vertical direction, and the panels are displayed as a new group of tabbed pages. Only one panel is active in the new group. The area may be divided into up to four panels. This item is not displayed in the following cases. - Only one panel is open. - The group has already been divided in the horizontal direction. - The group has already been divided into four panels.
Go to Next Tab Group	When the display area is divided in the horizontal direction, this moves the displayed panel to the group under that displaying the selected panel. When the display area is divided in the vertical direction, this moves the displayed panel to the group to the right of that displaying the selected panel. This item is not displayed if there is no group in the given direction.
Go to Previous Tab Group	When the display area is divided in the horizontal direction, this moves the displayed panel to the group over that displaying the selected panel. When the display area is divided in the vertical direction, this moves the displayed panel to the group to the left of that displaying the selected panel. This item is not displayed if there is no group in the given direction.

(2) Coverage area [Simulator]

Clear Coverage Information	Clears all the coverage measurement results currently being stored in the debug tool.

(3) Event area

Set Timer Start Event	Sets a timer start event to start measuring the execution time of the program when the line at caret is executed (see "2.13.2 Measure execution time in the arbitrary section [Simulator]").
Set Timer End Event	Sets a timer end event to stop measuring the execution time of the program when the line at caret is executed (see "2.13.2 Measure execution time in the arbitrary section [Simulator]").
Set Trace Start Event	Sets a trace start event to start collecting the trace data when the line at the caret is executed (see "2.12.3 Collect execution history in the arbitrary section").
Set Trace End Event	Sets a trace end event to stop collecting the trace data when the line at the caret is executed (see "2.12.3 Collect execution history in the arbitrary section").
Register Action Event...	Opens the Action Events dialog box to set an action event to the corresponding address of the line at the caret position (see "2.15.1 Inset printf").
Enable Event(s)	Changes the state of a selected event to a Valid state. If the event mark ( ) which indicates that multiple events have been set is selected, all of the events that have been set are enabled.
Disable Events(s)	Changes the state of a selected event to an Invalid state. If the event mark ( ) which indicates that multiple events have been set is selected, all of the events that have been set are disabled.
Delete Event(s)	Deletes a selected event. If the event mark ( ) which indicates that multiple events have been set is selected, all of the events that have been set are deleted.
View Details in Event Panel	Opens the Events panel to display the detailed information of the selected event.

(4) Main area (when connected to the debug tool)

Set Breakpoint	Sets a breakpoint to the line at the caret position (see "2.9.3 Stop the program at the arbitrary position (breakpoint)")[Note]. If a breakpoint is already being set to the line, then the breakpoint will be deleted.
Set Hardware Breakpoint **[Full-spec emulator] [E1][E20]**	Sets a breakpoint (Hardware Break event) to the line at the caret position.
Set Software Breakpoint **[Full-spec emulator] [E1][E20]**	Sets a breakpoint (Software Break event) to the line at the caret position.
Hardware Break First **[Full-spec emulator] [E1][E20]**	The type of break that can be set by a one click operation of the mouse is set as a hardware breakpoint (this is reflected in the setting of the [First using type of breakpoint] property in the [Break] category from the [Debug Tool Settings] tab on the Property panel).
Software Break First **[Full-spec emulator] [E1][E20]**	The type of break that can be set by a one click operation of the mouse is set as a software breakpoint (this is reflected in the setting of the [First using type of breakpoint] property in the [Break] category from the [Debug Tool Settings] tab on the Property panel).
Enable Breakpoint	Changes the selected breakpoint state to a Valid state. If the event mark ( ) which indicates that multiple events have been set is selected, all of the breakpoints that have been set are enabled.
Disable Breakpoint	Changes the selected breakpoint state to an Invalid state. If the event mark ( ) which indicates that multiple events have been set is selected, all of the breakpoints that have been set are disabled.

Delete Breakpoint	Deletes the selected breakpoint. If the event mark () which indicates that multiple events have been set is selected, all of the breakpoints that have been set are deleted.
View Details in Event Panel	Opens the Events panel to display the detailed information of the selected event.

Note [Full-spec emulator][E1][E20]

By default the debug tool will set a hardware break when resources are available.
This behavior can be customized by using the [Hardware Break First] or [Software Break First] menu items.

(5) Characters area (when disconnected from the debug tool)

Cut	Cuts the selected character string and copies it to the clipboard. If there is no selection, the entire line is cut.
Copy	Copies the contents of the selected range to the clipboard as character string(s). If there is no selection, the entire line is copied.
Paste	Inserts (insert mode) or overwrites (overwrite mode) the characters that are copied on the clip board into the caret position. When the contents of the clipboard are not recognized as characters, the operation is invalid.
Find...	Opens the Find and Replace dialog box with selecting [Quick Find] tab.
Go To...	Opens the Go to Line dialog box to move the caret to the specified line.
Forward To Next Cursor Position	Forwards to the position before operating [Back To Last Cursor Position]. Note that this item is disabled when this panel is in the mixed display mode.
Back To Last Cursor Position	Goes back to the position before operating [Jump to Function]. Note that this item is disabled when this panel is in the mixed display mode.
Jump to Function	Jumps to the function that is selected or at the caret position regarding the selected characters and the words at the caret position as functions (see "(7) Jump to functions").
Tag Jump	Jumps to the corresponding line and column in the corresponding file if the information of a file name, a line number and a column number exists in the line at the caret position (see "(g) Tag jump").
Bookmark	Displays a cascading menu for bookmarks (see "(9) Register a bookmark"). Note that these items are disabled when this panel is in the mixed display mode.
Toggle Bookmark	Registers a bookmark to the line at the current caret position. If a bookmark is already being registered to the line, then the bookmark will be deleted.
Next Bookmark	Moves the caret to the bookmark position that registered next, in the active Editor panel.
Previous Bookmark	Moves the caret to the bookmark position that registered previously, in the active Editor panel.
Clear All Bookmarks	Deletes all bookmarks currently being registered, in the active Editor panel.
List Bookmarks...	Opens the Bookmarks dialog box to list bookmarks currently being registered.
Advanced	Displays a cascading menu for performing an advanced operation for the Editor panel.

Increase Line Indent	Increases the indentation of the current cursor line by one tab.
Decrease Line Indent	Decreases the indentation of the current cursor line by one tab.
Uncomment Lines	Removes the first set of line-comment delimiters from the start of the current cursor line, appropriate to the current language. This operation will only be available when the language of the current source file has line-comment delimiters specified.
Comment Lines	Places line-comment delimiters at the start of the current cursor line, appropriate to the current language. This operation will only be available when the language of the current source file has line-comment delimiters specified.
Convert Tabs to Spaces	Converts all tabs on the current cursor line into spaces.
Convert Spaces to Tabs	Converts each set of consecutive space characters on the current line to tab characters, but only for those sets of spaces that are at least equal to one tab size.
Tabify Selected Lines	Tabifies the current line, causing all spaces at the start of the line (prior to any text) to be converted to tabs where possible.
Untabify Selected Lines	Untabifies the current line, causing all tabs at the start of the line (prior to any text) to be converted to spaces.
Make Uppercase	Converts all letters within the selection to uppercase.
Make Lowercase	Converts all letters within the selection to lowercase.
Toggle Character Casing	Toggles the character cases (uppercase / lowercase) of all letters within the selection.
Capitalize	Capitalizes the first character of every word within the selection.
Delete Horizontal Whitespace	Deletes any excess white space either side of the cursor position, leaving only one whitespace character remaining. If there the cursor is within a word or not surrounded by whitespace, this operation will have no effect.
Trim Trailing Whitespace	Deletes any trailing whitespace that appears after the last non-whitespace character on the cursor line.
Delete Line	Completely delete the current cursor line.
Duplicate Line	Duplicates the cursor line, inserting a copy of the line immediately after the cursor line.
Delete Blank Lines	Deletes the line at the cursor if it is empty or contains only whitespace.

(6) Characters area (when connected to the debug tool)

Register to Watch1	Registers a selected character string or a word at the caret position to the Watch panel (Watch1) as a watch-expression (the judgment of the word depends on current build tool). Note that this item is disabled when no corresponding address exists in the line at caret.
Register to Analysis Chart	Registers a selected character string or a word at the caret position to the Analysis Chart panel of the analyze tool (Program Analyzer) as a variable. If variables have been already registered to all channels, a message is displayed and this operation will have no effect. Note that this item is disabled when the active project does not support a plug-in of the analyze tool.
Register Action Event...	Opens the Action Events dialog box to set an action event to the corresponding address of the line at the caret position[Note 1]. Note that this item is disabled when no corresponding address exists in the line at caret.
Cut	Deletes the selected character string(s) and copies them to the clipboard. If there is no selection, the entire line is cut. Note that this item is disabled when this panel is in the mixed display mode.

Copy	Copies the contents of the selected range to the clipboard as character string(s). If there is no selection, the entire line is copied.
Paste	Inserts (insert mode) or overwrites (overwrite mode) the characters that are copied on the clip board into the caret position. Note that this item is disabled when the contents of the clipboard are not recognized as characters or this panel is in the mixed display mode.
Find...	Opens the Find and Replace dialog box with selecting [Quick Find] tab.
Go To...	Opens the Go to Line dialog box to move the caret to the specified line.
Forward To Next Cursor Position	Forwards to the position before operating [Back To Last Cursor Position]. Note that this item is disabled when this panel is in the mixed display mode.
Back To Last Cursor Position	Goes back to the position before operating [Jump to Function]. Note that this item is disabled when this panel is in the mixed display mode.
Go to Here	Executes the program from the address indicated by the current PC value to the address corresponding to the line at the caret position[Note 1]. Note that this item is disabledd during execution of a program or [Build & Download].
Set PC to Here	Sets the address of the line at the current caret position to the current PC value[Note 1]. Note that this item is disabled when no corresponding address exists in the line at caret, or during execution of a program or [Build & Download].
Jump to Function	Jumps to the function that is selected or at the caret position regarding the selected characters and the words at the caret position as functions (see "(7) Jump to functions").
Tag Jump	Jumps to the corresponding line and column in the corresponding file if the information of a file name, a line number and a column number exists in the line at the caret position (see "(g) Tag jump").
Jump to Disassemble	Opens the Disassemble panel and jumps to the address corresponding to the line at the caret[Note 1]. Note that this item is disabled when no corresponding address exists in the line at caret.
Bookmark	Displays a cascading menu for bookmarks (see "(9) Register a bookmark"). Note that these items are disabled when this panel is in the mixed display mode.
Toggle Bookmark	Registers a bookmark to the line at the current caret position. If a bookmark is already being registered to the line, then the bookmark will be deleted.
Next Bookmark	Moves the caret to the bookmark position that registered next, in the active Editor panel.
Previous Bookmark	Moves the caret to the bookmark position that registered previously, in the active Editor panel.
Clear All Bookmarks	Deletes all bookmarks currently being registered, in the active Editor panel.
List Bookmarks...	Opens the Bookmarks dialog box to list bookmarks currently being registered.
Advanced	Displays a cascading menu for performing an advanced operation for the Editor panel. Note that these items are disabled when this panel is in the mixed display mode.

Increase Line Indent	Increases the indentation of the current cursor line by one tab.
Decrease Line Indent	Decreases the indentation of the current cursor line by one tab.
Uncomment Lines	Removes the first set of line-comment delimiters from the start of the current cursor line, appropriate to the current language. This operation will only be available when the language of the current source file has line-comment delimiters specified.
Comment Lines	Places line-comment delimiters at the start of the current cursor line, appropriate to the current language. This operation will only be available when the language of the current source file has line-comment delimiters specified.
Convert Tabs to Spaces	Converts all tabs on the current cursor line into spaces.
Convert Spaces to Tabs	Converts each set of consecutive space characters on the current line to tab characters, but only for those sets of spaces that are at least equal to one tab size.
Tabify Selected Lines	Tabifies the current line, causing all spaces at the start of the line (prior to any text) to be converted to tabs where possible.
Untabify Selected Lines	Untabifies the current line, causing all tabs at the start of the line (prior to any text) to be converted to spaces.
Make Uppercase	Converts all letters within the selection to uppercase.
Make Lowercase	Converts all letters within the selection to lowercase.
Toggle Character Casing	Toggles the character cases (uppercase / lowercase) of all letters within the selection.
Capitalize	Capitalizes the first character of every word within the selection.
Delete Horizontal Whitespace	Deletes any excess white space either side of the cursor position, leaving only one whitespace character remaining. If there the cursor is within a word or not surrounded by whitespace, this operation will have no effect.
Trim Trailing Whitespace	Deletes any trailing whitespace that appears after the last non-whitespace character on the cursor line.
Delete Line	Completely delete the current cursor line.
Duplicate Line	Duplicates the cursor line, inserting a copy of the line immediately after the cursor line.
Delete Blank Lines	Deletes the line at the cursor if it is empty or contains only whitespace.
Break Settings	The following cascade menus are displayed to set the break-related event.
Set Hardware Break	Sets a breakpoint (Hardware Break event) to the line at the caret position (see "2.9.3 Stop the program at the arbitrary position (breakpoint)")Note 1.
Set Software Break **[Full-spec emulator] [E1][E20]**	Sets a breakpoint (Software Break event) to the line at the caret position (see "2.9.3 Stop the program at the arbitrary position (breakpoint)")Note 1.
Set Combination Break	In this product version, this item is not supported.
Set Read Break to	Sets a break event with read access condition to the line at the caret or the selected variable (global variable, static variable inside functions, or file-internal static variable)/I/O register (see "2.9.4 Stop the program at the arbitrary position (break event)").
Set Write Break to	Sets a break event with write access condition to the line at the caret or the selected variable (global variable, static variable inside functions, or file-internal static variable)/I/O register (see "2.9.4 Stop the program at the arbitrary position (break event)").
Set R/W Break to	Sets a break event with read/write access condition to the line at the caret or the selected variable (global variable, static variable inside functions, file-internal static variable)/I/O register (see "2.9.4 Stop the program at the arbitrary position (break event)").
Break Option	Opens the Property panel to set the break function.

Trace Settings	The following cascade menus are displayed to set the trace-related event.
Start Tracing	Sets a trace start event to start collecting the trace data when the line at the caret is executed (see "2.12.3 Collect execution history in the arbitrary section")Note 1, 2.
Stop Tracing	Sets a trace end event to stop collecting the trace data when the line at the caret is executed (see "2.12.3 Collect execution history in the arbitrary section")Note 1, 2.
Record Reading Value	Sets a Point Trace event to record the access value as the trace data when a variable at the caret or the selected variable (global variable, static variable inside functions, file-internal static variable) /I/O register is read accessed (see "2.12.4 Collect execution history only when the condition is met [Simulator]").
Record Writing Value	Sets a Point Trace event to record the access value as the trace data when a variable at the caret or the selected variable (global variable, static variable inside functions, file-internal static variable) /I/O register is write accessed (see "2.12.4 Collect execution history only when the condition is met [Simulator]").
Record R/W Value	Sets a Point Trace event to record the access value as the trace data when a variable at the caret or the selected variable (global variable, static variable inside functions, file-internal static variable) /I/O register is read/ write accessed (see "2.12.4 Collect execution history only when the condition is met [Simulator]").
Show Trace Result	Opens the [Trace] and displays the acquired trace data.
Trace Settings	Opens the Property panel to set the trace function. Note that this item is disabled the trace function is in operation.
Timer Settings	The following cascade menus are displayed to set the timer-related event (see "2.13.2 Measure execution time in the arbitrary section [Simulator]").
Start timer	Sets a timer start event to start measuring the execution time of the program when an instruction of an address at the caret position is executedNote 1, 3.
Set Timer *n*	Specify a channel (*n*: 1 to 8) in which a timer start event is set.
Stop timer	Sets a timer end event to stop measuring the execution time of the program when an instruction of an address at the caret position is executedNote 1, 3.
Set Timer *n*	Specify a channel (*n*: 1 to 8) in which a timer stop event is set.
View Result of Timer	Opens the Events panel and displays only timer-related events.
Clear Coverage Information **[Simulator]**	Clears all the coverage measurement results currently being stored in the debug tool.
Save Source Mixed Data As...	Opens the Save As dialog box to newly save the contents of the currently editing Editor panel. Note that this item is enabled only when the Editor panel is in the mixed display mode.

Notes 1. A message is displayed if these items are selected when the downloaded load module file is older than the opened source file.

2. [Simulator]
The [Use trace function] property in the [Trace] category on the Property panel is automatically set to [Yes].

3. [Simulator]
The [Use timer function] property in the [Timer] [Simulator] category on the Property panel is automatically set to [Yes].

Memory panel

This panel is used to display the contents of the memory and change the memory value (see "2.10.1 Display/change the memory").

Furthermore, the contents of data flash memory (including ID tag) can be displayed and changed when the selected microcontroller incorporates the data flash memory.

Up to a maximum of four of these panels can be opened. Each panel is identified by the names "Memory1", "Memory2", "Memory3", and "Memory4" on the titlebar.

The display contents are automatically updated when the value of the memory changes after a program is executed (when the execution is done in steps, the display is updated after each step). In addition, by enabling the Real-time display update function, it is also possible to update the display contents in real-time even while a program is being executed.

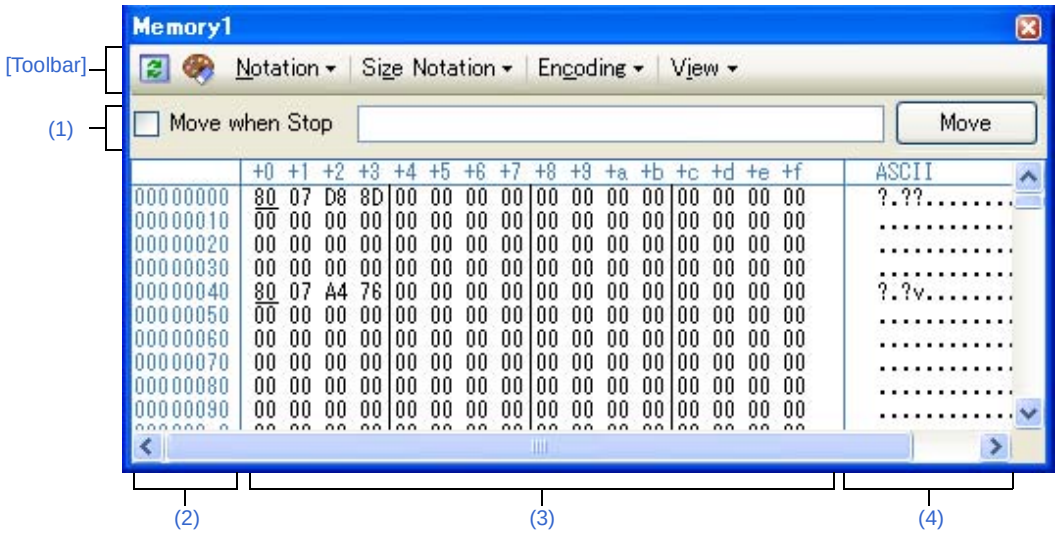
This panel appears only when connected to the debug tool.

Caution CPU reset may be generated depending on the selected microcontroller if you change the memory value in the data flash area.

Remarks 1. This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

2. You can set the scroll range of the vertical scroll bar on this panel via the Scroll Range Settings dialog box which is opened by clicking the button from [View] on the toolbar.

Figure A-25. Memory Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Memory panel-dedicated items)]
- [[Edit] menu (Memory panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [Memory] >> [Memory*1-4*].

[Description of each area]

(1) Display position specification area

It is possible to specify the display start position of the memory contents by specifying an address expression.
Specify the following items.

(a) Specify an address expression

Directly input the address expression of the memory value address to display in the text box. You can specify an input expression with up to 1024 characters. The result of the expression is treated as the display start position address.
Note that an address value greater than the microcontroller address space cannot be specified.
In addition, an address value greater than the value expressed within 32 bits cannot be specified.

Remarks 1. A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

2. If the specified address expression is the symbol and its size can be recognized, everything from the start address to the end address of that symbol is displayed selected.

(b) Specify automatic/manual evaluation of the address expression

The timing to change the display start position can be determined by specifying in the [Move when Stop] check box and the [Move] button.

[Move when Stop]	☑	The caret is moved to the address which is automatically calculated from the address expression after the program is stopped.
	☐	The address expression is not automatically evaluated after the program is stopped. Click the [Move] button to manually evaluate the address expression.
[Move] button		When the [Move when Stop] check box is not checked, click this button to evaluate the address expression and move the caret to the result address of the evaluation.

(2) Address area

The address of the memory is displayed (hexadecimal number notation fixing).
The display starts from address 0x0 by default. However, an offset value of the start address can be set via the Address Offset Settings dialog box that is opened by selecting [Address Offset Value Settings...] from the context menu.
The address width corresponds to the one in memory space of the specified microcontroller in the project.
This area cannot be edited.

Caution The offset value that have been set is automatically changed in accordance with the number of view columns in the Memory value area.

(3) Memory value area

The value of the memory is displayed and changed.
Specification of the display notation, display width of memory values or the number of view columns is performed by selecting the buttons on the toolbar or [Notation]/[Size Notation]/[View] from the context menu (see "(2) Change display format of values").
The meanings of the marks and colors displayed as memory values are as follows (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box):

Display Example (Default)			Description	
00	Character color	Blue	Memory value that the user is changing Press the [Enter] key to write to the target memory.	
	Background color	Standard color		
00 (Under line)	Character color	Standard color	Memory value of the address whose symbol has been defined. Watch-expressions can be registered to this memory (see "(f) Registering watch-expressions").	
	Background color	Standard color		
00	Character color	Brown	Memory value that has been changed because of the execution of a program[Note] To reset the highlighting, select the button on the toolbar.	
	Background color	Cream		
00	Character color	Pink	Memory value for which the Real-time display update function is being operated	
	Background color	Standard color		
00	Character color	Standard color	Read/Fetch	Current access condition of the memory value when the Real-time display update function is being operated
	Background color	Palegreen		
00	Character color	Standard color	Write	
	Background color	Orange		
00	Character color	Standard color	Read and Write	
	Background color	Paleturquoise		
00	Character color	Gray	Memory value of the read-protected area	
	Background color	Standard color		
??	Character color	Gray	Areas not memory-mapped	
	Background color	Standard color		
--	Character color	Gray	Areas not rewritable (e.g. I/O register area/I/O protection area) or when acquisition of memory values failed	
	Background color	Standard color		
**	Character color	Standard color	When display is specified for other than the real-time display update area during program execution or when acquisition of memory values failed	
	Background color	Standard color		

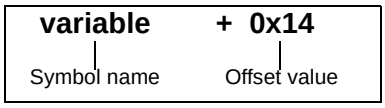
Note Just before execution of a program, only the memory value in the address range for which the Memory panel had been displayed becomes the target.
In addition, the value is not highlighted if it is same for before and after the execution of the program.

Caution The number of view columns is automatically changed in accordance with the set value of [Size Nortation] of the context menu.

This area is provided with the following functions.

(a) Pop-up display

The following contents are pop-up displayed based on the nearest existing symbol forward from the address the mouse is designating when hovering the mouse cursor over the memory value.
Note that if there is no symbol information (the underlining is non-display), no pop-up display is done.



Symbol name	Indicates the name of the symbol.
Offset value	When a symbol has not been defined for the addresses, the offset value from the nearest symbol exists forward is displayed (hexadecimal number notation fixing).

(b) Real-time display update function

Using the real-time display update function allows you to display/modify the value of the memory contents not only while the program is stopped, but also in execution.

See "(4) Display/modify the memory contents during program execution" for details on the real-time display update function.

(c) Changing memory values

Directly edit from the keyboard after moving the caret to the memory value to be edited.

The color of the memory value changes when it is in editing. Press the [Enter] key to write the edited value to the target memory (if the [Esc] key is pressed before the [Enter] key is pressed, the editing is cancelled).

See "(3) Modify the memory contents" for details on the method for changing the memory value.

(d) Searching/initializing memory value

The Memory Search dialog box is opened to search the memory contents in the specified address range by selecting [Find...] from the context menu (see "(5) Search the memory contents").

In addition, the Memory Initialize dialog box is opened to change the memory contents collectively in the specified address range by selecting [Fill...] from the context menu (see "(6) Modify the memory contents in batch (initialize)").

(e) Copying and pasting

By selecting a range of memory values with the mouse, the contents of the range can be copied to the clipboard as a character string, and these contents can be pasted to the caret position.

These operations are performed by selecting from the context menu or selecting from the [Edit] menu.

However, the paste operation is possible only when the character string to be pasted and the display notation (radix and size) of the area match.

If the display notation does not match, a message is displayed.

The following table shows the character code and character strings that can be used in this area (a message will appear when a character string other than those listed here is pasted).

Character code	ASCII
Character string	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, A, B, C, D, E, F

(f) Registering watch-expressions

A memory value with underline indicates that a symbol has been defined in the address, and its symbol can be registered as a watch-expression.

After selecting the memory value or placing the caret on the memory value, the symbol name of the address is registered in the Watch panel (Watch1) as a watch-expression by selecting [Register to Watch1] from the context menu.

Caution A memory value without underline cannot be registered as a watch-expression.

(g) Saving the contents of memory values

The Data Save dialog box can be opened by selecting the [File] menu >> [Save Memory Data As...], and the contents of this panel can be saved in a text file (*.txt) or CSV file (*.csv).

See "(7) Save the memory contents" for details on the method for saving the contents of memory values.

(4) Character strings area

Memory values converted into character code are displayed.
The character code can be specified by selecting [Encoding] from the toolbar or context menu (ASCII code is selected by default). Furthermore, in this area, memory values converted into a floating-point value can be displayed as character strings. To do this, select the following item from [Encording] of the context menu.

Item	Display Format		Size
Float	Single-precision floating-point value		32-bit
	Numeric value	<*sign*><*mantissa*>e<*sign*><*exponent*>	
	Infinite number	Inf, and -Inf	
	Not a number	NaN	
	Example	+ 1.234567e+123	
Double	Double-precision floating-point value		64-bit
	Numeric value	<*sign*><*mantissa*>e<*sign*><*exponent*>	
	Infinite number	Inf, and -Inf	
	Not a number	NaN	
	Example	+ 1.2345678901234e+123	
Float Complex	Complex number of single-precision floating-point		64-bit
	<*Single-precision floating-point value*> <*Single-precision floating-point value*> * I		
Double Complex	Complex number of double-precision floating-point		128-bit
	<*Double-precision floating-point value*> <*Double-precision floating-point value*> * I		
Float Imaginary	Imaginary number of single-precision floating-point		32-bit
	<*Single-precision floating-point value*> * I		
Double Imaginary	Imaginary number of double-precision floating-point		64-bit
	<*Double-precision floating-point value*> * I		

Caution Nothing is displayed when the minimum size of a character code or a floating-point value is greater than "the number of bytes of display width of memory values" x "the number of bytes of the number of view columns".

This area is provided with the following functions.

(a) Changing character strings

Directly edit from the keyboard after moving the caret to the character string to be edited.
The color of the character string changes when it is in editing. Press the [Enter] key to write the edited value to the target memory (if the [Esc] key is pressed before the [Enter] key is pressed, the editing is cancelled).

Caution Character strings displayed as floating-point values cannot be searched.

(b) Searching character strings

The Memory Search dialog box is opened to search for character strings by selecting [Find...] from the context menu (see "(5) Search the memory contents").

(c) Copying and pasting

By selecting a range of character strings with the mouse, the contents of the range can be copied to the clipboard as a character string, and these contents can be pasted to the caret position.
These operations are performed by the selecting from the context menu or selecting from the [Edit] menu.
However, the paste operation is possible only when [ASCII] has been selected as the character code. If other than [ASCII] is selected, a message is displayed.

[Toolbar]

	Acquires the latest data from the debug tool, and updates the contents of this panel.
	Resets highlighting of values that have been changed by executing a program. This item is disabled during execution of a program.
Notation	The following buttons to change the notation of memory values are displayed.
	Displays memory values in hexadecimal number (default).
	Displays memory values in signed decimal number.
	Displays memory values in unsigned decimal number.
	Displays memory values in octal number.
	Displays memory values in binary number.
Size Notation	The following buttons to change the notation of sizes of memory values are displayed.
	Displays memory values in 4-bit width.
	Displays memory values in 8-bit width (default).
	Displays memory values in 16-bit width. Values are converted depending on the endian of the target memory area.
	Displays memory values in 32-bit width. Values are converted depending on the endian of the target memory area.
	Displays memory values in 64-bit width. Values are converted depending on the endian of the target memory area.
Encoding	The following buttons to change the encoding of character strings are displayed.
	Displays character strings in ASCII code (default).
	Displays character strings in Shift-JIS code.
	Displays character strings in EUC-JP code.
	Displays character strings in UTF-8 code.
	Displays character strings in UTF-16 code.
	Displays character strings as a single-precision floating-point value.
	Displays character strings as a double-precision floating-point value.
	Displays character strings as a complex number of single-precision floating-point.
	Displays character strings as a complex number of double-precision floating-point.
	Displays character strings as an imaginary number of single-precision floating-point.
	Displays character strings as an imaginary number of double-precision floating-point.
View	The following buttons to change the display format are displayed.

	Opens the Scroll Range Settings dialog box to set the scroll range for this panel.
Column Number Settings...	Opens the Column Number Settings dialog box to set the number of view columns in the Memory value area.
Address Offset Value Settings...	Opens the Address Offset Settings dialog box to set an offset value for addresses displayed in the Address area.

[[File] menu (Memory panel-dedicated items)]

The following items are exclusive for the [File] menu in the Memory panel (other items are common to all the panels).
Note that all these items are disabled during execution of a program.

Save Memory Data	Overwrites the contents of this panel to the previously saved text file (*.txt)/CSV file (*.csv) (see "(g) Saving the contents of memory values"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Memory Data As...].
Save Memory Data As...	Opens the Data Save dialog box to newly save the contents of this panel to the specified text file (*.txt)/CSV file (*.csv) (see "(g) Saving the contents of memory values").

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

The following items are exclusive for [Edit] menu in the Memory panel (all other items are disabled).
Note that all these items are disabled during execution of a program.

Copy	Copies the contents of the selected range to the clipboard as character string(s).
Paste	Pastes the character string(s) copied in the clipboard to the caret position. To the memory value area: See "(e) Copying and pasting". To the character strings area: See "(c) Copying and pasting".
Find...	Opens the Memory Search dialog box. The search is operated either in the Memory value area or the Character strings area, in which a caret is.

[Context menu]

Register to Watch1	Registers the symbol at the caret to the Watch panel (Watch1). At this time, since it is registered as a variable name, the symbol name that is displayed changes depending on the scope. Note that this item is disabled when no symbol has been defined in the address corresponding to the memory value at the caret position (see "(f) Registering watch-expressions").
Find...	Opens the Memory Search dialog box. The search is operated either in the Memory value area or the Character strings area (unless the floating-point value display is selected), in which a caret is. This item is disabled during execution of a program.
Fill...	Opens the Memory Initialize dialog box.
Refresh	Acquires the latest data from the debug tool, and updates the contents of this panel.
Copy	Copies the contents of the selected range to the clipboard as character string(s). This item is disabled during execution of a program.

Paste	Pasts the character string(s) copied in the clipboard to the caret position. This item is disabled during execution of a program. To the memory value area: See "(e) Copying and pasting". To the character strings area: See "(c) Copying and pasting".
Notation	The following cascade menus are displayed to specify the notation of memory values.
Hexadecimal	Displays memory values in hexadecimal number (default).
Signed Decimal	Displays memory values in signed decimal number.
Unsigned Decimal	Displays memory values in unsigned decimal number.
Octal	Displays memory values in octal number.
Binary	Displays memory values in binary number.
Size Notation	The following cascade menus are displayed to specify the notation of sizes of memory values.
4 Bits	Displays memory values in 4-bit width.
1 Byte	Displays memory values in 8-bit width (default).
2 Bytes	Displays memory values in 16-bit width. Values are converted depending on the endian of the target memory area.
4 Bytes	Displays memory values in 32-bit width. Values are converted depending on the endian of the target memory area.
8 Bytes	Displays memory values in 64-bit width. Values are converted depending on the endian of the target memory area.
Encoding	The following cascade menus are displayed to specify the display format in the character strings area.
ASCII	Displays character strings in ASCII code (default).
Shift_JIS	Displays character strings in Shift-JIS code.
EUC-JP	Displays character strings in EUC-JP code.
UTF-8	Displays character strings in UTF-8 code.
UTF-16	Displays character strings in UTF-16 code.
Float	Displays character strings as a single-precision floating-point value.
Double	Displays character strings as a double-precision floating-point value.
Float Complex	Displays character strings as a complex number of single-precision floating-point.
Double Complex	Displays character strings as a complex number of double-precision floating-point.
Float Imaginary	Displays character strings as an imaginary number of single-precision floating-point.
Double Imaginary	Displays character strings as an imaginary number of double-precision floating-point.
View	The following cascade menus are displayed to specify the display format.
Settings Scroll Range...	Opens the Scroll Range Settings dialog box to set the scroll range for this panel.
Column Number Settings...	Opens the Column Number Settings dialog box to set the number of view columns in the Memory value area.
Address Offset Value Settings...	Opens the Address Offset Settings dialog box to set an offset value for addresses displayed in the Address area.
Highlight Accessed	Highlights memory values that have changed by execution of a program if this item is checked (default). This item is disabled during execution of a program.

Periodic Updating	The following cascade menus are displayed to set for the real-time display update function (see "(b) Real-time display update function").
Periodic Updating Options	Opens the Property panel to set for the real-time display update function.

Disassemble panel

This panel is used to display the results of disassembling the contents of the memory (disassembled text), and execute line assembly (see "2.6.4 Perform line assembly").

Furthermore, the instruction level debugging (see "2.8.3 Execute programs in steps") and the code coverage measurement result display **[Simulator]** (see "2.14 Measure Coverage [Simulator]") can be performed in this panel.

Up to a maximum of four of these panels can be opened. Each panel is identified by the names "Disassemble1", "Disassemble2", "Disassemble3" and "Disassemble4" on the titlebar.

The source text in the source file corresponding to the code data can also be displayed by setting to the Mixed display mode (default).

This panel appears only when connected to the debug tool.

Caution A step execution is performed in instruction level units when the focus is in this panel (see "2.8.3 Execute programs in steps").

Remarks 1. You can set the scroll range of the vertical scroll bar on this panel via the Scroll Range Settings dialog box which is opened by clicking the button from [View] on the toolbar.

2. You can print the current screen image of this panel by selecting [Print...] from the [File] menu.

3. This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

Figure A-26. Disassemble Panel (When Mixed Display Mode Is Selected)

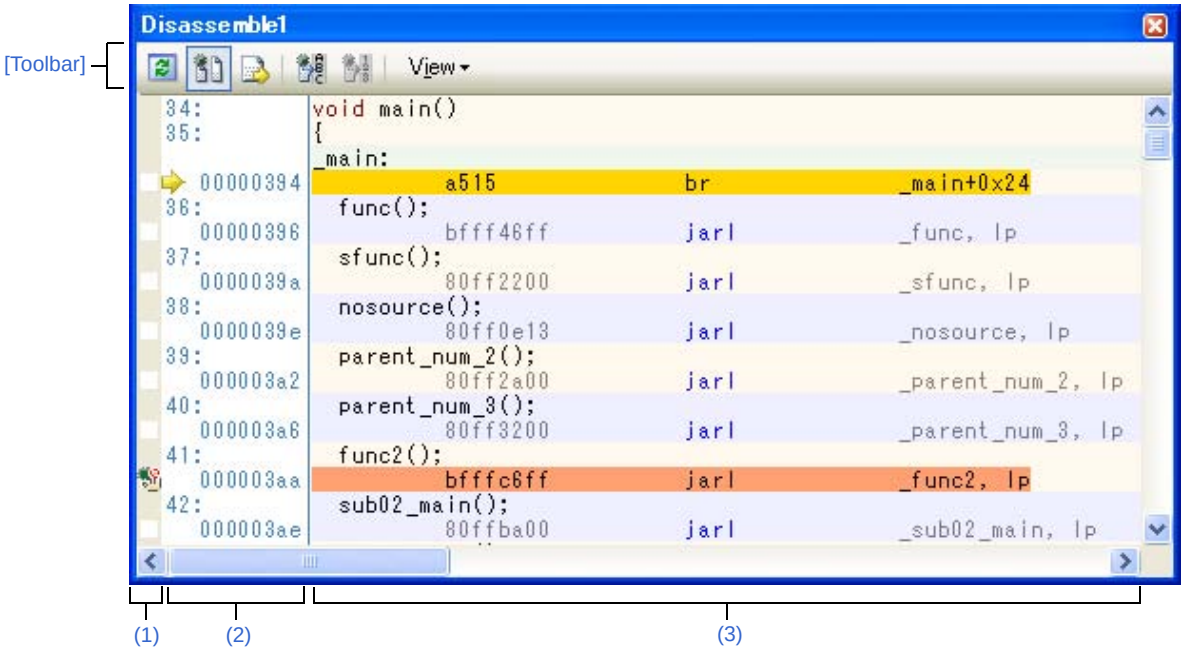


Figure A-27. Disassemble Panel (When Disassemble Display Mode Is Selected)

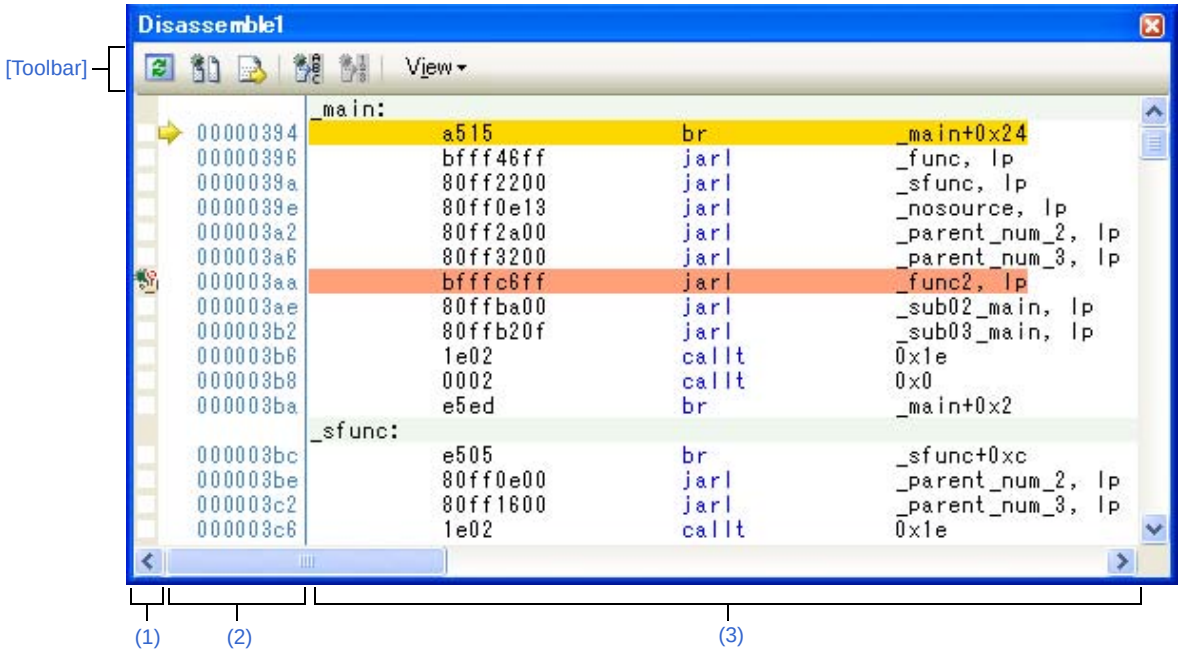
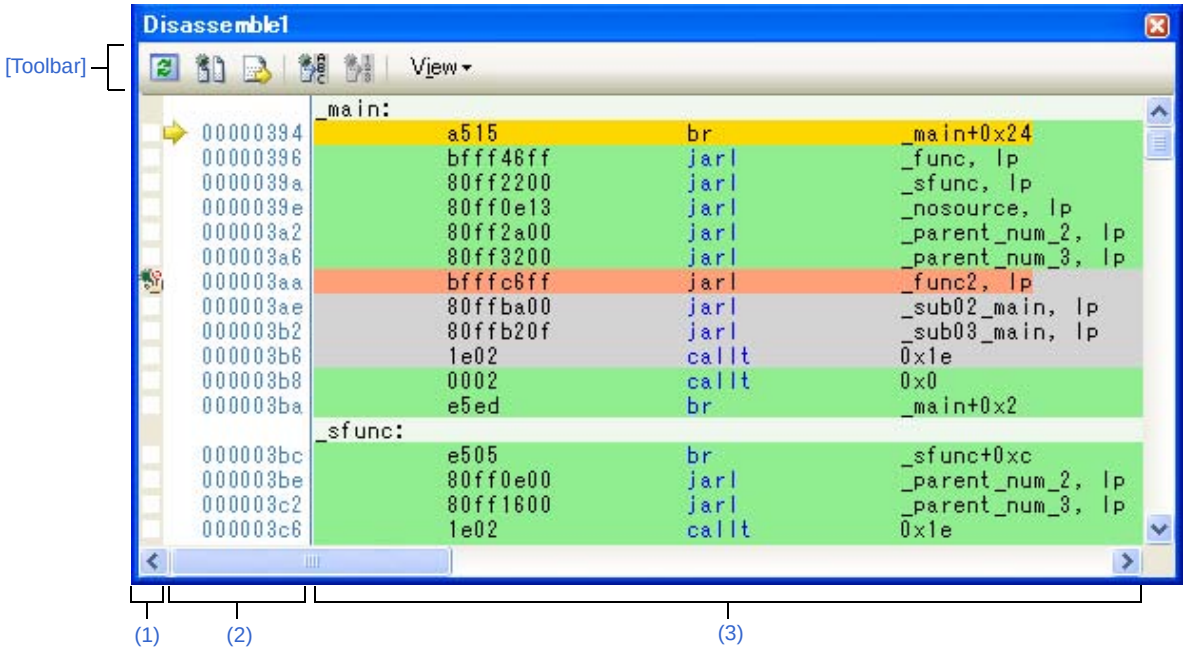


Figure A-28. Disassemble Panel (When Code Coverage Measurement Result Is Displayed) [Simulator]



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Disassemble panel-dedicated items)]
- [[Edit] menu (Disassemble panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [Disassemble] >> [Disassemble*1 - 4*].

[Description of each area]

(1) Event area

The lines for which events can be set are shown with the background color in white (this mean that events cannot be set for those lines whose background color in gray).
In addition, the Event mark corresponding to an event that has been currently set is displayed.
This area is provided with the following functions.

(a) Setting/deleting breakpoints

By clicking where you want to set a breakpoint with the mouse, the breakpoint can be set easily.
The breakpoint is set to the instruction at the start address of the clicked line.
Once the breakpoint is set, the Event mark is displayed at the line that is set. In addition, the detailed information about the set breakpoint is reflected in the Events panel.
When this operation is performed at a place where any one of the event marks is already being displayed, that event is deleted and the setting of breakpoints cannot be done.
Note that the setting of events can be done only for those lines where the background color is shown in white.
See "2.9.3 Stop the program at the arbitrary position (breakpoint)" for details on how to set the breakpoint.

(b) Changes event status

Event status can be changed from the following menu displayed by right-clicking the event mark.

Enable Event	Changes the selected event state to a Valid state. Event occurs when the specified condition is met. When the event mark ( ) which indicates that multiple events have been set is selected, all of the events that have been set are enabled.
Disable Event	Changes the selected event state to an Invalid state. Event does not occur when the specified condition is met. When the event mark ( ) which indicates that multiple events have been set is selected, all of the events that have been set are disabled.
Delete Event	Deletes the selected event. When the event mark ( ) which indicates that multiple events have been set is selected, all of the events that have been set are deleted.
View Event Detailed Setup	Opens the Events panel to display the detailed information of the selected event.

(c) Pop-up display

By hovering the mouse cursor over the Event mark, the name of the event, the detailed information for the event and the comments added to the event are pop-up displayed.
When multiple events have been set in the applicable place, information for each event, up to a maximum of three events, is listed and displayed.

(2) Address area

The address per line to start disassembling is displayed (hexadecimal number notation fixing).
In addition, the current PC mark ( ) that corresponds to the current PC position (PC register value) is displayed.
The address width corresponds to the one in memory space of the specified microcontroller in the project.

For the source text line in the Mixed display mode, line numbers (*xxx*:) in the source file correspond to the start address are displayed.

Remark The current PC mark changes from ➡ to ➪ when the position of the current PC is invalid (e.g., in cases where the current target core for debugging differs from the one selected at the time the program was stopped).

This area is provided with the following functions.

(a) Pop-up display

By hovering the mouse cursor over a address or line number, the following information is pop-up displayed.

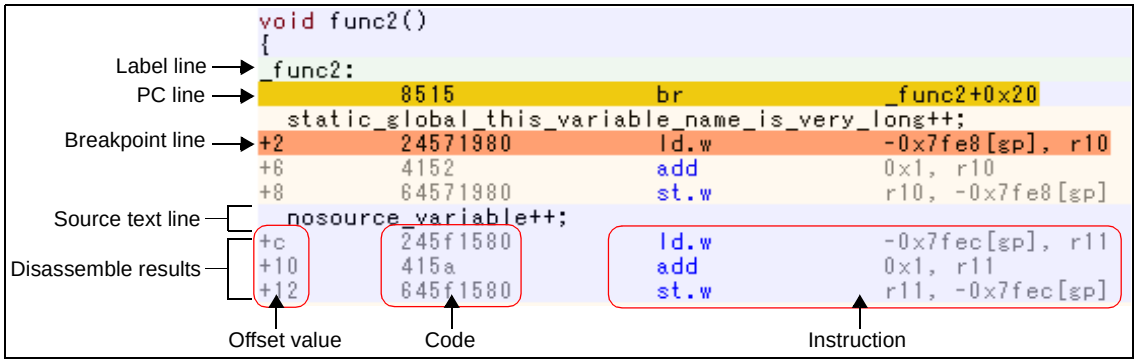
Address	Format: *<Label name>* + *<Offset value>* Example1: main + 0x10 Example2: sub function + 0x20
Source line number	Format: *<Load module name>*[Note]$*<File name>* # *<Line number>* Example1: test1.out$main.c#40 Example2: main.c#100

Note *<Load module name>* is displayed only when multiple load modules have been downloaded to the debug tool.

(3) Disassemble area

The results of disassembling are displayed next to the corresponding source text as follows:

Figure A-29. Display Contents of Disassemble Area (In Case of Mixed Display Mode)



Label line		The label is displayed when a label is defined for the address, and its corresponding line is shown highlighted in lightgreen.
PC line		A line corresponding to an address of the current PC (PC register value) is shown highlighted[Note 1].
Breakpoint line		A line at which a breakpoint is set is shown highlighted[Note 1].
Source text line		The source text corresponding to the code data is displayed[Note 2].
Disassemble results	Offset value	The offset value from the nearest label is displayed when a label is defined for the address[Note 3].
	Code	The code that is the target of disassembly is displayed in hexadecimal number.
	Instruction	Instruction is displayed as the result of disassembling. The mnemonics are shown highlighted in blue.

Notes 1. The highlighting color depends on the configuration in the [General - Font and Color] category of the Option dialog box.

2. The source text can be set to non-display by clicking the  button (toggle) on the toolbar or removing the check for [Mixed Display] from the context menu (this option is checked by default).

3. Offset values are not displayed by default. They can be displayed by clicking the  button on the toolbar or selecting [Show Offset] from the context menu.

This area is provided with the following functions.

(a) Line assembly

Instructions and code displayed in this panel can be edited (line assembly).
See "2.6.4 Perform line assembly" for details on how to operate it.

(b) Program execution by instruction level

Execution can be controlled at the instruction level unit by step executing a program in a state where there is a focus on this panel.
See "2.8.3 Execute programs in steps" for details on how to operate it.

(c) Setting of various events

Various events can be set to the addresses/lines where the caret currently exists by selecting [Bread Settings], [Trace Settings] or [Timer Settings] from the context menu.
The corresponding Event mark is displayed in the Event area when an event is set. In addition, the detailed information about the set event is reflected in the Events panel.
+Note, however, that the setting of events can be done only for those lines where the background color is shown in white in the event area.
See the following for details on how to set events.

- "2.9.4 Stop the program at the arbitrary position (break event)"
- "2.9.5 Stop the program with the access to variables/I/O registers"
- "2.12.3 Collect execution history in the arbitrary section"
- "2.12.4 Collect execution history only when the condition is met [Simulator]"
- "2.13.2 Measure execution time in the arbitrary section [Simulator]"
- "2.15.1 Inset printf"

Remark A breakpoint can be set or deleted easily in the Event area as well (see "(a) Setting/deleting breakpoints").

(d) Registering a watch-expression

Variable names of C language, CPU registers, I/O registers, and assembler symbols can be registered in the Watch panel as watch-expressions.
See "(1) Register a watch-expression" for details on how to operate it.

(e) Moving to symbol definition place

By clicking the  button on the toolbar or selecting [Go to Symbol] from the context menu in a state where the caret has been moved to a instruction that has referenced a symbol, the caret position is moved to the address where the symbol at the caret position has been defined.
In addition, when following on this operation you click on the  button on the toolbar or select [Back to *Address*] from the context menu, the caret position is returned to the instruction that has referenced a symbol before the caret was moved (the address value of the instruction that has referenced a symbol is displayed in *Address*).

(f) Jump to source line and memory

Selecting [Jump to Source] from the context menu will open the Editor panel with the caret moved to the source line corresponding to the address at the current caret position (if the Editor panel is already open, the screen will jump to the panel).

In addition, similarly, selecting [Jump to Memory] will open the Memory panel (Memory1) with the caret moved to the memory value corresponding to the address at the current caret position (if the Memory panel (Memory1) is already open, the screen will jump to the panel).

(g) Code coverage measurement result display [Simulator]

When the coverage function is valid, lines corresponding to the specified coverage measurement area are shown highlighted based on the code coverage measurement result that is acquired by executing the program.

See "2.14 Measure Coverage [Simulator]" for details on the coverage measurement.

(h) Saving the contents of disassembled data

The Data Save dialog box can be opened by selecting the [File] menu >> [Save Disassemble Data As…], and the contents of this panel can be saved in a text file (*.txt) or CSV file (*.csv).

See "(5) Save the disassembled text contents" for details on the method for saving the contents of disassembled data.

[Toolbar]

	Acquires the latest data from the debug tool, and updates the contents of this panel.
	Toggles between the mixed display mode (default) and disassemble display mode, as the display mode of this panel (see "(1) Change display mode").
	Specifies the caret position so that it follows the current PC value.
	Moves the caret to the define position of the selected symbol.
	Moves the caret to the position (*address*) immediately before it is moved with the  button.
View	The following buttons to set the display contents in the disassemble area are displayed.
	Displays the offset value of the label. The offset value from the nearest label is displayed when a label is defined for the address.
	Displays the address value in the format "symbol + offset value" (default). Note that when a symbol has been defined as the address value, only the symbol is displayed.
	Displays the name of the register by its function name (default).
	Displays the name of the register by its absolute name.
	Opens the Scroll Range Settings dialog box to set the scroll range for this panel.

[[File] menu (Disassemble panel-dedicated items)]

The following items are exclusive for the [File] menu in the Disassemble panel (other items are common to all the panels).

Note that all these items are disabled during execution of a program.

Save Disassemble Data	Overwrites the contents of the disassembling to the previously saved text file (*.txt)/CSV file (*.csv) (see "(h) Saving the contents of disassembled data"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Disassemble Data As…].

Save Disassemble Data As…	Opens the Data Save dialog box to newly save the contents of the disassembling to the specified text file (*.txt)/CSV file (*.csv) (see "(h) Saving the contents of disassembled data").
Print...	Opens the Print Address Range Settings dialog box for printing the contents of this panel.

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

The following items are exclusive for the [Edit] menu in the Disassemble panel (all other items are disabled).

Copy	When a line is selected, copies the contents of the selected line to the clipboard as a character string. In the case of the edit mode, copies the selected character string to the clipboard.
Rename	Changes to the edit mode to edit the instruction/code at the caret position (see "2.6.4 Perform line assembly"). This item is disabled during execution of a program.
Find…	Opens the Find and Replace dialog box with selecting the [Find in Files] tab.
Replace…	Opens the Find and Replace dialog box with selecting the [Replace in Files] tab.
Move…	Opens the Go to the Location dialog box to move the caret to the specified address.

[Context menu]

(1) Disassemble area and Address area
(2) Event area [Full-spec emulator][E1][E20]

(1) Disassemble area and Address area

Register to Watch1	Registers the selected character string or the word at the caret position to the Watch panel (Watch1) as a watch-expression (the judgment of the word depends on current build tool). At this time, since it is registered as a variable name, the symbol name that is displayed changes depending on the scope.
Register Action Event...	Opens the Action Events dialog box to set an action event to the address at the caret position.
Go to Here	Executes the program from the address indicated by the current PC value to the address corresponding to the line at the caret position. This item is disabled during execution of a program/[Build & Download].
Set PC to Here	Sets the address of the line at the current caret position to the current PC value. This item is disabled during execution of a program/[Build & Download].
Move…	Opens the Go to the Location dialog box to move the caret to the specified address.
Go to Symbol	Moves the caret to the define position of the selected symbol.
Back to *Address*	Moves the caret to the position (*address*) immediately before it is moved by [Go to Symbol]. Note that this item is disabled when no symbol name is displayed in the address.
Break Settings	The following cascade menus are displayed to set the break-related event.

Set Hardware Break	Sets a breakpoint (Hardware Break event to the address at the caret position (see "2.9.3 Stop the program at the arbitrary position (breakpoint)").
Set Software Break **[Full-spec emulator] [E1][E20]**	Sets a breakpoint (Software Break event) to the address at the caret position (see "2.9.3 Stop the program at the arbitrary position (breakpoint)").
Set Combination Break	In this product version, this item is not supported.
Set Read Break to	Sets a break event with read access condition to a variable at the caret or a selected variable (global variable/static variable inside functions/file-internal static variable)/I/O register (see "2.9.4 Stop the program at the arbitrary position (break event)").
Set Write Break to	Sets a break event with write access condition to a variable at the caret or a selected variable (global variable/static variable inside functions/file-internal static variable)/I/O register (see "2.9.4 Stop the program at the arbitrary position (break event)").
Set R/W Break to	Sets a break event with read/write access condition to a variable at the caret or a selected variable (global variable/static variable inside functions/file-internal static variable)/I/O register (see "2.9.4 Stop the program at the arbitrary position (break event)").
Break Option	Opens the Property panel to set the break function.
Trace Settings	The following cascade menus are displayed to set the trace-related event.
Start Tracing	Sets a trace start event to start collecting the trace data when an instruction of an address at the caret position is executed (see "2.12.3 Collect execution history in the arbitrary section")Note 1.
Stop Tracing	Sets a trace end event to stop collecting the trace data when an instruction of an address at the caret position is executed (see "2.12.3 Collect execution history in the arbitrary section")Note 1.
Record Reading Value	Sets a Point Trace event to record the access value as the trace data when a variable at the caret or the selected variable (global variable, static variable inside functions, file-internal static variable) or I/O register is read accessed (see "2.12.4 Collect execution history only when the condition is met [Simulator]").
Record Writing Value	Sets a Point Trace event to record the access value as the trace data when a variable at the caret or the selected variable (global variable, static variable inside functions, file-internal static variable) or I/O register is write accessed (see "2.12.4 Collect execution history only when the condition is met [Simulator]").
Record R/W Value	Sets a Point Trace event to record the access value as the trace data when a variable at the caret or a selected variable (global variable/static variable inside functions/file-internal static variable)/I/O register is read/write accessed (see "2.12.4 Collect execution history only when the condition is met [Simulator]").
Show Trace Result	Opens the Trace panel and displays the acquired trace data.
Trace Settings	Opens the Property panel to set the trace function.
Timer Settings	The following cascade menus are displayed to set the timer-related event (see "2.13.2 Measure execution time in the arbitrary section [Simulator]").
Start timer	Sets a timer start event to start measuring the execution time of the program when an instruction of an address at the caret position is executedNote 2.
Set Timer *n*	Specify a channel (*n*: 1 to 8) in which a timer start event is set.
Stop timer	Sets a timer end event to stop measuring the execution time of the program when an instruction of an address at the caret position is executedNote 2.
Set Timer *n*	Specify a channel (*n*: 1 to 8) in which a timer stop event is set.
View Result of Timer	Opens the Events panel and displays only timer-related events.

Clear Coverage Information **[Simulator]**	Clears all the coverage measurement results currently being stored in the debug tool.
Edit Disassemble	Changes to the edit mode to edit the instruction of the line at the caret position (see "2.6.4 Perform line assembly"). This item is disabled during execution of a program.
Edit Code	Changes to the edit mode to edit the code of the line at the caret position (see "2.6.4 Perform line assembly"). This item is disabled during execution of a program.
View	The following cascade menus to set the display contents in the disassemble area are displayed.
Show Offset	Displays the offset value of the label. The offset value from the nearest label is displayed when a label is defined for the address.
Show Symbol	Displays the address value in the format "symbol + offset value" (default). Note that when a symbol has been defined as the address value, only the symbol is displayed.
Show Function Name	Displays the name of the register by its function name (default).
Show Absolute Name	Displays the name of the register by its absolute name.
Settings Scroll Range...	Opens the Scroll Range Settings dialog box to set the scroll range for this panel.
Mixed Display	Toggles between the mixed display mode (default) and disassemble display mode, as the display mode of this panel (see "(1) Change display mode").
Jump to Source	Opens the Editor panel and jumps to the source line corresponding to the address at the caret position in this panel.
Jump to Memory	Opens the Memory panel (Memory1) and jumps to the memory value corresponding to the address at the caret position in this panel.

Notes 1. [Simulator]
The [Use trace function] property in the [Trace] category on the Property panel is automatically set to [Yes].

2. [Simulator]
The [Use timer function] property in the [Timer] [Simulator] category on the Property panel is automatically set to [Yes].

(2) Event area [Full-spec emulator][E1][E20]

Hardware Break First	The type of break that can be set by a one click operation of the mouse is set as a hardware breakpoint (this is reflected in the setting of the [First using type of breakpoint] property in the [Break] [Full-spec emulator][E1][E20] category on the Property panel).
Software Break First	The type of break that can be set by a one click operation of the mouse is set as a software breakpoint (this is reflected in the setting of the [First using type of breakpoint] property in the [Break] [Full-spec emulator][E1][E20] category on the Property panel).

CPU Register panel

This panel is used to display the contents of the CPU register (program registers/system registers) and change the CPU register values (see "2.10.2 Display/change the CPU register").

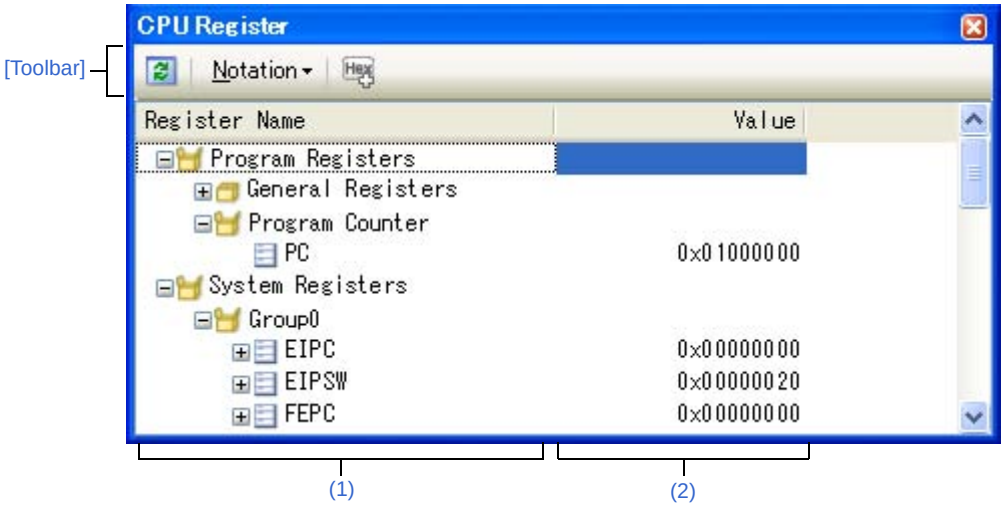
This panel appears only when connected to the debug tool.

Caution When the selected microcontroller supports multi-core, this panel displays/changes the value regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

Remarks 1. This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

2. When the separator line of each area in this panel is double-clicked, the width of the area changes to the shortest possible size that can display the contents of the area.

Figure A-30. CPU Register Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (CPU Register panel-dedicated items)]
- [[Edit] menu (CPU Register panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [CPU Register].

[Description of each area]

(1) [Register Name] area

The types of register are classified as categories (folders), and a list of the respective register names is displayed.

Note that neither category names nor register names can be edited and deleted.

The meanings of the icons are as follows:

	Indicates that the register name belonging to this category is displayed. When you double-click on the icon, or click on the "-" mark, the category is closed and the register name is hidden.
	Indicates that the register name belonging to this category is hidden. When you double-click on the icon, or click on the "+" mark, the category is opened and the register name is displayed.
	Indicates the name of the register. When you double-click on the icon, or click on the "+" or "-" marks, the name of the register part is displayed or hidden.
	Indicates the name of the register part.

This area is provided with the following functions.

(a) Registering a watch-expression

CPU registers/categories can be registered in the Watch panel as watch-expressions.
See "(1) Register a watch-expression" for details on how to operate it.

Remarks 1. When you have registered a watch-expression with a category as the object, all of the CPU registers belonging to that category are registered as watch-expressions.

2. A scope specification is automatically added to a registered watch-expression.

(2) [Value] area

The values of each CPU register are displayed and changed.
The radix of a data value can be selected by the button on the toolbar or the context menu item. In addition, a display format adding the value in hexadecimal number constantly can also be selected as well.
The meanings of the colors of the CPU register values are as follows (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box):

Display Example (Default)			Description
0x0	Character color	Blue	The value of the CPU register that the user is changing Press the [Enter] key to write to the target memory.
	Background color	Standard color	
0x0	Character color	Brown	The value of the CPU register that has been changed because of the execution of a program. The highlighting is rest by executing again the program.
	Background color	Cream	

This area is provided with the following functions.

(a) Changing the CPU register value

To edit the CPU register value, select the value to edit, then change the value directly from the keyboard after clicking again on it (press the [Esc] key to cancel the edit mode).
After you edit the value of the CPU register, it is written to the target memory of the debug tool by pressing the [Enter] key or moving the focus to outside the edit region.

(b) Saving the contents of the CPU register

The Save As dialog box can be opened by selecting the [File] menu >> [Save CPU Register Data As…], and all the contents of this panel can be saved in a text file (*.txt) or CSV file (*.csv).
See "(4) Save the CPU register contents" for details on the method for saving the contents of the CPU register.

[Toolbar]

	Acquires the latest data from the debug tool, and updates the contents of this panel. This item is disabled during execution of a program.
Notation	The following buttons to change the notation of a data value are displayed.
	Displays the value of the selected item (including sub-items) in the default notation (default).
	Displays the value of the selected item (including sub-items) in hexadecimal number.
	Displays the value of the selected item (including sub-items) in signed decimal number.
	Displays the value of the selected item (including sub-items) in unsigned decimal number.
	Displays the value of the selected item (including sub-items) in octal number.
	Displays the value of the selected item (including sub-items) in binary number.
	Displays the character string of the selected item (including sub-items) in ASCII code. If the character size is 2 bytes and above, it is displayed with the characters for each 1 byte arranged side-by-side.
	Displays the value of the selected item in float. Note that when the value is not 4-byte data, displays it in the default notation.
	Displays the value of the selected item in double. Note that when the value is not 8-byte data, displays it in the default notation.
	Adds the value in hexadecimal number enclosing with "()" at the end of the value.

[[File] menu (CPU Register panel-dedicated items)]

The following items are exclusive for the [File] menu in the CPU Register panel (other items are common to all the panels).

Note that all these items are disabled during execution of a program.

Save CPU Register Data	Overwrites the contents of this panel to the previously saved text file (*.txt)/CSV file (*.csv) (see "(b) Saving the contents of the CPU register"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save CPU Register Data As…].
Save CPU Register Data As…	Opens the Save As dialog box to newly save the contents of this panel to the specified text file (*.txt)/ CSV file (*.csv) (see "(b) Saving the contents of the CPU register").

[[Edit] menu (CPU Register panel-dedicated items)]

The following items are exclusive for [Edit] menu in the CPU Register panel (all other items are disabled).

Cut	Deletes the selected character string and copies it to the clipboard. This item becomes valid only when the character string is being edited.
Copy	Copies the selected character string to the clipboard during editing. If a line is selected, copies the register or the category to the clipboard. The copied item can be pasted to the Watch panel.
Paste	Pasts the character string copied in the clipboard to the caret position. This item becomes valid only when the character string is being edited.
Select All	Selects all the items of this panel.
Find…	Opens the Find and Replace dialog box with selecting the [Find in Files] tab.
Replace…	Opens the Find and Replace dialog box with selecting the [Replace in Files] tab.

[Context menu]

Register to Watch1	Registers the selected register or category to the Watch panel (Watch1).
Copy	Copies the selected character string to the clipboard during editing. If a line is selected, copies the register or the category to the clipboard. The copied item can be pasted to the Watch panel.
Notation	The following cascade menus to specify the notation of a data value are displayed.
AutoSelect	Displays the value of the selected item (including sub-items) in the default notation (default).
Hexadecimal	Displays the value of the selected item (including sub-items) in hexadecimal number.
Signed Decimal	Displays the value of the selected item (including sub-items) in signed decimal number.
Unsigned Decimal	Displays the value of the selected item (including sub-items) in unsigned decimal number.
Octal	Displays the value of the selected item (including sub-items) in octal number.
Binary	Displays the value of the selected item (including sub-items) in binary number.
ASCII	Displays the character string of the selected item (including sub-items) in ASCII code. If the character size is 2 bytes and above, it is displayed with the characters for each 1 byte arranged side-by-side.
Float	Displays the value of the selected item in float. Note that when the value is not 4-byte data, displays it in the default notation.
Double	Displays the value of the selected item in double. Note that when the value is not 8-byte data, displays it in the default notation.
Include Hexadecimal Value	Adds the value in hexadecimal number enclosing with "()" at the end of the value.

IOR panel

This panel is used to display the contents of the I/O register and change the I/O register values (see "2.10.3 Display/change the I/O register").

This panel appears only when connected to the debug tool.

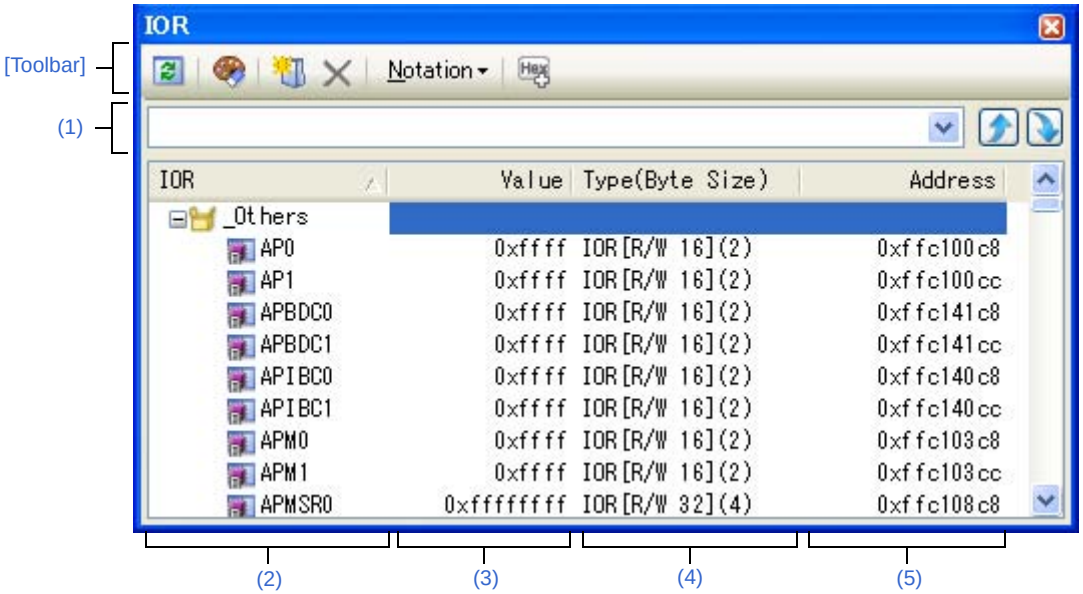
Cautions 1. The I/O registers that cause the microcontroller to operate when they are read are read-protected and therefore cannot be read ("?" is displayed in [Value]).
To read out the value of a read-protected I/O register, select [Force Read Value] from the context menu. Reading of each register is allowed only once. After [Force Read Value] is applied, the register is no longer marked "?" so it will not be instantly recognizable as read-protected.

2. When the selected microcontroller supports multi-core, this panel displays/changes the value regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

Remarks 1. This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

2. When the separator line of each area in this panel is double-clicked, the width of the area changes to the shortest possible size that can display the contents of the area.

Figure A-31. IOR Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (IOR panel-dedicated items)]
- [[Edit] menu (IOR panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [IOR].

[Description of each area]

(1) Search area

This area is used to search for the I/O register name.

	Specifies the character strings to search (case-insensitive). You can either type character strings directly from the key board (up to 512 characters), or select one from the input history via the drop-down list (up to 10 items).
	Searches up for the I/O register name containing the string specified in the text box, and selects the I/O register that is found.
	Searches down for the I/O register name containing the string specified in the text box, and selects the I/O register that is found.

Remarks 1. A hidden I/O register name being classified with a category can be searched (the category is opened and the I/O register is selected).

2. After typing character strings to search, to press the [Enter] key is the same function as clicking the  button, and to press the [Shift] + [Enter] key is the same function as clicking the  button.

(2) [IOR] area

The types of I/O register are classified as categories (folders), and a list of the respective I/O register name is displayed.

The meanings of the icons are as follows:

	Indicates that the I/O register name belonging to this category is displayed. When you double-click on the icon, or click on the "-" mark, the category is closed and the I/O register name is hidden. Note that no categories exist by default. Perform Tree editing if you need a category.
	Indicates that the I/O register name belonging to this category is hidden. When you double-click on the icon, or click on the "+" mark, the category is opened and the I/O register name is displayed. Note that no categories exist by default. Perform Tree editing if you need a category.
	Indicates the name of the I/O register.

Remark The category names are sorted in character code order by clicking on the header part of this area (the I/O register names in the category are also similarly sorted).

This area is provided with the following functions.

(a) Tree editing

The each I/O register can be categorized (by folders) and displayed in the tree view.

To create a category, Click the  button on the toolbar or select [Create Category] from the context menu after moving the caret to a I/O register name to create a category, and then input a desired name from the keyboard (up to 1024 characters).

To delete a category, select the category then click the  button on the toolbar or select [Delete] from the context menu. However, the categories that can be deleted are only the empty categories.

To rename the created category, select the category then do either one of the following.

- Click the name again, then directly rename the category name.
- Select the [Edit] menu >> [Rename], then directly rename the category name.
- Press the [F2] key, then directly rename the category name.

By directly dragging and dropping the I/O register in the created category, each I/O register is displayed in the categorized tree view.
Also, the display order of the categories and the I/O register names (upper or lower position) can be changed easily by drag and drop operation.

Cautions 1. Categories cannot be created within categories.
2. I/O registers cannot be added or deleted.

(b) Registering a watch-expression

Variable names of C language, CPU registers, I/O registers, and assembler symbols can be registered in the Watch panel as watch-expressions.
See "(1) Register a watch-expression" for details on how to operate it.

Remarks 1. When you have registered a watch-expression with a category as the object, all of I/O registers belonging to that category are registered as watch-expressions.
2. A scope specification is automatically added to a registered watch-expression.

(3) [Value] area

The value of I/O register is displayed and changed.
The radix of a data value can be selected by the button on the toolbar or the context menu item. In addition, a display format adding the value in hexadecimal number constantly can also be selected as well.
The meanings of the marks and colors displayed as I/O register values are as follows (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box):

Display Example (Default)			Description
0x0	Character color	Blue	The value of the I/O register that the user is changing (press the [Enter] key to write to the target memory).
	Background color	Standard color	
0x0	Character color	Brown	The value of the I/O register that has been changed because of the execution of a program To reset the highlighting, select the button on the toolbar or [Reset Color] from the context menu.
	Background color	Cream	
?	Character color	Gray	The value of the I/O register that is a read-protected object[Note]
	Background color	Standard color	

Note An I/O register for which the microcontroller ends up being activated by a read operation is shown.
To read the value of read-protected I/O register, select [Force Read Value] from the context menu.

Caution The timing for acquiring the values differs in the case of a 1 byte/2 bytes I/O register and that of 1 bit I/O registers that have been allocated to a 1 byte/2 bytes I/O register. Owing to this, there are also cases where the values differ even if the value of the same I/O register is displayed.

Remark The values are sorted in ascending order of the numerical values by clicking on the header part of this area.

This area is provided with the following functions.

(a) Changing I/O register values

To edit the I/O register value, select the value to edit, then change the value directly from the keyboard after clicking again on it (press the [Esc] key to cancel the edit mode).

After you edit the value of the I/O register, it is written to the register of the debug tool by pressing the [Enter] key, or moving the focus to outside the edit region.

See "(4) Modify the I/O register contents" for details on the method for changing the I/O register value.

(b) Saving the contents of the I/O register

The Save As dialog box can be opened by selecting the [File] menu >> [Save *IOR* Data As...], and all the contents of the I/O register can be saved in a text file (*.txt) or CSV file (*.csv).

See "(6) Save the I/O register contents" for details on the method for saving the contents of the I/O register.

(4) [Type (Byte Size)] area

The type information of each I/O register is displayed in the form shown below.

- <*Type of I/O register*> [<*Access attribute*> <*Accessible sizes*>](<*Size*>)

Access attribute	One of the following is displayed as the access attribute.	
	R	Read only
	W	Write only
	R/W	Read/Write
Accessible sizes	All accessible sizes are demarcated by a comma and listed in order of the smallest size in bit units (1 to 32 bits).	
Size	The size of the I/O register is displayed. It is displayed by supplying the unit, in byte units in the event that it can be displayed in byte units, and in bit units in the event that it can be displayed on in bit units.	

Examples 1. "The case of "IOR [R/W 1.8] (1 byte)"
An I/O register that is readable/writable and 1 bit accessible/8 bit accessible, and whose size is 1 byte

2. "The case of "IOR [R/W 1] (1 bit)"
An I/O register that is readable/writable and 1 bit accessible, and whose size is 1 byte

Remark The type information is sorted in the character code order by clicking on the header part of this area.

(5) [Address] area

The address that each I/O register is mapped is displayed (hexadecimal number notation fixing).

However, in the case of the bit register, it is displayed by providing a bit offset value like the following examples.

Examples 1. The case of "0xFF40"
This is allocated to the address "OxFF40"

2. The case of "0xFF40.4"
This is allocated to bit 4 of the address "0xFF40.4" (bit register)

Remark The addresses are sorted in ascending order of numerical values by clicking on the header part of this area.

[Toolbar]

	Acquires the latest data from the debug tool, and updates the contents of this panel. Note that the values of read-protected I/O register are not re-read. This item is disabled during execution of a program.
	Resets highlighting of the selected I/O register whose value has been changed by executing a program. Note that this item is disabled during execution of a program.
	Adds a new category (folder). Directly input the category name in the text box. There are no restrictions on the number of categories that can be created anew (however, it is not possible to create a category inside a category). Note that this item is disabled during execution of a program.
	Deletes the selected character string(s). If an empty category is in a select state, its category is deleted (it is not possible to delete I/O registers).
Notation	The following buttons to change the notation of a data value are displayed.
	Displays the value of the selected item in hexadecimal number (default).
	Displays the value of the selected item in signed decimal number.
	Displays the value of the selected item in unsigned decimal number.
	Displays the value of the selected item in octal number.
	Displays the value of the selected item in binary number.
	Displays the value of the selected item in ASCII code.
	Adds the value in hexadecimal number enclosing with "()" at the end of the value of the selected item.

[[File] menu (IOR panel-dedicated items)]

The following items are exclusive for the [File] menu in the IOR panel (other items are common to all the panels).
Note that all these items are disabled during execution of a program.

Save IOR Data	Overwrites the contents of this panel to the previously saved text file (*.txt)/CSV file (*.csv) (see "(b) Saving the contents of the I/O register"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save IOR Data As...].
Save IOR Data As...	Opens the Save As dialog box to newly save the contents of this panel to the specified text file (*.txt)/CSV file (*.csv) (see "(b) Saving the contents of the I/O register").

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

The following items are exclusive for [Edit] menu in the IOR panel (all other items are disabled).

Cut	Deletes the selected character string(s) and copies them to the clipboard (it is not possible to cut I/O registers/categories).
Copy	Copies the contents of the selected range to the clipboard as character string(s). If the I/O register(s)/category(s) are selected, copies them to the clipboard. The copied item can be pasted to the Watch panel.
Paste	If texts are in editing, pastes the contents of the clipboard to the caret position (it is not possible to paste I/O registers/categories).

Delete	Deletes the selected character string(s). If an empty category is in a select state, its category is deleted (it is not possible to delete I/O registers).
Select All	If texts are in editing, selects all the character strings. If texts are not in editing, selects all the I/O registers/categories.
Rename	Edits the name of the selected category.
Find...	Moves the focus to the text box in the Search area.
Move...	Opens the Go to the Location dialog box to move the caret to the specified I/O register.

[Context menu]

Register to Watch1	Registers the selected I/O register or category to the Watch panel (Watch1).
Refresh	Acquires the latest data from the debug tool, and updates the contents of this panel. Note that the values of read-protected I/O register are not re-read. This item is disabled during execution of a program.
Force Read Value	Forcibly reads once the value of the read-protected I/O register.
Move...	Opens the Go to the Location dialog box.
Create Category	Adds a new category (folder). Directly input the category name in the text box. There are no restrictions on the number of categories that can be created anew (however, it is not possible to create a category inside a category). Note that this item is disabled during execution of a program.
Copy	Copies the contents of the selected range to the clipboard as character string(s). If the I/O register(s)/category(s) are selected, copies them to the clipboard. The copied item can be pasted to the Watch panel.
Delete	Deletes the selected character string(s). If an empty category is in a select state, its category is deleted (it is not possible to delete I/O registers).
Notation	The following cascade menus are displayed to specify the notation.
Hexadecimal number	Displays the value of the selected item in hexadecimal number (default).
Signed Decimal	Displays the value of the selected item in signed decimal number.
Unsigned decimal number	Displays the value of the selected item in unsigned decimal number.
Octal	Displays the value of the selected item in octal number.
Binary	Displays the value of the selected item in binary number.
ASCII	Displays the value of the selected item in ASCII code.
Include Hexadecimal Value	Adds the value in hexadecimal number enclosing with "()" at the end of the value of the selected item.
Reset Color	Resets highlighting of the selected I/O register whose value has been changed by executing a program.

Local Variables panel

This panel is used to display the contents of the local variable and change the local variable values (see "2.10.5 Display/change local variables").

This panel appears only when connected to the debug tool.

Cautions 1. Nothing is displayed on this panel during execution of a program. When the execution of a program is stopped, items in each area are displayed.

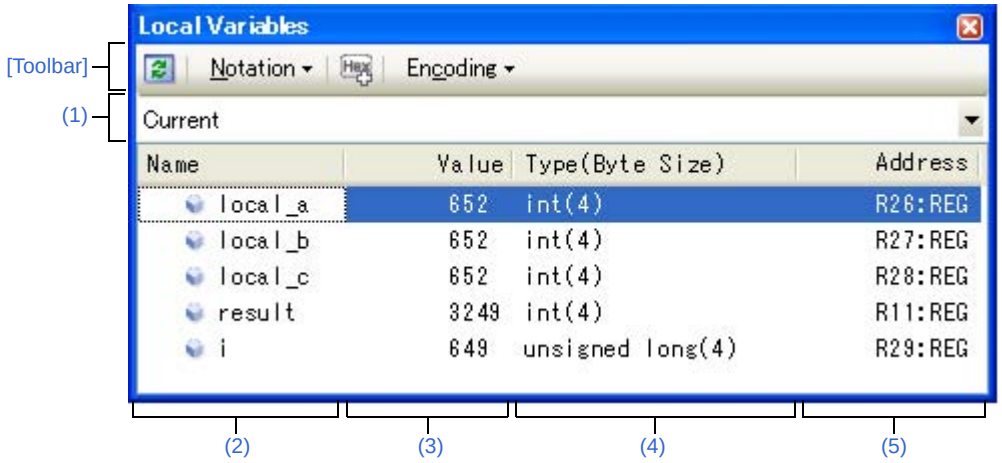
2. Due to compiler optimization, the data for the target variable may not be on the stack or in a register in blocks where that variable is not used. In this case, the target variable will not be displayed.

3. When the selected microcontroller supports multi-core, this panel displays/changes the value regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

Remarks 1. This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

2. When the separator line of each area in this panel is double-clicked, the width of the area changes to the shortest possible size that can display the contents of the area.

Figure A-32. Local Variables Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Local Variables panel-dedicated items)]
- [[Edit] menu (Local Variables panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [Local Variable].

[Description of each area]

(1) Scope area

Select the scope of the local variable to be displayed from the following drop-down list.

Item	Operation
Current	Displays local variables in the scope of the current PC value.
<*Depth*> <*Function name*() [*file name#line number*]>[Note]	Displays local variables in the scope of the calling function. After the program is executed, the scope that is selected is maintained as long as the selected scope exists.

Note The calling functions displayed in the Call Stack panel are displayed.

(2) [Name] area

The local variable name or function name is displayed.

The argument of the function is also displayed as the local variable.

In addition, the hierarchical structure is displayed in tree format for arrays, pointer variables, and structures or unions.

This area cannot be edited.

The meanings of the icons are as follows:

	Indicates the variable. Auto variables, internal static variables, and register variables are also displayed[Note]. In addition, the hierarchical structure is displayed in tree format for arrays, pointer variables, and structures or unions. If "+" mark exist at the top of the name, the next structure is expanded by clicking it (the mark changes to "-" after the expansion).	
	Array	All elements in the array
	Pointer variables	Variables that the pointer designates If the pointer designates a pointer, add "+" mark and expand it by clicking the mark. Note that if the pointer designates an unknown, "?" mark is displayed.
	Structures/Unions	All the member of structures/unions
	Indicates the argument.	
	Indicates the function.	

Note When Auto variables are used to display local variables, accurate values cannot be displayed at a prologue ("{") or epilogue ("}") of a function. The Auto variable addresses are the relative addresses from the address pointed to by the stack pointer (SP), so their addresses are not determined until the SP value is determined in the function. The SP is manipulated via prologues or epilogues, so the accurate value cannot be displayed.

This area is provided with the following functions.

(a) Registering a watch-expression

Variable names of C language can be registered in the Watch panel as watch-expressions.

See "(1) Register a watch-expression" for details on how to operate it.

Remark A scope specification is automatically added to a registered watch-expression.

(b) Jump to memory

By selecting [Jump to Memory] from the context menu, the Memory panel (Memory1) opens with moving the caret to the source line corresponding to the address where the selected local variable is disposed (if the Memory panel (Memory1) is already open, the screen will jump to the panel).

(3) [Value] area

The value of the local variable is displayed and changed.

The notation of a data value can be selected by the button on the toolbar or the context menu item. In addition, a display format adding the value in hexadecimal number constantly can also be selected as well.

The meanings of the marks and colors displayed as the values of the local variables are as follows (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box):

Display Example (Default)			Description
0x0	Character color	Blue	The value of the local variable that the user is changing Press the [Enter] key to write to the target memory.
	Background color	Standard color	
0x0	Character color	Brown	The value of the local variable that have been changed because of the execution of a program[Note]. The highlighting is rest by executing again the program.
	Background color	Cream	
?	Character color	Gray	The value of the local variable that could not be acquired.
	Background color	Standard color	

Note Variables that the name stays same from the start point where the program started executing to the breakpoint and their values are changed are the target.

This area is provided with the following functions.

(a) Changing the local variable/argument value

To edit the local variable value or the argument value, select the value to edit, then change the value directly from the keyboard after clicking again on it (press the [Esc] key to cancel the edit mode).

After you edit the value of the local variable or the argument, it is written to the target memory of the debug tool by pressing the [Enter] key or moving the focus to outside the edit region.

See "(2) Modify the contents of local variables" for details on the method for changing the local variable/ argument value.

(b) Saving the contents of the local variable

The Save As dialog box can be opened by selecting the [File] menu >> [Save Local Variables Data As...], and all the contents of this panel can be saved in a text file (*.txt) or CSV file (*.csv).

See "(3) Save the contents of local variables" for details on the method for saving the contents of the local variable.

(4) [Type (Byte Size)] area

The type name of the local variable is displayed. The notation accords with the description of C language.

For an array, an element number is displayed in "[]". For a function, its size (number of bytes) is displayed in "()".

This area cannot be edited.

(5) [Address] area

The address of the local variable is displayed. When a variable is assigned to the register, the name of the register is displayed.

This area cannot be edited.

[Toolbar]

	Acquires the latest data from the debug tool, and updates the contents of this panel.
Notation	The following buttons to specify the notation of values are displayed.
	Displays values on this panel in the default notation according to the type of variable (default).
	Displays values on this panel in hexadecimal number.
	Displays values on this panel in decimal number.
	Displays values on this panel in octal number.
	Displays values on this panel in binary number.
	Displays array indexes on this panel in decimal number (default).
	Displays array indexes on this panel in hexadecimal number.
	Displays values on this panel in float. Note that when the value is not 4-byte data, or has the type information, displays it in the default notation.
	Displays values on this panel in double. Note that when the value is not 4-byte data, or has the type information, displays it in the default notation.
	Adds the value in hexadecimal number enclosing with "()" at the end of the value.
Encoding	The following buttons to specify the encoding of character variables are displayed.
	Displays character variables in ASCII code (default).
	Displays character variables in Shift-JIS code.
	Displays character variables in EUC-JP code.
	Displays character variables in UTF-8 code.
	Displays character variables in UTF-16 code.

[[File] menu (Local Variables panel-dedicated items)]

The following items are exclusive for the [File] menu in the Local Variables panel (other items are common to all the panels).

Note that all these items are disabled during execution of a program.

Save Local Variables Data	Overwrites the contents of this panel to the previously saved text file (*.txt)/CSV file (*.csv) (see "(b) Saving the contents of the local variable"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Local Variables Data As...].
Save Local Variables Data As...	Opens the Save As dialog box to newly save the contents of this panel to the specified text file (*.txt)/CSV file (*.csv) (see "(b) Saving the contents of the local variable").

[[Edit] menu (Local Variables panel-dedicated items)]

The following items are exclusive for [Edit] menu in the Local Variables panel (all other items are disabled).

Copy	Copies the contents of the selected line or the character string to the clipboard.
Select All	Selects all the items of this panel.
Rename	Changes to the edit mode to edit the selected local variable value (see "(2) Modify the contents of local variables"). This item is disabled during execution of a program.
Find...	Opens the Find and Replace dialog box with selecting the [Find in Files] tab.
Replace...	Opens the Find and Replace dialog box with selecting the [Replace in Files] tab.

[Context menu]

Register to Watch1	Registers the selected local variable to the Watch panel (Watch1).
Copy	Copies the contents of the selected line or the character string to the clipboard.
Notation	The following cascade menus to specify the notation of values are displayed.
AutoSelect	Displays values on this panel in the default notation according to the type of variable (default).
Hexadecimal	Displays values on this panel in hexadecimal number.
Decimal	Displays values on this panel in decimal number.
Octal	Displays values on this panel in octal number.
Binary	Displays values on this panel in binary number.
Decimal Notation for Array Index	Displays array indexes on this panel in decimal number (default).
Hexadecimal Notation for Array Index	Displays array indexes on this panel in hexadecimal number.
Float	Displays values on this panel in float. Note that when the value is not 4-byte data, or has the type information, displays it in the default notation.
Double	Displays values on this panel in double. Note that when the value is not 4-byte data, or has the type information, displays it in the default notation.
Include Hexadecimal Value	Adds the value in hexadecimal number enclosing with "()" at the end of the value.
Encoding	The following cascade menus to specify the encoding of character variables are displayed.
ASCII	Displays character variables in ASCII code (default).
Shift_JIS	Displays character variables in Shift-JIS code.
EUC-JP	Displays character variables in EUC-JP code.
UTF-8	Displays character variables in UTF-8 code.
UTF-16	Displays character variables in UTF-16 code.
Jump to Memory	Opens the Memory panel (Memory1) and jumps to the memory value corresponding to the address of the selected line in this panel.

Watch panel

This panel is used to display the contents of the registered watch-expressions and change their values (see "2.10.6 Display/change watch-expressions").

Up to a maximum of four of these panels can be opened. Each panel is identified by the names "Watch1", "Watch2", "Watch3", and "Watch4" on the titlebar, and the watch-expressions can be registered/deleted/moved individually.

Watch-expressions can be registered in this panel as well as in the Editor panel, Disassemble panel, Memory panel, CPU Register panel, Local Variables panel or IOR panel.

When the panel is closed with registered watch-expressions, the panel closes but the information on the registered watch-expressions is retained. Therefore, if the same panel is opened again, it is opened with the watch-expressions registered.

The display contents are automatically updated when the value of the watch-expression changes after a program is executed (when the execution is done in steps, the display is updated after each step).

In addition, by enabling the Real-time display update function, it is also possible to update the display contents in real-time even while a program is being executed.

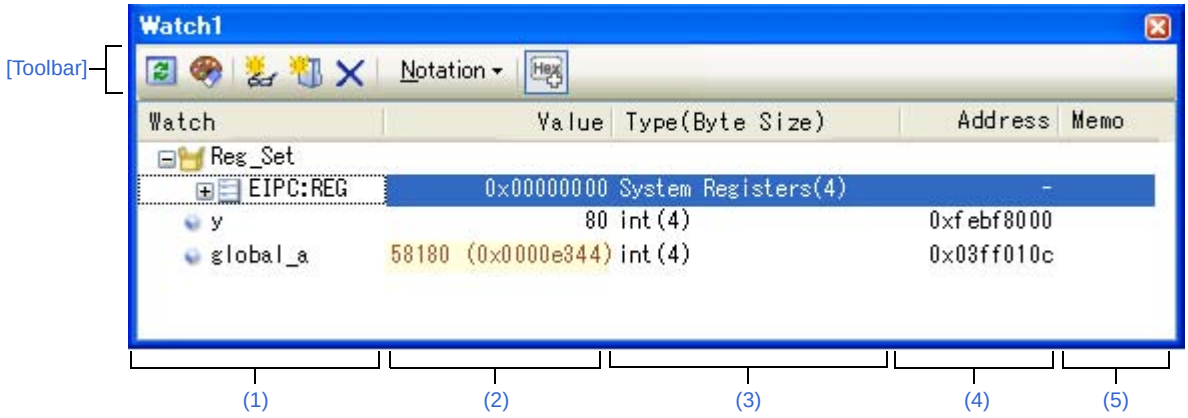
This panel appears only when connected to the debug tool.

Caution When the selected microcontroller supports multi-core, this panel displays/changes the value regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

Remarks 1. This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

2. When the separator line of each area in this panel is double-clicked, the width of the area changes to the shortest possible size that can display the contents of the area.

Figure A-33. Watch Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Watch panel-dedicated items)]
- [[Edit] menu (Watch panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [Watch] >> [Watch*1* - *4*].

[Description of each area]

(1) [Watch] area

All the registered watch-expressions are displayed in a list.
Clicking the title of the list in this area sorts the watch-expressions in the list in alphabetical order. Categories (folders) can be created to categorize the watch-expressions and display them in the tree view (see "(a) Tree editing").
The meanings of the icons are as follows:

	Indicates that the watch-expression belonging to this category is displayed. When you double-click on the icon, or click on the "-" mark, the category is closed and the watch-expression is hidden.
	Indicates that the watch-expression belonging to this category is hidden. When you double-click on the icon, or click on the "+" mark, the category is opened and the watch-expression is displayed.
	Indicates that the watch-expression is a variable. At the top of the watch-expression represents arrays, pointer type variables, and structures/unions, "+"/"-" mark is displayed. Click the mark to Expand/shrink display.
	Indicates that the watch-expression is a function.
	Indicates that the watch-expression is an immediate value.
	Indicates that the watch-expression is an expression.
	Indicates that the watch-expression is I/O register.
	Indicates that the watch-expression is CPU register. At the top of the watch-expression that has the lower level register (part of the register), "+"/"-" mark is displayed. Click the mark to Expand/shrink display.

This area is provided with the following functions.

(a) Tree editing

Watch-expressions can be categorized (by folders) and displayed in the tree view.
To create a category, click the  button on the toolbar or select [Create Category] from the context menu after moving the caret to the position to create a category, and then input a desired name from the keyboard.
To delete a category, select the category then click the  button on the toolbar or select [Delete] from the context menu.
To rename the created category, select the category then do either one of the following.

- Click the name again, then directly rename the category name.
- Select the [Edit] menu >> [Rename], then directly rename the category name.
- Press the [F2] key, then directly rename the category name.

By directly dragging and dropping the registered watch-expression in the created category, each category is displayed in the categorized tree view. Also, the display order of the categories and the watch-expressions (upper or lower position) can be changed easily by drag and drop operation.

Cautions 1. Categories cannot be created within categories.
2. Up to 1500 categories can be created in one watch panel (if this restriction is violated, a message appears).

Remark Drag and drop the watch-expressions/categories in other watch panel (Watch1 to Watch4) to copy them.

(b) Expand/shrink display

At the top of the watch-expression represents arrays, pointer type variables, structures/unions, and registers (with the name of the part), "+"/"-" mark is displayed. Click the mark to expand the contents ("+" mark is changed to "-" after the expansion).

Watch-Expression	Contents When Expanded
Array	All elements in the array Select [Encoding] >> [ASCII] from the context menu to display the value as a string (up to 256 characters). Note, however, that any characters that cannot be displayed in the encoding will be shown as periods "." or "?".
Pointer type variable	Variables that the pointer designates
Structure/Union	All the member of structure/union
Register	Name of the bit/bit string that constructs register Example) ECR register FECC register EICC register

(c) Registering new watch-expression

There are following three methods of registering a new watch expression.

<1> Register from other panels

Do either one of the following to register watch-expressions in other panels.

- Drag and drop the target character string onto this area in the desired watch panel (Watch1 to Watch4).
- Select [Register to Watch1] from the context menu after selecting the target character string or place the caret on either of the target character string (the target is automatically determined).
- Select the [Edit] menu >> [Paste] in this area in the desired watch panel (Watch1 to Watch4) after selecting the [Edit] menu >> [Copy] for the target character string.

The relationship between panels that can use this operation and targets that can be registered as watch-expressions is as follows:

Table A-2. Relationship between Panels and Targets That Can be Registered as Watch-Expressions

Panel Name	Targets That can be Registered as Watch-Expressions
Editor panel	Variable names of C language, CPU registers, I/O registers, and assembler symbols
Disassemble panel	Variable names of C language, CPU registers, I/O registers, and assembler symbols
CPU Register panel	CPU registers[Note]
Local Variables panel	Variable names of C language (local variables)
IOR panel	I/O registers[Note]

Note The scope-specification is automatically added to the registered watch-expression.

<2> Directly register in the Watch panel

Click the  button on the toolbar or select [Add New Watch] from the context menu in the desired watch panel (Watch1 to Watch4) to display an entry box for a new watch-expression in the bottom of this area.

Directly input a watch-expression from the keyboard in the [Watch] area in the entry box then press the [Enter] key.

For details on the input format of the watch-expression, see "(b) Watch-expression and operator".

Watch-expressions can be registered with specifying the scope. The scope specifications with watch-expression registration are as follows:

Cautions 1. If a load module name or file name contains a space or one of the following symbols, enclose the name in double-quotes (" ").
$, #, (,), [,], &, ^,~ , %, +, - *, /, :, ?, ', |, \, <, >, !
Example: "c:\folder\prog.abs" $file.c#func

2. If functions with the same name exist, write the type of parameter expressly.
Example: func(int, int)

Table A-3. Handling of a C Language Function When Registered in Watch by Specifying Scope

Scope Specification	Load Module File Name	Source File Name	Function Name	Subject to be searched
prog$file#func	prog	file	func	Static functions
prog$func	prog	Global	func	Global functions
file#func	Current	file	func	Static functions
func	Current	Current	func	All[Note]

Note A search is made for static functions and global functions from the scope of the current PC value in that order. Static functions out of scope are not searched for.

Table A-4. Handling of a C Language Variable When Registered in Watch by Specifying Scope

Scope Specification	Load Module File Name	Source File Name	Function Name	Variable Name	Subject to be searched
prog$file#func#var	prog	file	func	var	Static variables inside a static function[Note 1]
prog$file#var	prog	file	Global	var	Static variables inside a file
prog$var	prog	Global	Global	var	Global variables
file#func#var	Current	file	func	var	Static variables inside a static function[Note 1]
file#var	Current	file	Global	var	Static variables inside a file
var	Current	Current	Current	var	All[Note 2]

Notes 1. If the current PC value exists in a specified function, the local variables that are not declared as static also comprise the subject to be searched.

2. A search is made for local variables, static variables inside a file and global variables from the scope of the current PC value in that order. The local variables and the static variables inside a file that are out of scope are not searched for.

Table A-5. Handling of a CPU Register When Registered in Watch by Specifying Scope

Scope Specification	Register Bank	Name of CPU Register
r10:REG	(None)	r10

Table A-6. Handling of an I/O Register when Registered in Watch by Specifying Scope

Scope Specification	Name of I/O register
P0:IOR	P0
P0	P0

Remarks 1. A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this area (see "2.18.2 Symbol name completion function").

2. An immediate value is treated as an address. Note, however, that an immediate value with operators cannot be used.

3. An arithmetic expression with symbols cannot be used for a watch-expression.

4. If the same name exists either in C language variables, CPU registers or I/O registers, and it is registered without specifying scopes, then its value will be displayed after the symbol is determined in the following order.

Variable of C language > CPU registers > I/O register

If "$" is specified at the top of a watch-expression, then its value will be displayed after the symbol is determined in the following order.

CPU registers > I/O register > Variable of C language

5. If a local variable and a global variable exist with the same name, and its symbol name is registered without specifying scopes, then its value will be displayed after the symbol is determined based on the scope of the current PC value.

6. If the letter "I" alone is specified as a watch-expression, it is interpreted as an imaginary keyword. To acquire the value of a register "I," add ":REG" after the register.

7. When watch-expressions are registered from the IOR panel or the CPU Register panel, the scope specification is automatically added.

<3> Register from other application

Select a character string of a variable of C language, CPU register, I/O register or assembler symbol from a external editor then do either one of the following.

- Drag and drop the target character string in this area in the desired watch panel (Watch1 to Watch4).
- Select the [Edit] menu >> [Paste] in this area in the desired watch panel (Watch1 to Watch4) after copying the target character string.

Cautions 1. Up to 3000 watch-expressions can be registered in one watch panel (if this restriction is violated, a message appears).

2. Due to compiler optimization, the data for the target variable may not be on the stack or in a register in blocks where that variable is not used. In this case, the target watch-expression value is displayed as "?".

Remarks 1. Each watch-expression registered in each watch panel (Watch1 to Watch4) is managed in each panel and saved as the user information of the project.

2. More than one watch-expression with the same name can be registered.

3. You can export registered watch-expressions to a file and import it so that the watch-expressions can be re-registered (see "(8) Export/import watch-expressions").

(d) Editing watch-expression

To edit the registered watch-expression, double-click the watch-expression to be edited to change the watch-expression to edit mode then directly edit from the keyboard (press the [Esc] key to cancel the edit mode). After editing the watch-expression, press the [Enter] key to complete the editing.

(e) Deleting watch-expression

To delete the registered watch-expression, select the watch-expression(s) to be deleted then click the  button on the toolbar or select [Delete] from the context menu.

(f) Setting of various events

Various events can be set to the selected watch-expression by selecting [Access Break] or [Trace Output] from the context menu.

If an access event is set, the mark of the watch-expression is changed (the event mark of a break event is displayed under the icon of the watch-expression in layers).

When an event is set, the detailed information about the set event is reflected in the Events panel.

Note that events are only set to the watch-expressions that are global variables, static variables inside functions, or file-internal static variables.

See the following for details on how to set events.

- "2.9.5 Stop the program with the access to variables/I/O registers"
- "2.12.4 Collect execution history only when the condition is met [Simulator]"

(g) Jump to the address with memory definition

By selecting [Jump to Memory] from the context menu, the Memory panel (Memory1) opens with moving the caret to the address in which the selected watch-expression is defined (if the Memory panel (Memory1) is already open, the screen will jump to the panel).

Note that this operation is disabled when more than one watch-expression is selected at the same time or the CPU register/I/O register is selected.

(2) [Value] area

The value of the registered watch-expression is displayed and changed (if the watch-expression is a function pointer, the function name is displayed in this area).

Notations and encodes can be selected by the button on the toolbar or the context menu item. In addition, a display format adding the value in hexadecimal number constantly can also be selected as well.

The default display format of the values is automatically decided depending on the type of the watch-expression.

Table A-7. Display Format of Watch-Expressions (Default)

Type of Watch-Expression	Display Format
char, signed char, unsigned char	ASCII code with hexadecimal number
short, signed short, short int, signed short int, int, signed, signed int, long, signed long, long int, signed long int	Signed decimal number with hexadecimal number
unsigned short, unsigned short int, unsigned, unsigned int, unsigned long, unsigned long int	Unsigned decimal number with hexadecimal number
float	Float (when the size is 4-byte) with hexadecimal number
double, long double	Double (when the size is 8-byte) with hexadecimal number

Type of Watch-Expression	Display Format
Pointers to char, signed char, unsigned char	Characters Encoding: ASCII
Pointers to other than char, signed char, unsigned char	Hexadecimal number
Arrays of char, signed char, unsigned char types	Characters Encoding: ASCII
bit, boolean, _boolean	Unsigned decimal number with hexadecimal number
Enumeration type	Enumeration constant value with hexadecimal number
Label, address of immediate value, EQU symbol	Signed decimal number with hexadecimal number
bit symbol	Unsigned decimal number with hexadecimal number
Others	Hexadecimal number

The meanings of the marks and colors displayed as the values of watch-expressions are as follows (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box):

Display Example (Default)			Description
0x0	Character color	Blue	The value of the watch-expression that the user is changing Press the [Enter] key to write to the target memory.
	Background color	Standard color	
0x0	Character color	Pink	The value of the watch-expression that is displayed with the Real-time display update function
	Background color	Standard color	
0x0	Character color	Brown	The value of the watch-expression that has been changed because of the execution of a program To reset the highlighting, select the button on the toolbar or [Reset Color] from the context menu.
	Background color	Cream	
?	Character color	Gray	Variable that does not exist is registered as a watch-expression or the value of the watch-expression cannot be retrieved (i.e. when the I/O register is read-protected[Note], or a variable is out of the scope, etc.)
	Background color	Standard color	

Note The I/O register that cause the microcontroller to operate when it is read is read-protected and therefore cannot be read ("?" is displayed in the value).
To read out the value of a read-protected I/O register, select [Force Read Value] from the context menu.
Reading of each register is allowed only once.

Remarks 1. Each watch-expression acquires the value in the order it was registered.
As the timing to acquire a value is different, the values displayed may be different if the same I/O register is registered more than once.

2. When a hexadecimal value is also given, then values in the specified notation and hexadecimal values are read separately. For this reason, the values with the specified notion and the hexadecimal values may differ due to the time lag between being read.

This area is provided with the following functions.

(a) Real-time display update function

Using the real-time display update function allows you to display/modify the value of the watch-expression not only while the program is stopped, but also in execution.

See "(4) Display/modify the memory contents during program execution" for details on the real-time display update function.

(b) Changing values of watch-expressions

To edit the value of the watch-expression, change the value directly from the keyboard after double-clicking on the value to be edited (press the [Esc] key to cancel the edit mode).

After you edit the value of the watch-expression, it is written to the target memory of the debug tool by pressing the [Enter] key, or moving the focus to outside the edit region.

See "(6) Modify the contents of watch-expressions" for detail on how to change values of watch-expressions.

(c) Saving the contents of watch-expressions

By selecting the [File] menu >> [Save Watch Data As...], the Save As dialog box can be opened, and all the contents of this panel can be saved in a text file (*.txt) or CSV file (*.csv).

See "(9) Save the contents of watch-expressions" for details on the method for saving the contents of watch-expressions.

(3) [Type (Byte Size)] area

The type information of watch-expressions with the following format is displayed.

Watch-Expression	Display Format	
Single CPU register	<*Types of CPU register*> (<*Size*Note 1>)	
Single I/O register	<*I/O register type*> (<*Access attribute*> <*Accessible sizes*><*Size*Note 1>)	
	Access attribute	R: Read only W: Write only R/W: Read/Write only
	Accessible sizes	All accessible sizes are demarcated by a comma and listed in order of the smallest size in bit units (1 to 32 bits).
Unknown	?	
Others	<*Watch-expression type that follow the C compiler's determination*Note 2> (<*Size*Note 1>)	

Notes 1. The size of the watch-expression is displayed in bytes.
However, for bit I/O register or C language bit field, the size is displayed in bits and "bits" is added to the end of the number.

2. Types to be treated are displayed when compiling the watch-expression.

(4) [Address] area

The address that each watch-expression is mapped is displayed (hexadecimal number notation fixing).
If the watch-expression is single CPU register or is unknown, "-" or "?" is displayed instead.

Remark If the watch-expression is the bit I/O register, the bit-offset value is also displayed as follows:
Example When the bit register is allocated to bit 4 of the address "0xFF40":
Display example: 0xFF40.4

(5) [Memo] area

The user can write comments for the watch-expressions/categories.

Each comment for a watch-expression/category written in this area is saved individually as the user information of the project. Therefore, when any of the watch-expression/category is deleted, the comment corresponding to it is also deleted.

Note that when arrays or register are displayed expanded, the comment cannot be input for each element.
To edit the comment, input the character strings directly from the keyboard after double-clicking on the item to be edited (press the [Esc] key to cancel the edit mode). Up to 256 character strings can be input (line feed code is ignored).
After editing the character strings, complete the editing by pressing the [Enter] key or moving the focus to outside the edit region.

[Toolbar]

	Reacquires all the values of the registered watch-expression and updates the display. Note that read-protected I/O register values are not re-read.
	Resets highlighting of the selected watch-expression whose value has been changed by executing a program. This item is disabled during execution of a program.
	Registers a new watch-expression. Directly input the watch-expression in the text box (see "(c) Registering new watch-expression") Note that up to 3000 watch-expressions can be registered in one watch panel.
	Adds a new category (folder). Directly input the category name in the text box. Note that up to 1500 categories can be created in one watch panel (categories cannot be created in categories).
	Deletes the selected character string(s). If the watch-expression(s)/category(s) are selected, deletes them (except when the expanded item of the watch-expression is selected).
Notation	The following buttons to change the notation of a data value are displayed.
	Displays the value of the selected watch-expression in the default notation (see "Table A-7. Display Format of Watch-Expressions (Default)") according to the type of variable (default).
	Displays the value of the selected item in hexadecimal number.
	Displays the value of the selected item in signed decimal number.
	Displays the value of the selected item in unsigned decimal number.
	Displays the value of the selected item in octal number.
	Displays the value of the selected item in binary number.
	Displays the value of the selected item in ASCII code.
	Displays the value of the selected item in float. Note that this item becomes valid only when the selected watch-expression value is 4-byte data.
	Displays the value of the selected item in double. Note that this item becomes valid only when the selected watch-expression value is 8-byte data.
	Adds the value in hexadecimal number enclosing with "()" at the end of the value of the selected item (except the item displayed in hexadecimal number).

[[File] menu (Watch panel-dedicated items)]

The following items are exclusive for the [File] menu in the Watch panel (other items are common to all the panels).
Note that all these items are disabled during execution of a program.

Save Watch Data	Overwrites the contents of this panel to the previously saved text file (*.txt)/CSV file (*.csv) (see "(c) Saving the contents of watch-expressions"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Watch Data As...].
Save Watch Data As...	Opens the Save As dialog box to newly save the contents of this panel to the specified text file (*.txt)/CSV file (*.csv) (see "(c) Saving the contents of watch-expressions").

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

The following items are exclusive for [Edit] menu in the Watch panel (all other items are disabled).

Cut	Deletes the selected character string(s) and copies them to the clipboard. If the watch-expression(s)/category(s) are selected, deletes them (except when the expanded item of the watch-expression is selected).
Copy	Copies the contents of the selected range to the clipboard as character string(s). If the watch-expression(s)/category(s) are selected, copies them to the clipboard (except when the expanded item of the watch-expression is selected).
Paste	If texts are in editing, pastes the contents of the clipboard to the caret position. If texts are not in editing and the watch-expression(s) are copied in the clipboard, registers them to the caret position.
Delete	Deletes the selected character string(s). If the watch-expression(s)/category(s) are selected, deletes them (except when the expanded item of the watch-expression is selected).
Select All	If texts are in editing, selects all the character strings. If texts are not in editing, selects all the watch-expressions/categories.
Rename	Renames the selected watch-expression/category.
Find...	Opens the Find and Replace dialog box with selecting the [Find in Files] tab.
Replace...	Opens the Find and Replace dialog box with selecting the [Replace in Files] tab.

[Context menu]

Access Break	This item becomes valid only when the selected watch-expression is the global variable, the static variable inside functions, the file-internal static variable, or I/O register (multiple selections not allowed). The following cascade menus are displayed to set the access break event (see "(1) Set a break event (access type)").
Set Read Break to	Sets a break event with read access condition to the selected watch-expression.
Set Write Break to	Sets a break event with write access condition to the selected watch-expression.
Set R/W Break to	Sets a break event with read/write access condition to the selected watch-expression.
Trace Output	This item becomes valid only when the selected watch-expression is a global variable, static variable inside functions, file-internal static variable, or I/O register (multiple selections not allowed). The following cascade menus are displayed to set the trace-related event (see "(1) Set a Point Trace event").

Record Reading Value	Sets a Point Trace event to record the values in the trace memory when the selected watch-expression is accessed for read.
Record Writing Value	Sets a Point Trace event to record the values in the trace memory when the selected watch-expression is accessed for write.
Record R/W Value	Sets a Point Trace event to record the values in the trace memory when the selected watch-expression is accessed for read/write.
Trace	Opens the Trace panel and displays the acquired trace data.
Periodic Updating	The following cascade menus are displayed to set for the real-time display update function (see "(a) Real-time display update function").
Periodic Updating Options	Opens the Property panel to set for the real-time display update function.
Refresh	Reacquires all the values of the registered watch-expression and updates the display. Note that the values of read-protected I/O register are not re-read.
Force Read Value	Forcibly reads once the values of the read-protected I/O register. This item is disabled during execution of a program.
Add New Watch	Registers a new watch-expression. Directly input the watch-expression in the text box (see "(c) Registering new watch-expression") Note that up to 3000 watch-expressions can be registered in one watch panel.
Create Category	Adds a new category (folder). Directly input the category name in the text box. Note that up to 1500 categories can be created in one watch panel (categories cannot be created in categories).
Delete	Deletes the selected character string(s). If the watch-expression(s)/category(s) are selected, deletes them (except when the expanded item of the watch-expression is selected).
Cut	Deletes the selected character string(s) and copies them to the clipboard. If the watch-expression(s)/category(s) are selected, deletes them (except when the expanded item of the watch-expression is selected).
Copy	Copies the contents of the selected range to the clipboard as character string(s). If the watch-expression(s)/category(s) are selected, copies them to the clipboard (except when the expanded item of the watch-expression is selected).
Paste	If texts are in editing, pastes the contents of the clipboard to the caret position. If texts are not in editing and the watch-expression(s) are copied in the clipboard, registers them to the caret position.
Rename	Renames the selected watch-expression/category.
Import Watch Expression...	Opens the Open Watch Expression Data File dialog box to import watch-expressions (see "(8) Export/import watch-expressions").
Notation	The following cascade menus are displayed to specify the notation.

AutoSelect	Displays the value of the selected watch-expression in the default notation (see "Table A-7. Display Format of Watch-Expressions (Default)") according to the type of variable (default).
Hexadecimal number	Displays the value of the selected item in hexadecimal number.
Signed Decimal	Displays the value of the selected item in signed decimal number.
Unsigned decimal number	Displays the value of the selected item in unsigned decimal number.
Octal	Displays the value of the selected item in octal number.
Binary	Displays the value of the selected item in binary number.
ASCII	Displays the value of the selected item in ASCII code.
Include Hexadecimal Value	Adds the value in hexadecimal number enclosing with "()" at the end of the value of the selected item (except the item displayed in hexadecimal number).
Float	Displays the value of the selected item in float. Note that when the selected watch-expression value is not 4-byte data, or has the type information, displays it in the default notation (see "Table A-7. Display Format of Watch-Expressions (Default)").
Double	Displays the value of the selected item in double. Note that when the selected watch-expression value is not 8-byte data, or has the type information, displays it in the default notation (see "Table A-7. Display Format of Watch-Expressions (Default)").
Decimal Notation for Array Index	Displays array indexes on this panel in decimal number (default).
Hexadecimal Notation for Array Index	Displays array indexes on this panel in hexadecimal number.
Encoding	The following cascade menus are displayed to specify the character code.
ASCII	Displays the value of the selected item in ASCII code (default).
Shift_JIS	Displays the value of the selected item in Shift-JIS code.
EUC-JP	Displays the value of the selected item in EUC-JP code.
UTF-8	Displays the value of the selected item in UTF-8 code.
UTF-16	Displays the value of the selected item in UTF-16 code.
Size Notation	The following cascade menus are displayed to specify the size notation.
1 Bytes	Displays the value of the selected item as 8-bit data.
2 Bytes	Displays the value of the selected item as 16-bit data.
4 Bytes	Displays the value of the selected item as 32-bit data.
8 Bytes	Displays the value of the selected item as 64-bit data.
Jump to Memory	Opens the Memory panel (Memory1) and jumps to the address which the selected watch-expression is defined (see "(g) Jump to the address with memory definition").
Reset Color	Resets highlighting of the selected watch-expression whose value has been changed by executing a program. This item is disabled during execution of a program.

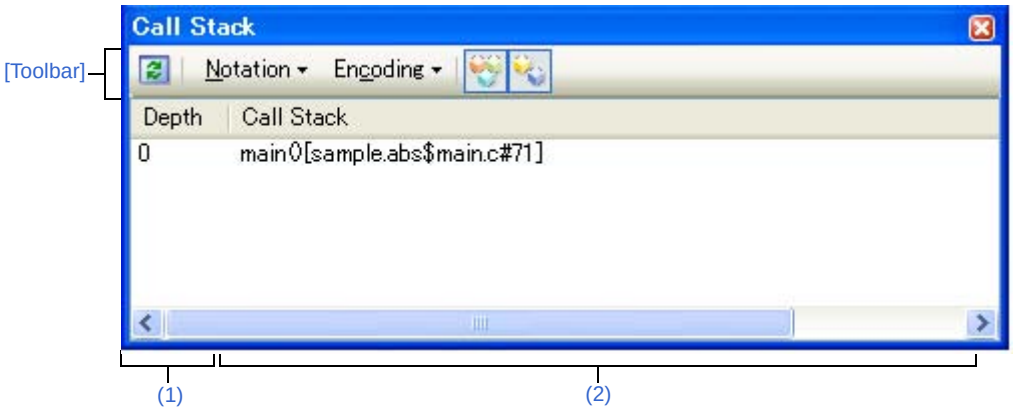
Call Stack panel

This panel is used to display the call stack information for the function call (see "2.11.1 Display call stack information"). This panel appears only when connected to the debug tool.

Cautions 1. Nothing is displayed on this panel during execution of a program.
When the execution of a program is stopped, items in each area are displayed.
2. When the selected microcontroller supports multi-core, this panel displays the value regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

Remark This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

Figure A-34. Call Stack Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Call Stack panel-dedicated items)]
- [[Edit] menu (Call Stack panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [Call Stack].

[Description of each area]

(1) [Depth] area

The depth of the call is displayed.
The line at the current PC position becomes 0 and incremented numbers from 1 is added to the calling function in the order.

(2) [Call Stack] area

The current source position and the call stack information pushed on the stack (position of the calling function and arguments of a each function, etc.) are displayed.

The display format in this area differs depending on the selection condition of the / button on the toolbar, or of [Show Parameter]/[Show Module File Name] from the context menu.

Condition	Display Format
- Display arguments - Display module file name	*<Function>*(*<Argument>*=*<Argument Value*[Note]*>*, ...)[*<Module file name>*$*<File name>*#*<Line number>*] (default)
- Display arguments - Do not display module file name	*<Function>*(*<Argument>*=*<Argument value*[Note]*>*, ...)[*<File Name>*#*<Line number>*]
- Do not display arguments - Display module file name	*<Function>*()[*<Module file name>*$*<File name>*#*<Line number>*]
- Do not display arguments - Do not display module file name	*<Function>*()[*<File name>*#*<Line number>*]

Note When the argument value is character string, up to 20 characters can be displayed.

Remark Array arguments are passed as pointers rather than arrays (C language specification). For this reason, if the argument is an array, it is displayed as a pointer.

This area is provided with the following functions.

(a) Jump to source line and disassemble

By selecting [Jump to Source] from the context menu, the Editor panel is opened with moving the caret to the source line corresponding to the calling function at the current caret position (if the Editor panel is already open, the screen will jump to the panel).
In addition, similarly by selecting [Jump to Disassemble], the Disassemble panel (Disasemmble1) is opened with moving the caret to the address corresponding to the calling function at the current caret position (if the Disassemble panel is already open, the screen will jump to the panel (Disassemble1)).

Remark It is possible to jump to the target source line by double-clicking on that line as well.

(b) Saving the contents of call stack information

By selecting the [File] menu >> [Save Call Stack Data As...], the Save As dialog box can be opened, and all the contents of this panel can be saved in a text file (*.txt) or CSV file (*.csv).
See "(4) Save the contents of call stack information" for details on the method for saving the contents of call stack information.

[Toolbar]

	Acquires the latest data from the debug tool, and updates the contents of this panel.
Notation	The following buttons to specify the notation of values are displayed.
auto	Displays values on this panel in the default notation according to the type of variable (default).
Hex	Displays values on this panel in hexadecimal number.
Dec	Displays values on this panel in decimal number.
Oct	Displays values on this panel in octal number.
Bin	Displays values on this panel in binary number.
Encoding	The following buttons to specify the encoding of character variables are displayed.

	Displays character variables in ASCII code (default).
	Displays character variables in Shift-JIS code.
	Displays character variables in EUC-JP code.
	Displays character variables in UTF-8 code.
	Displays character variables in UTF-16 code.
	Displays the call stack information with the module file name (default).
	Displays the call stack information with the parameters (arguments) of the function call (default).

[[File] menu (Call Stack panel-dedicated items)]

The following items are exclusive for the [File] menu in the Call Stack panel (other items are common to all the panels). Note that all these items are disabled during execution of a program.

Save Call Stack Data	Overwrites the contents of this panel to the previously saved text file (*.txt)/CSV file (*.csv) (see "(b) Saving the contents of call stack information"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Call Stack Data As...].
Save Call Stack Data As...	Opens the Save As dialog box to newly save the contents of this panel to the specified text file (*.txt)/CSV file (*.csv) (see "(b) Saving the contents of call stack information").

[[Edit] menu (Call Stack panel-dedicated items)]

The following items are exclusive for [Edit] menu in the Call Stack panel (all other items are disabled).

Copy	Copies the contents of the selected line to the clipboard.
Select All	Selects all the items of this panel.
Find...	Opens the Find and Replace dialog box with selecting the [Find in Files] tab.
Replace...	Opens the Find and Replace dialog box with selecting the [Replace in Files] tab.

[Context menu]

Copy	Copies the contents of the selected line to the clipboard.
Show Module File Name	Displays the call stack information with the module file name (default).
Show Parameter	Displays the call stack information with the parameters (arguments) of the function call (default).
Notation	The following cascade menus to specify the notation of values are displayed.
AutoSelect	Displays values on this panel in the default notation according to the type of variable (default).
Hexadecimal	Displays values on this panel in hexadecimal number.
Decimal	Displays values on this panel in decimal number.
Octal	Displays values on this panel in octal number.
Binary	Displays values on this panel in binary number.
Encoding	The following cascade menus to specify the encoding of character variables are displayed.

ASCII	Displays character variables in ASCII code (default).
Shift_JIS	Displays character variables in Shift-JIS code.
EUC-JP	Displays character variables in EUC-JP code.
UTF-8	Displays character variables in UTF-8 code.
UTF-16	Displays character variables in UTF-16 code.
Jump to Disassemble	Opens the Disassemble panel (Disassemble1) and jumps to the address corresponding to the calling function of the selected line in this panel.
Jump to Source	Opens the Editor panel and jumps to the source line corresponding to the calling function of the selected line in this panel.
Jump to Local Variable at This Time	Opens the Local Variables panel to display the local variable corresponding to the selected line.

Trace panel

This panel is used to display trace data recording the execution history of the program (see "2.12 Collect Execution History of Programs").

The trace data displays by mixing the disassembled text and source text by default, but it is also possible to display either one of these by selecting the Display mode.

After the execution of the program is stopped, the display position is automatically updated such that the latest trace data is displayed.

This panel appears only when connected to the debug tool.

Caution [Full-spec emulator][E1][E20]

When trace data has been collected after selecting [all core] in the [Trace target setting] property in the [trace] category on the [Debug Tool Settings] tab of the Property panel, this panel displays the trace data regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

Remarks 1. When the separator line of each area in this panel is double-clicked, the width of the area changes to the shortest possible size that can display the contents of the area.

2. This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

Figure A-35. Trace Panel [Full-spec emulator][E1][E20]

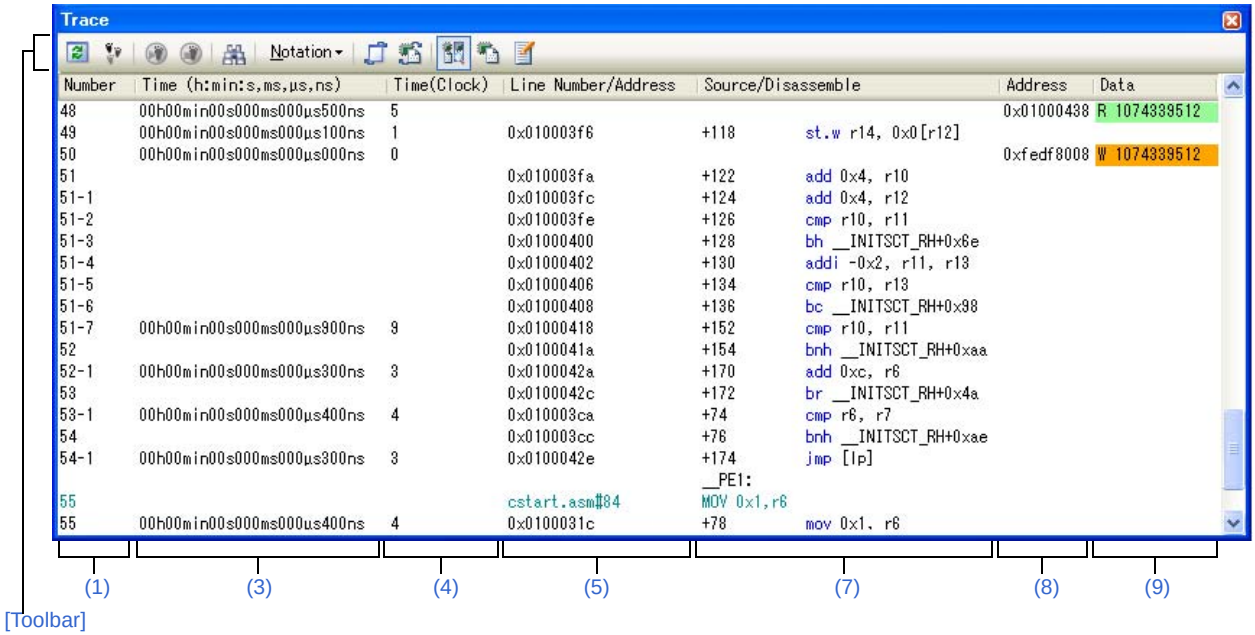
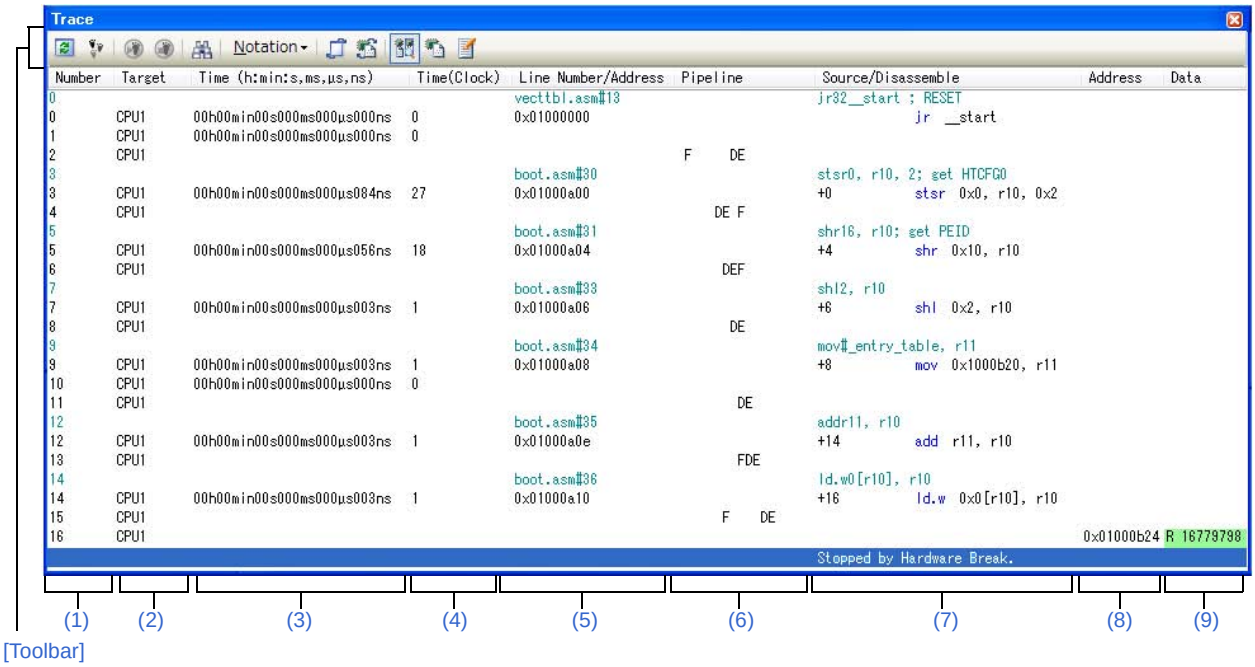


Figure A-36. Trace Panel [Simulator]



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Trace panel-dedicated items)]
- [[Edit] menu (Trace panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [Trace].
- On the Editor panel/Disassemble panel, select [Trace Settings] >> [Show Trace Result] from the context menu.

[Description of each area]

(1) [Number] area

The trace number corresponding to the trace frame is displayed.

(2) [Target] area [Simulator]

The name of the target core is displayed.

(3) [Time (h:min:s,ms,µs,ns)] area

This area displays the time required from the execution start of the program to the execution start of an instruction of each frame or generation of memory access cause.

The time is displayed in units of "hours, minutes, seconds, milliseconds, microseconds and nanoseconds".

Remarks 1. [Full-spec emulator][E1][E20]

The time is displayed as a relative time.

2. [Simulator]

The question of whether to set the time display as an accumulated time or differential time depends on the setting of the [Accumulate trace time] property on the [Trace] category on the [Debug Tool Settings] tab of the Property panel.

(4) [Time(Clock)] area

This area displays the time required from the execution start of the program to the execution start of an instruction of each frame or generation of memory access cause. The time is displayed in CPU clock cycles.

Remarks 1. [Full-spec emulator][E1][E20]

The time is displayed as a differential CPU clock cycles.

2. [Simulator]

The question of whether to set the time display as an accumulated CPU clock cycles or differential CPU clock cycles depends on the setting of the [Accumulate trace time] property on the [Trace] category on the [Debug Tool Settings] tab of the Property panel.

(5) [Line Number/Address] area

The line number of a source file or the address of the assemble code is displayed.
The notation of a data value can be selected by the button on the toolbar or the context menu item.
The display formats are as follows:

Type of Display Line	Display Format
Source text	*<File name>#<Line number>*
Instruction (disassemble results)	*<Address>*
Other than above	-

Remark Since the following execution histories are not displayed, the line numbers displayed are not consecutive numbers.

- CPU register access
- Operand access
- Invalid fetch

(6) [Pipeline] area [Simulator]

This area displays the pipeline execution status.
A 20-character string is displayed in this field, and each character represents the stage of the pipeline in one clock cycle. Residues of 20 from the number of clock cycles are used as indices in display of the string representing the corresponding stages of execution.
The meanings of the letters used to represent the stages are as follows.

Stage	Character
Fetch	F
Decode	D
Execute	E

Examples 1. F: 10th clock cycle, D: 11th clock cycle, E: 13th clock cycle
Display: FD_E_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

2. F: 18th clock cycle, D: 19th clock cycle, E: 20th clock cycle
Display: E_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _FD

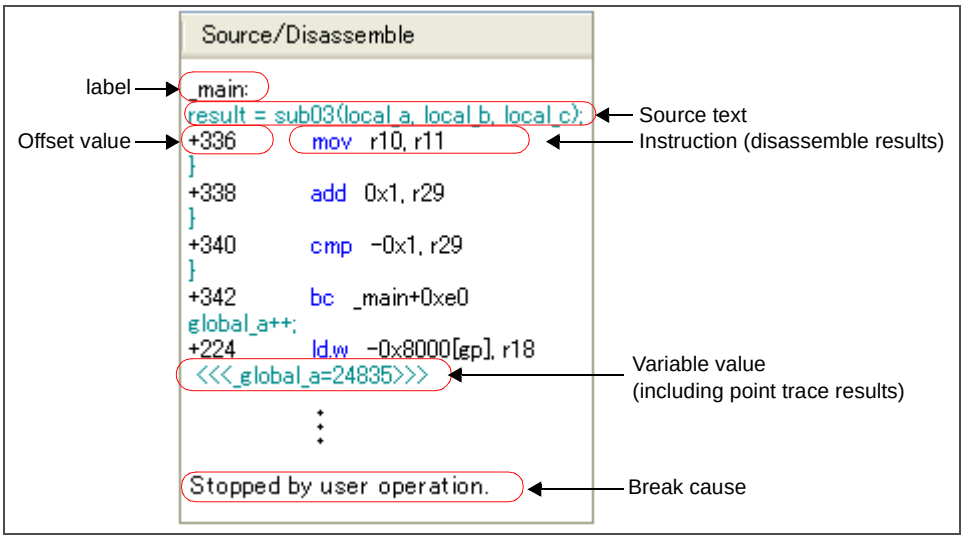
Remark "_" indicates a space.

(7) [Source/Disassemble] area

The collected trace data is displayed as follows:

Note that the items displayed in this area differ depending on the selection of the display mode (see "(a) Display mode").

Figure A-37. Display Contents of [Source/Disassemble] Area (Default)



Label	The label is displayed when a label is defined for the address.
Offset value	The offset value from the nearest label is displayed when a label is defined for the address.
Source text	The corresponding source text is displayed when the Mixed display mode or Source display mode is selected. However, when a place where no debugging information is present is executed, "<No Debug Information>" is displayed. In addition, when the value of a variable[Note 1] or an I/O register that is accessed during execution of a source line can be analyzed, that value is displayed in the following format at the end of the source line. - <<<*Variable name = Variable value*>>> - <<<*I/O register name = I/O register value*>>> Example: a=b; <<<a=5>>> The results of the Point Trace are displayed as same as format above.
Instruction (disassemble results)	The corresponding instructions are displayed as the result of disassembling when the Mixed display mode or Disassemble display mode is selected[Note 2]. The mnemonics are shown highlighted.
Break cause **[Simulator]**	The reason why the program has broken down is displayed.

Notes 1. When there is a memory access, a symbol will be interpreted as a variable and displayed only if a symbol is assigned to the accessed address. Note, however, that only variables of up to 4 bytes are supported. If multiplication or other code is processed by the standard libraries, the label of the SADDR area used by the standard library may be shown.

2. At a frame for which not all the trace data was fetched, "(LOST)" is displayed. In this case, the corresponding line is shown in error color (the error color depends on the configuration in the [General - Font and Color] category of the Option dialog box).

This area is provided with the following functions.

(a) Display mode

It is possible to select the following three display modes by selection of a button on the toolbar or the context menu.

Display Mode	Displayed Content
Mixed display mode	Displays the instruction (disassemble results), labels, source text (corresponding source line), point trace results, and break causes (default).
Disassemble display mode	Displays the instruction (disassemble results), labels, point trace results, and break causes.
Source display mode	Displays the source text (corresponding source line) and break causes. However, when a place where no debugging information is present is executed, "<No Debug Information>" is displayed.

(b) Jumping to source line or disassemble

By selecting [Jump to Source] from the context menu, the Editor panel opens with moving the caret to the source line corresponding to the line at the current caret position (if the Editor panel is already open, the screen will jump to the panel).

In addition, similarly by selecting [Jump to Disassemble], the Disassemble panel (Disasemmble1) is opened with moving the caret to the address corresponding to the fetch address of the line at the current caret position (if the Disassemble panel is already open, the screen will jump to the panel (Disassemble1)).

(c) Linking with other panels

By clicking the  button on the toolbar, or selecting [Window Connecting] >> [Connect Source Window]/[Connect Disassemble Window] from the context menu, it is possible to link and display the corresponding places on the Editor panel/Disassemble panel, with the address of the caret position on this panel used as the pointer (no movement of the focus is done).

(d) Pop-up display

By hovering the mouse cursor over a line, all the area (item) data corresponding to that line is pop-up displayed in tandem shape.

(e) Saving trace data

The Data Save dialog box can be opened by selecting the [File] menu >> [Save Trace Data As...], and the contents of this panel can be saved in a text file (*.txt) or CSV file (*.csv).

See "2.12.9 Save the contents of execution history" for details on the method for saving trace data.

(8) [Address] area

The target address of memory access is displayed.
However, in the event of access to I/O register, the I/O register name is displayed instead of the address (when a plurality is accessed these are displayed in the following lines).
The radix of a data value can be selected by the button on the toolbar or the context menu item.

(9) [Data] area

The accessed data value and the access type at that time are displayed.
However, CPU register access is not displayed.
The notation of a data value can be selected by the button on the toolbar or the context menu item.
The display format of the data value and the access type are as follows (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box):

Display Example (Default)			Memory Access Type
R*Data value*	Character color	Standard color	Read access
	Background color	Palegreen	
W*Data value*	Character color	Standard color	Write access
	Background color	Orange	
RW*Data value*	Character color	Standard color	Read and write access
	Background color	Paleturquoise	
VECT*Data value*	Character color	Standard color	Vector read access
	Background color	Palegreen	

[Toolbar]

	Acquires the latest data from the debug tool, and updates the contents of this panel. This item is disabled while the tracer is running.
	Clears the trace memory and the display of this panel (initialized). This item is disabled while the tracer is running.
	Starts the tracer operation. The content currently being displayed in this panel is cleared. This item is disabled while the tracer is running.
	Stops the tracer operation. The contents of trace data newly acquired are displayed. This item is disabled while the tracer is stopped.
	Opens the Trace Search dialog box.
Notation	The following buttons to change the notation of a data value are displayed. This item is disabled while the tracer is running.
	Displays values on this panel in hexadecimal number (default).
	Displays values on this panel in decimal number.
	Displays values on this panel in octal number.
	Displays values on this panel in binary number.
	Links with the Editor panel.
	Links with the Disassemble panel.

	Sets to the Mixed display mode as the display mode (default). This item is disabled while the tracer is running.
	Sets to the Disassemble display mode as the display mode. This item is disabled while the tracer is running.
	Sets to the Source display mode as the display mode. This item is disabled while the tracer is running.

[[File] menu (Trace panel-dedicated items)]

The following items are exclusive for the [File] menu in the Trace panel (other items are common to all the panels). Note that all these items are disabled during execution of a program.

Save Trace Data	Overwrites the contents of this panel to the previously saved text file (*.txt)/CSV file (*.csv) (see "(e) Saving trace data"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Trace Data As...]. This item is disabled while the tracer is running.
Save Trace Data As...	Opens the Data Save dialog box to newly save the contents of this panel to the specified text file (*.txt)/CSV file (*.csv) (see "(e) Saving trace data"). This item is disabled while the tracer is running.

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

The following items are exclusive for [Edit] menu in the Trace panel (all other items are disabled).

Copy	Copies the contents of the selected line to the clipboard (multiple line selections impossible). This item is disabled while the tracer is running.
Find...	Opens the Trace Search dialog box.

[Context menu]

Clear Trace	Clears the trace memory and the display of this panel (initialized). This item is disabled while the tracer is running.
Start Trace	Starts the tracer operation (see "(2) Restart collection of execution history"). The content currently being displayed in this panel is cleared. This item is disabled while the tracer is running.
Stop Trace	Stops the tracer operation (see "(1) Stop collection of execution history temporarily"). The contents of trace data newly acquired are displayed. This item is disabled while the tracer is stopped.
Find...	Opens the Trace Search dialog box. This item is disabled while the tracer is running.
Copy	Copies the contents of the selected line to the clipboard (multiple line selections impossible). This item is disabled while the tracer is running.
Mixed Display	Sets to the Mixed display mode as the display mode. This item is disabled while the tracer is running.

Disassemble View	Sets to the Disassemble display mode as the display mode. This item is disabled while the tracer is running.
Source View	Sets to the Source display mode as the display mode. This item is disabled while the tracer is running.
Notation	The following cascade menus are displayed to specify the notation. This item is disabled while the tracer is running.
Hexadecimal number	Displays values on this panel in hexadecimal number (default).
Decimal	Displays values on this panel in decimal number.
Octal	Displays values on this panel in octal number.
Binary	Displays values on this panel in binary number.
Window Connecting	The following cascade menus are displayed to link with other panels (see "(c) Linking with other panels").
Connect Source Window	Links with the Editor panel.
Connect Disassemble Window	Links with the Disassemble panel.
Jump to Disassemble	Opens the Disassemble panel (Disassemble1) and jumps to the fetch address corresponding to the line at the caret position in this panel.
Jump to Source	Opens the Editor panel and jumps to the source line corresponding to the line at the caret position in this panel.
Jump to Memory	Opens the Memory panel and jumps to the memory value corresponding to the line at the caret position in this panel.

Events panel

This panel is used to display the detailed information about the events that are set on the Editor panel/Disassemble panel/Watch panel. On this panel, you can change the setting state of the event between valid/invalid and delete the event (see "2.16 Manage Events").

This panel appears only when connected to the debug tool.

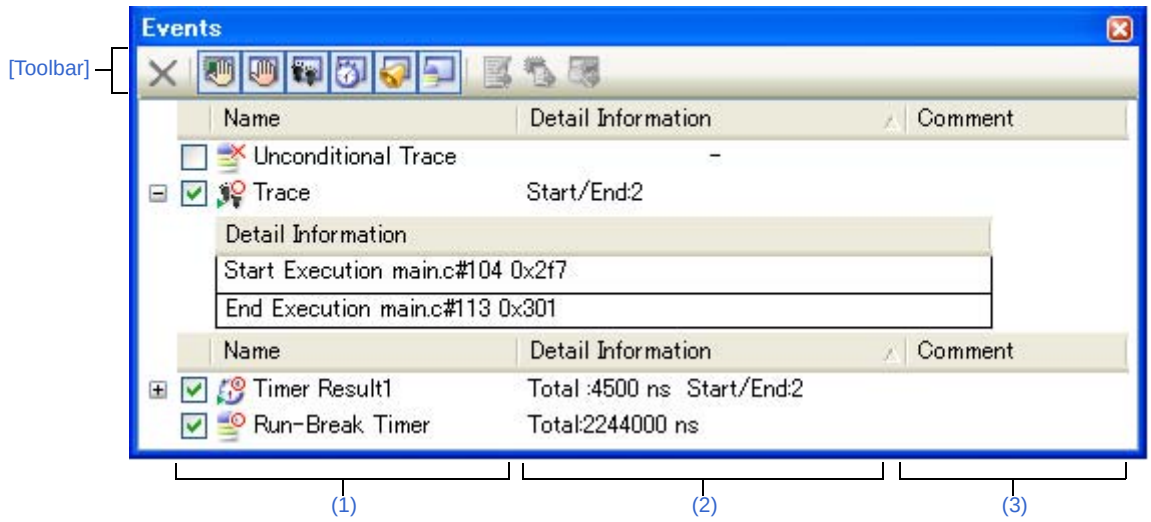
Remarks 1. Also see "2.16.6 Notes for setting events" for details on events (e.g. limits on the number of enabled events).

2. Events set via the Function List panel or Variable List panel of the analyze tool (Program Analyzer) are also managed on this panel.

3. This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

4. When the separator line of each area in this panel is double-clicked, the width of the area changes to the shortest possible size that can display the contents of the area.

Figure A-38. Events Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[Edit] menu (Events panel-dedicated items)]
- [Context menu]

[How to open]

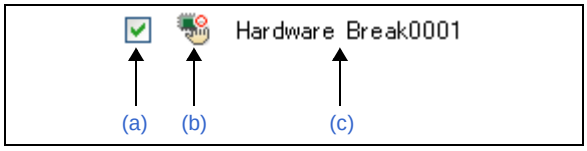
- From the [View] menu, select [Event].
- **[Simulator]**

  On the Editor panel/Disassemble panel, select [Timer Settings] >> [View Result of Timer] from the context menu.

[Description of each area]

(1) [Name] area

A list of the event names that have currently been set is displayed in the form shown below.



Remark It is possible to limit the event to be displayed by clicking the button on the toolbar (see "[Toolbar]").

(a) Check box

The setting state of the event is displayed/changed.

Note that the Event mark is changed depending on the setting state of the event.

☑	Valid state	Event occurs when the specified condition is met. It is possible to set the event to an invalid state by removing the check.
☐	Invalid state	Event does not occur when the specified condition is met. It is possible to set the event to a valid state by removing the check.
■	Suspended state	The conditions that have been specified cannot be set with the program of the debugging target. It is not possible to operate the check box.

Remarks 1. Both of the Timer Start event and Timer Stop event is must be set for the Timer Result event. Therefore, it is not possible to set a particular event to a valid state by only the setting of one of these (at the same time as both events are set, they are treated as grouped events as a Timer Result).

2. It is not possible to set the Run-Break Timer event to an invalid/suspended state.

3. The setting of the Unconditional Trace event and the Trace event to valid or invalid state is exclusively controlled. Therefore, the Unconditional Trace event, which is a built-in event, is valid state by default, but if either a trace start event/trace end event is set, it automatically becomes invalid state, and the Trace event, which is a event name that is collectively called with a trace start event and a trace end event, becomes valid state. Conversely, if the set Trace event is invalid state, the Unconditional Trace event automatically becomes valid state.

(b) Event mark

The event mark shows the type of event, and in addition shows the current setting state.

The meanings of the marks displayed are as follows:

Table A-8. Event Mark

Event Type	Valid State	Invalid State	Suspended State	Note
Hardware Break				Including a hardware break point
Software Break				Including a software break point
Break at start of function				A break event that can be set via the analyze tool.
Access break to variable				

Event Type	Valid State	Invalid State	Suspended State	Note
Unconditional Trace			None	-
Run-Break Timer		None	None	-
Trace				Displayed on only the Events panel
Trace start				Displayed on only the Editor panel/ Disassemble panel
Trace end				
Timer Result				Displayed on only the Events panel
Timer start				Displayed on only the Editor panel/ Disassemble panel
Timer end				
Point Trace				-
Printf (Action event)				-
Setting of two or more events	 Note 1	 Note 2	 Note 3	Displayed on only the Editor panel/ Disassemble panel

Notes 1. There is one or more event with valid state.

2. There is no event with valid state and at least one event with invalid state.

3. All the set events are suspended state.

(c) Event name

The event type and ID number are displayed as the event name.

A number from 0001 is automatically provided as the ID number for each event (no renumbering of the ID number is done even in the event that an event that has been set once is deleted).

Event types that are displayed are as follows:

Table A-9. Event Type

Event Type	Description
Hardware Break (Break[Note 1])	Breaks the program when the condition is met while the debug tool monitors the break condition all the time during program execution. -> See "2.9.3 Stop the program at the arbitrary position (breakpoint)" -> See "2.9.4 Stop the program at the arbitrary position (break event)" -> See "2.9.5 Stop the program with the access to variables/I/O registers"
Software Break (Break[Note 1])	Breaks the program when the instruction, which an address code to break is rewritten for the break instruction, is executed. -> See "2.9.3 Stop the program at the arbitrary position (breakpoint)"
Break at start of function	This event type is a Hardware Break (execution type) that is set in the Function panel of the analyze tool (Program Analyzer).
Access break to variable	This event type is a Hardware Break (access type) that is set in the Variable panel of the analyze tool (Program Analyzer).
Unconditional Trace	Automatically collects the trace data with start of a program execution, and stops collecting the trace data with stop of the program execution. This event cannot be deleted because of the built-in event[Note 2] (this event is set to a Valid state by default). -> See "2.12.2 Collect execution history until stop of the execution"

Event Type	Description
Run-Break Timer	Automatically measures the execution time of a program with start of the program execution, and stops the measurement with stop of the program execution. This event cannot be deleted because of the built-in event[Note 2] (this event is set to a Valid state by default). -> See "2.13.1 Measure execution time until stop of the execution"
Trace	Starts/stops collecting the trace data when the condition specified with a trace start event and a trace end event is met (this event is displayed when either a trace start event or a trace end event is set). -> See "2.12.3 Collect execution history in the arbitrary section"
Timer Result *n*	Starts/stops measuring the execution time of a program when the condition specified with a timer start event and a timer end event is met (this event is displayed when either a timer start event or a timer end event is set). "*n*" indicates the channel number in which a Timer Result event is set. -> See "2.13.2 Measure execution time in the arbitrary section [Simulator]"
Point Trace	Records the information as the trace data only when accessing the specified variable or I/O register during execution of a program. -> See "2.12.4 Collect execution history only when the condition is met [Simulator]"
Printf	Executes printf command in software processing after temporary stopping a program in execution at an arbitrary position (action event). -> See "2.15.1 Inset printf"

Notes 1. A breakpoint that is set by a one click operation of the mouse is displayed "Break" (see "(1) Set a breakpoint").

2. This is set in the debug tool by default.

(2) [Detail Information] area

Detailed information about each event is displayed.

The contents of the information that is displayed differ depending on the event type as follows:

Table A-10. Detailed Information with Event Type

Event Type	Displayed Content[Note 1]	
Hardware Break (Condition: execution)	Format1	*<Condition to occur> <File name#Line number> <Address>*
	Example	Before Execution main.c#39 0x100
		After Execution sub.c#100 0x200
		Before Execution - 0x300
		Execution main.c#39 0x300 **[Simulator]**
	Format2	*<Condition to occur> <Symbol + Offset> <Address>*
	Example	Before Execution funcA + 0x10 0x100
		After Execution funcB + 0x20 0x200
		Before Execution - 0x300

Event Type	Displayed Content[Note 1]	
Hardware Break (Condition: access)	Format1	<*Condition to occur*> <*File name#Variable name*> <*Address(range)*> <*Comparison condition*> <*Comparison value*>
	Example	Read main.c#variable1 0x100 - 0x101 == 0x5
		Write sub.c#variable2 0x200 - 0x200 == 0x7
		Read/Write sub2.c#variable3 0x300 - 0x303 == 0x8
	Format2	<*Condition to occur*> <*File name#Function name#Variable name*> <*Address(range)*> <*Comparison condition*> <*Comparison value*>
	Example	Read main.c#func1#variable1 0x100 - 0x101 == 0x10
	Format3	<*Condition to occur*> <*Variable name*> <*Address(range)*> <*Comparison condition*> <*Comparison value*>
	Example	Write variable1 0x100 - 0x101 == 0x10
Software Break	Format1	<*Condition to occur*> <*File name#Line number*> <*Address*>
	Example	Before Execution main.c#40 0x102
		Before Execution sub.c#101 0x204
	Format2	<*Condition to occur*> <*Symbol + Offset*> <*Address*>
	Example	Before Execution funcA + 0x12 0x102
Unconditional Trace	Format	-
	Example	-
Run-Break Timer	Format	Total: <*Total execution time*>
	Example	Total: 1000ms
		Total: OVERFLOW
Trace (Condition: execution)	Format	Total of Start/End: <*Total number of trace start/trace end events*> [Note 2] <Start/End> <*Detailed information of trace start/trace end event*>
	Example	Total of Start/End: 4 - Start After Execution main.c#1000x300 - Start After Execution funcA + 0x1000x300 - End After Execution main.c#2000x100 - End After Execution funcA + 0x100x100
Timer Result *n* (Condition: execution)	Format	Total:<*Total execution time* > Total of Start/End: <*Total number of timer start event/timer end event*>[Note 2] - <*Total execution time*> <*Pass Count*> <*Average*> <*Max*> <*Min*> - <Start/End> <*Detailed information of timer start event/timer end event*>
	Example	Total: 10ms Total of Start/End: 4 - Total: 10ms Pass Count: 5 Average: 2ms Max: 4ms Min: 1ms - Start After Execution main.c#1000x300 - Start After Execution funcA + 0x300x100 - End After Execution main.c#1000x300 - End After Execution funcA + 0x500x100

Event Type	Displayed Content[Note 1]	
Point Trace (Condition: access)	Format1	<*Condition to occur*> <*Variable name*> <*Variable address*>
	Example	Read variable1 0x100
	Format2	<*Condition to occur*> <*File name*#*Variable name*> <*Variable address*>
	Example	Write sub.c#variable2 0x200
	Format3	<*Condition to occur*> <*File name*#Function name#*Variable name*> <*Variable address*>
	Example	Read/Write sub.c#func1#variabl3 0x300
Printf (Action event)	Format	<*Condition to occur*> <*File name#Line number*> <*Address*> <*Setting of Printf event*>
	Example	Before Execution main.c#39 0x100 aaa, bbb, ccc
		After Execution sub.c#100 0x200 Result of aaa : aaa

Notes 1. Following are the details on the display format.

<*Condition to occur*>	Displays one of the following conditions. **[Full-spec emulator][E1][E20]** Execution: Before Execution or After Execution Access: Read, Write, Read/Write **[Simulator]** Execution: Execution Access: Read, Write, Read/Write
<*File name#Line number*>	Shows the line number of the source. Display format is the same as the watch type scope specification expression. When multiple load module files are downloaded, <*Load module file name*$*File name#Line number*> is displayed. For those events set in the Disassemble panel, display <*Line number*> in the format <*Symbol + offset*> in the condition below. - Line information exists and the specified position that the event is set not the top of the line information - Line information does not exist and symbol information exists. Show <*Line number*> in "-" in the following condition. - Line information and symbol information does not exist.
<*Variable name*>	Shows the variable name in the source file. Display format is the same as the watch type scope specification expression.
<*Comparison condition*>	Condition to compare (==) is shown. If the comparison value is not specified, comparison condition is not shown.
<*Comparison value*>	Comparison value is shown. If the comparison value is not specified, comparison condition is not shown.
<*Address*>	Address in the memory area is shown (only in hex number).
<*Start/End*>	Shows whether the contents of the detailed information is start event or the stop event.
<*Pass Count*>	Shows the measurement result of the pass count of the timer. If a timer overflow occurs (see "2.13.3 Measurable time ranges"), or if the illegal value was acquired, "OVERFLOW" is displayed. If measurements have not been performed yet, "Not measured" is displayed.

<*Total*>	Shows the measurement result of the timer total execution time. The unit is either of ns/µs,/ms/s/min/clock (if, however, the unit is in "min", a value in "s" unit also appears). If a timer overflow occurs (see "2.13.3 Measurable time ranges"), or if the illegal value was acquired, "OVERFLOW" is displayed. If measurements have not been performed yet, "Not measured" is displayed.
<*Average*>	Shows the measurement result of average execution of the timer. The unit is either of ns/µs,/ms/s/min/clock (if, however, the unit is in "min", a value in "s" unit also appears). If a timer overflow occurs (see "2.13.3 Measurable time ranges"), or if the illegal value was acquired, "OVERFLOW" is displayed. If measurements have not been performed yet, "Not measured" is displayed.
<*Max*>	Shows the measurement result of the maximum execution time of the timer. The unit is either of ns/µs,/ms/s/min/clock (if, however, the unit is in "min", a value in "s" unit also appears). If a timer overflow occurs (see "2.13.3 Measurable time ranges"), or if the illegal value was acquired, "OVERFLOW" is displayed. If measurements have not been performed yet, "Not measured" is displayed.
<*Min*>	Shows the measurement result of the minimum execution time of the timer. The unit is either of ns/µs,/ms/s/min/clock (if, however, the unit is in "min", a value in "s" unit also appears). If a timer overflow occurs (see "2.13.3 Measurable time ranges"), or if the illegal value was acquired, "OVERFLOW" is displayed. If measurements have not been performed yet, "Not measured" is displayed.
<*Set print event*>	Shows the variable expression and the character strings specified in the Action Events dialog box.

2. Click this line to display the detailed information of the lower lines.

(3) [Comment] area

The user can write comments for each event that has been set.

To input comments, click on this area, or select [Edit Comment] form the context menu after selecting the event in which you want to input comments, and then input directly the desired text from the keyboard (the edit mode is cancelled by pressing down the [Esc] key).

After editing the comments, complete the editing by pressing the [Enter] key or moving the focus to outside the edit region. Up to a maximum of 256 characters can be inputted for the comments, and this is saved as the settings of the user during use.

[Toolbar]

	Deletes the selected event and event condition. Note that it is not possible to delete the built-in events (Unconditional Trace event and Run-Break Timer event).
	Displays events related to Hardware Break (default).
 **[Full-spec emulator]** **[E1][E20]**	Displays events related to Software Break (default).
	Displays events related to the trace (default).

	Displays events related to the timer (default).
	Displays events related to the action event (Printf event) (default).
	Displays events related to the built-in event (Unconditional Trace event/Run-Break Timer event) (default).
	Opens the Editor panel and jumps to the source line corresponding to the address where the selected event[Note] is being set.
	Opens the Disassemble panel and jumps to the disassemble results corresponding to the address where the selected event[Note] is being set.
	Opens the Memory panel and jumps to the memory corresponding to the address where the selected event[Note] is being set.

Note Events other than Trace events, Timer Result events and built-in events (Unconditional Trace events/Run-Break Timer events) can be objects of this button.

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

The following items are exclusive for [Edit] menu in the Events panel (all other items are disabled).

Delete	Deletes the selected event and event condition. Note that it is not possible to delete the built-in events (Unconditional Trace event and Run-Break Timer event).
Select All	Selects all the events displayed on the panel.
Find...	Opens the Find and Replace dialog box with selecting [Find in Files] tab.
Replace...	Opens the Find and Replace dialog box with selecting [Replace in Files] tab.

[Context menu]

Enable Event	Enables the selected event (valid state). Note that this item is disabled if the selected event is a valid state.
Disable Event	Disables the selected event (invalid state). Note that this item is disabled if the selected event is an invalid state.
Delete	Deletes the selected event. Note that it is not possible to delete the built-in events (Unconditional Trace event and Run-Break Timer event).
Select All	Selects all the events of this panel.
View Select	The following cascade menus are displayed to limit the event type to be displayed. All of the items have been selected by default.
Hardware Break	Displays events related to Hardware Break.
Software Break	Displays events related to Software Break.
Timer Event	Displays events related to the timer.
Trace Event	Displays events related to the trace.
Action Event	Displays events related to action events (Printf events).
Built-in Event	Displays events related to built-in events (Unconditional Trace event or Run-Break Timer event).
Timer Settings	The following cascade menus are displayed to do the settings related to the timer. Note that this item is enabled only when a timer-related event has been selected.

Init Timer	Initializes the timer used by the selected event (except for Run-Break Timer).
Nanosecond	Displays the result of a selected event measured by a timer in nanosecond (ns) units.
Microsecond	Displays the result of a selected event measured by a timer in microsecond (μs) units.
Millisecond	Displays the result of a selected event measured by a timer in millisecond (ms) units.
Second	Displays the result of a selected event measured by a timer in second (s) units.
Minute	Displays the result of a selected event measured by a timer in minute (min) units.
Clock	Displays the result of a selected event measured by a timer in clock units.
Jump to Memory	Opens the Memory panel (Memory1) and jumps to the memory corresponding to the address where the selected event[Note] is being set.
Jump to Disassemble	Opens the Disassemble panel (Disassemble1) and jumps to the disassemble results corresponding to the address where the selected event[Note] is being set.
Jump to Source	Opens the Editor panel and jumps to the source line corresponding to the address where the selected event[Note] is being set.
Edit Condition...	Opens one of the following dialog box to edit the selected event - For an action event (Printf event) Action Events dialog box
Edit Comment	Sets to the edit mode to input comments for the selected event. When comments are already present, all of that character string is set to a select state.

Note Events other than Trace events, Timer Result events and built-in events (Unconditional Trace events/Run-Break Timer events) can be objects of this item.

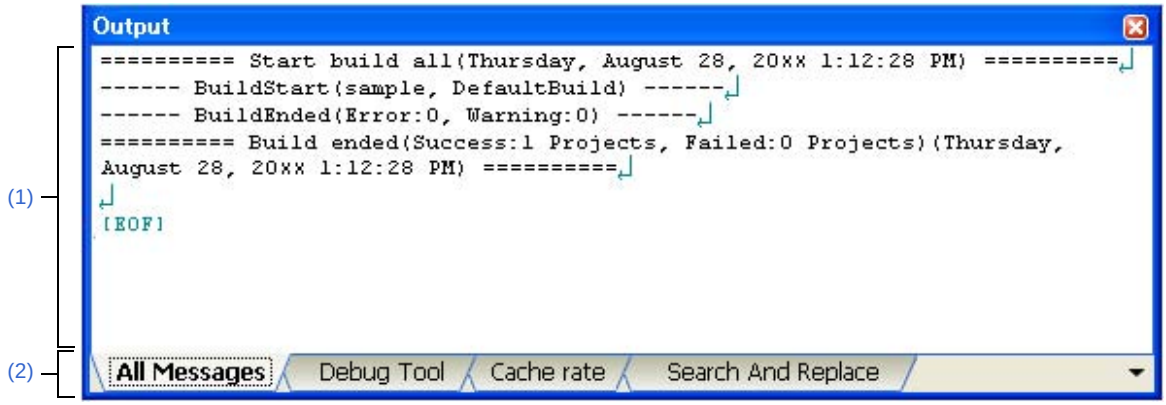
Output panel

This panel is used to display operation logs for various components (debug tool, design tool, build tool, etc.) provided by CubeSuite+, in addition to results of batch searches by the Find and Replace dialog box and a Printf event (see "2.15.1 Inset printf").

The messages are classified by the message origination tool and displayed on the individual tabs.

Remark This panel can be zoomed in and out by 100% in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

Figure A-39. Output Panel



This section describes the following.

- [How to open]
- [Description of each area]
- [[File] menu (Output panel-dedicated items)]
- [[Edit] menu (Output panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [Output].

[Description of each area]

(1) Message area

The output messages of each tool, search results and results by a Printf event are displayed.

In the case of search results (batch search), every time a search is performed, a new message will be displayed after the previous message is cleared (except for the [All Messages] tab).

The colors of message display differ with the type of message as shown below (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box).

Message Type	Display Example (Default)			Description
Normal message	AaBbCc	Character color	Black	Displayed with information notices
		Background color	White	
Warning message	AaBbCc	Character color	Blue	Displayed with warnings about operations
		Background color	Standard color	

Message Type	Display Example (Default)			Description
Error message	AaBbCc	Character color	Red	Displayed when there is a critical error, or when execution is not possible due to a operational mistake
		Background color	Light gray	

This area is provided with the following functions.

(a) Tag jump

By double-clicking on the output message, the Editor panel is opened and the number of the corresponding line in the corresponding file is displayed.

This allows you to jump from error messages that are output when building, etc. to the corresponding error line in the source file.

(b) Help display

If there is a caret on the line where a warning message or error message is being displayed, you can select [Help for Message] from the context menu. You can also display help for that line's message by pressing the [F1] key.

(c) Saving a log

The Save As dialog box can be opened by selecting the [File] menu >> [Save Output-*tab name* As...], and the contents that are displayed on the currently selected tab can be saved in a text file (*.txt) (messages on deselected tabs will not be saved).

(2) Tab selection area

Select the tab that indicates the origin of message.

The following tabs are available for the debug tool.

Tab Name	Description
All Messages	Displays operation logs for all components (debug tool, design tool, build tool, etc.) provided by CubeSuite+ in order of output.
Debug Tool	Displays messages output from the debug tool. Display only operation logs for the debug tool out of those for various components (debug tool, design tool, build tool, etc.) provided by CubeSuite+.
Cashe rate **[Simulator]**	Displays the cache hit rate (the ratio of the cache hit count to the cache access count).
Find and Replace	Displays the batch search results from the Find and Replace dialog box.

Caution Even if a new message is output on a deselected tab, tab selection will not automatically switch. In this case, "*" mark will be added in front of the tab name, indicating that a new message has been output.

[[File] menu (Output panel-dedicated items)]

The following items are exclusive for the [File] menu in the Output panel (other items are common to all the panels). Note that all these items are disabled during execution of a program.

Save Output-*tab name*	Overwrites the contents that are displayed on the currently selected tab to the preciously saved text file (see "(c) Saving a log"). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Output-*tab name* As...]. This item is disabled while building.
Save Output-*file name* As...	Opens the Save As dialog box to newly save the contents that are displayed on the currently selected tab to the specified text file (*.txt) (see "(c) Saving a log").

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

The following items are exclusive for [Edit] menu in the Output panel (all other items are disabled).

Copy	Copies the contents of the selected range to the clipboard as character string(s).
Select All	Selects all the messages displayed on the currently selected tab.
Find...	Opens the Find and Replace dialog box with selecting [Quick Find] tab.
Replace...	Opens the Find and Replace dialog box with selecting [Replace in Files] tab.

[Context menu]

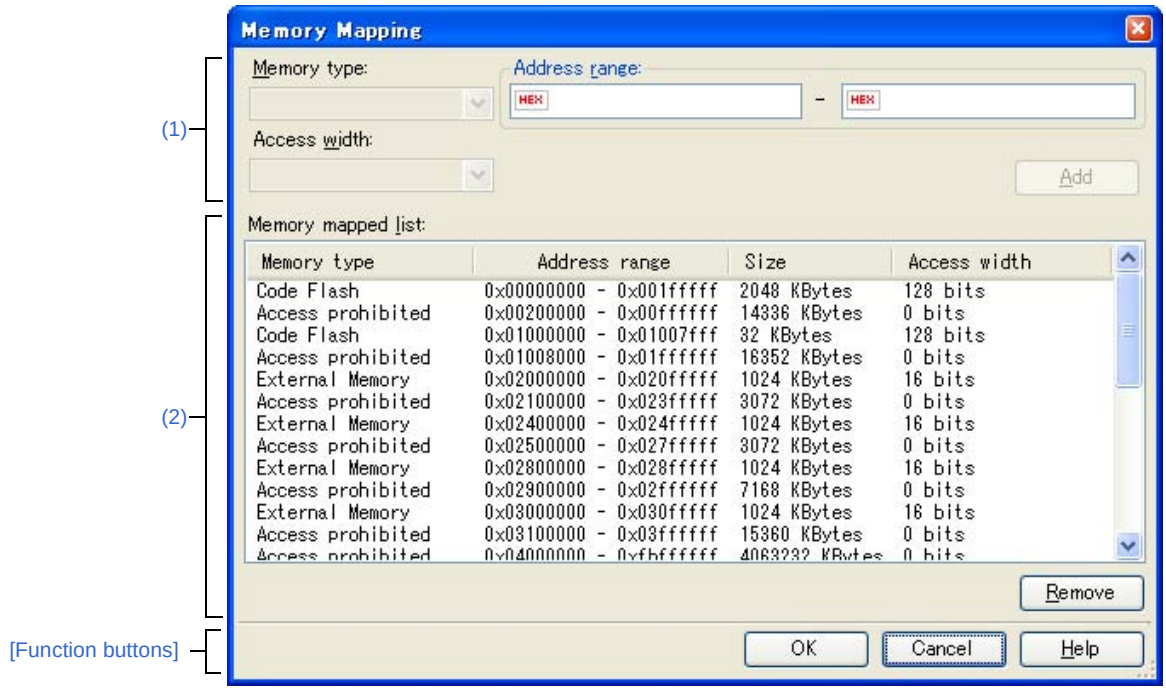
Copy	Copies the contents of the selected range to the clipboard as character string(s).
Select All	Selects all the messages displayed on the currently selected tab.
Clear	Deletes all the messages displayed on the currently selected tab.
Tag Jump	Opens the Editor panel and jumps to the number of the corresponding line in the corresponding file of the message at the caret position.
Stop Searching	Cancels the search currently being executed. This item is disabled when a search is not being executed.
Help for Message	Displays help for the massage on the current caret position. This item only applies to warning messages and error messages.

Memory Mapping dialog box

This dialog box is used to display the state of the memory mapping.

Caution When the selected microcontroller supports multi-core, this property displays the memory mapping status regarding a core (PE*n*) by switching selection between the target cores (see "2.7 Select a Core (PE)").

Figure A-40. Memory Mapping Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Debug Tool Settings] tab of the Property panel, click the [...] button displayed by selecting one of the values of the [Memory mappings] property in the [Memory] category.

Caution This dialog box cannot be opened during execution of a program.

[Description of each area]

(1) Added memory mapping specification area

This area is always invalid.

(2) [Memory mapped list] area

(a) List display

Information about the microcontroller's internal memory mapping is displayed. This area cannot be edited.

Memory type	Displays the memory types.
Address range	Displays the address range as <*Start address*> - <*End address*>. Display is fixed as "0x"-prefixed hexadecimal numbers.
Size	Displays size as a decimal number (unit: bytes/Kbytes[Note]).
Access width	Displays the access width (unit: bits).

Note Only in the case of multiple of 1024, displays in kilobyte units.

(b) Button

Button	Function
Remove	This button is always invalid.

[Function buttons]

Button	Function
OK	Closes this dialog box.
Cancel	Closes this dialog box.
Help	Displays the help for this dialog box.

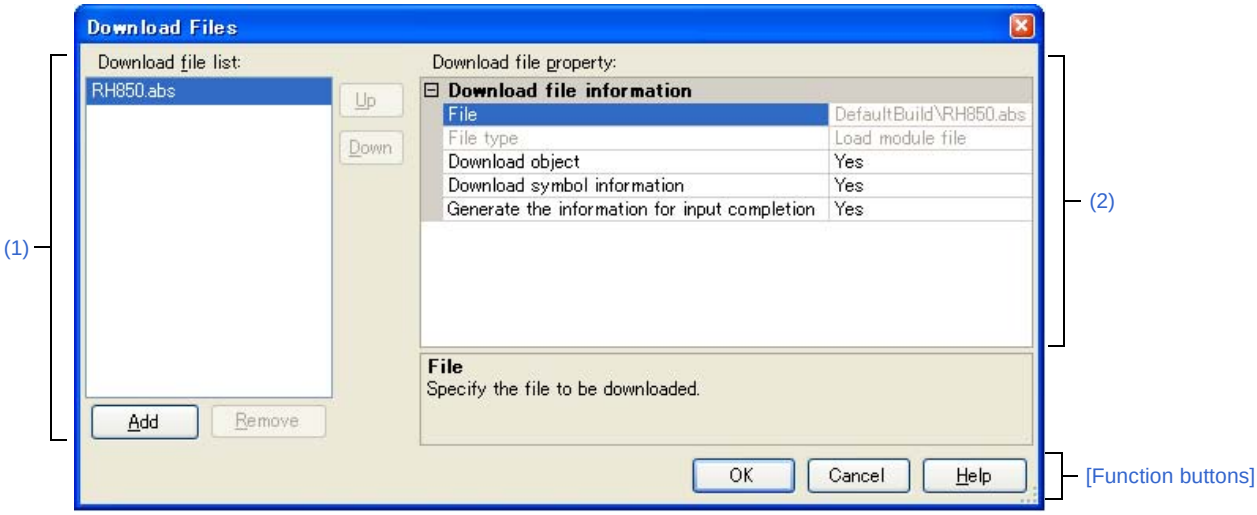
Download Files dialog box

This dialog box is used to select files for downloading and configure download conditions (see "2.5 Download/Upload Programs").

Note that files specified as build targets in a project (main project or sub-project) are automatically registered as download targets (they can be unregistered).

Caution This dialog box cannot be opened during execution of a program.

Figure A-41. Download Files Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Download File Settings] tab of the Property panel, click the [...] button displayed by selecting the [Download files] property in the [Download] category.

[Description of each area]

(1) [Download file list] area

(a) List display

Displays a list of files to download. The names of files specified as build targets in a project (main project or sub-project) are displayed by default (they can be removed).

Files are downloaded in the order that they are displayed here.

To add a new file to be downloaded, click the [Add] button in this area, then in the [Download file property] area, specify the download conditions of the file to add.

(b) Button

Button	Function
Up	Moves the selected file up one row in the list. Clicking this for the top file in the list has no effect.
Down	Moves the selected file down one row in the list. Clicking this for the bottom file in the list has no effect.
Add	Adds an empty item "-" to the list, and selects it. Specify the download conditions of the file to add in the [Download file property] area. Note that this button will be disabled if 20 files have already been registered.
Remove	Removes the selected file from the list. Note, however, that this button is disabled if the selected file is a project build target.

Remarks 1. By hovering the mouse cursor over a file name, the pass information of the file is pop-up displayed.

2. By dragging a file name with the mouse, the display order in the list can be changed.
Note, however, that the order of a project build target cannot be changed.

(2) [Download file property] area

(a) [Download file information]

This area is used to display or edit the download conditions of the file selected in the [Download file list] area.
It can also be used to specify the download conditions of new download files added via the [Add] button.

File	Specify the name of the file to download.		
	Default	*File name* (but it will be blank for newly added files)	
	Modifying	Directly enter from the keyboard, or specify with the Select Download File dialog box opened by clicking the [...] button[Note 1] appears at right by selecting this item.	
	Available values	See "Table 2-1. Downloadable File Formats" Up to 259 characters	
File type	Select the type of the file to download.		
	Default	Load module file	
	Modifying	Select from the drop-down list.	
	Available values	Load module file	Specifies a load module file (*.abs).
		Hex file	Specifies an Intel HEX file (*.hex).
		S-record file	Specifies a Motorola S-record file (*.mot).
		Binary data file	Specifies an binary file (*.bin).
Offset	Specify the offset from the address at which the file's download is to start. Note that this item appears only when [File type] is set to [Hex file] or [S record file].		
	Default	0	
	Modifying	Directly enter from the keyboard.	
	Available values	0x0 to 0xFFFFFFFF in hexadecimal number	

Start address	Specify the address at which to start the file's download. Note that this item appears only when [File type] is set to [Binary file].		
	Default	0	
	Modifying	Directly enter from the keyboard.	
	Available values	0x0 to 0xFFFFFFFF in hexadecimal number	
Download object	Select whether to download the object information from the specified file. Note that this item appears only when [File type] is set to [Load module file].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Downloads object information.
		No	Does not download object information.
Download symbol information	Select whether to download the symbol information from the specified file[Note 2]. Note that this item appears only when [File type] is set to [Load module file].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Downloads symbol information.
		No	Does not download symbol information.
Generate the information for input completion	Select whether to generate the information for the Symbol name completion function when downloading[Note 3]. Note that this item appears only when [File type] is set to [Load module file].		
	Default	Yes	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Generates the information for the symbol name completion function. (i.e. uses the symbol name completion function.)
		No	Does not generate the information for the symbol name completion function. (i.e. does not use the symbol name completion function.)

Notes 1. When a file specified as build target in the project is selected in the [Download file list] area, or when the program is executing, the [...] button does not appear.

2. If the symbol information have not been downloaded, the source level debugging cannot be performed.

3. When [Yes] is selected, the time taken for downloading and the memory usage on the host machine will increase. We recommend selecting [No] in this item if you do not intend to use the symbol name completion function.

[Function buttons]

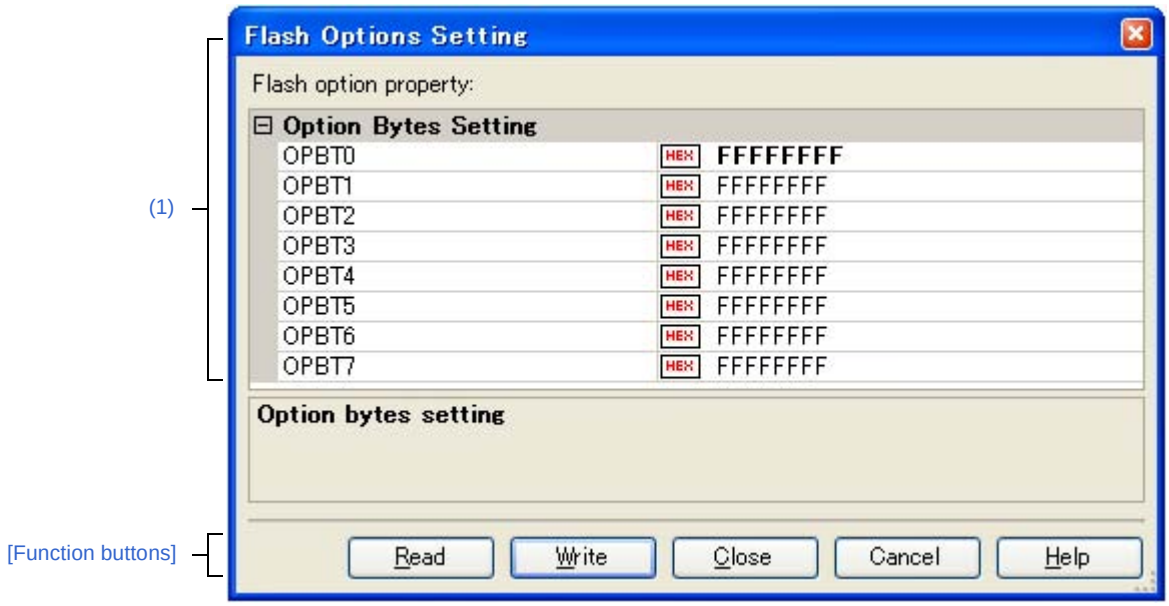
Button	Function
OK	Finishes configuring the download files, and closes this dialog box.
Cancel	Cancels any changes to the download files, and closes this dialog box.
Help	Displays the help for this dialog box.

Flash Options Setting dialog box [Full-spec emulator][E1][E20]

This dialog box is used to configure options for the flash memory incorporated in the microcontroller.
This dialog box appears only when connected to the debug tool.

Caution CPU reset may be generated automatically when you click the [Write] button after changing the configuration of this dialog box.

Figure A-42. Flash Options Setting Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Flash Options Settings] tab [Full-spec emulator][E1][E20] tab of the Property panel, click the [...] button displayed by selecting the [Flash options] property in the [Flash Options] category.

[Description of each area]

(1) [Flash option property] area

(a) [Option Bytes Setting]

You can configure the option bytes for the flash memory.

OPBT*0 - 7*	Specify the option bytes.	
	Default	*The value before shipment of the microcontroller*
	Modifying	Directly enter from the keyboard.
	Available values	0x0 to 0xFFFFFFFF in hexadecimal number

Caution The number of option bytes (OPB*0* - *7*) to be displayed differs with the selected microcontroller (the bytes for some numbers might actually be unused).

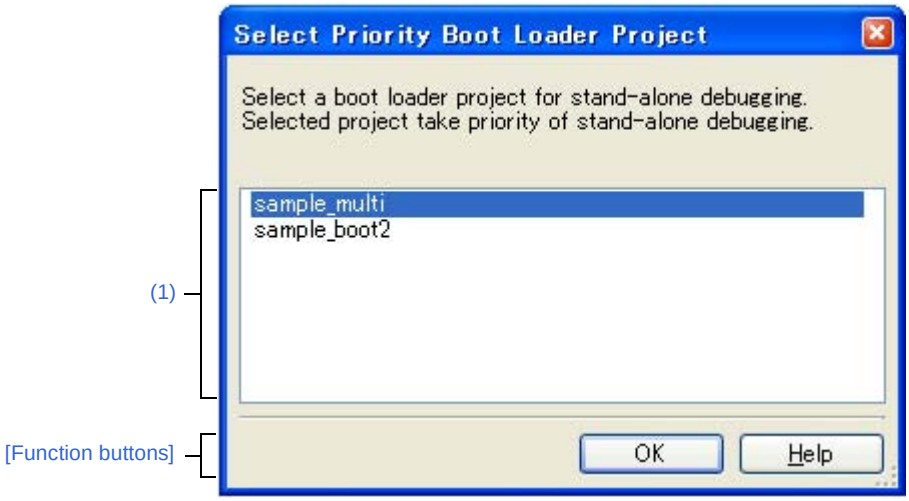
[Function buttons]

Button	Function
Read	Reads the values currently specified in the debug tool, and reflects them in this dialog box.
Write	Writes the currently set values in this dialog box to the debug tool, and reflects them in the project. Then, closes this dialog box.
Close	Specifies the currently set values in this dialog box to the project and closes this dialog box.
Cancel	Closes this dialog box without setting.
Help	Displays the help for this dialog box.

Select Priority Boot Loader Project dialog box

This dialog box is used to select a boot loader project for use in debugging of an individual core when its application project is included among multiple boot loader projects.

Figure A-43. Select Priority Boot Loader Project Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

When all of the [Priority Debugging] property in the [Debug] category on each boot loader project's Property panel are being set to [Yes] or [No] and the following operation is performed:

- Start communication with the debug tool (see "2.4.1 Connect to the debug tool")

[Description of each area]

(1) Priority boot loader project selection area

From the list of boot loader projects, select the one that you wish to take precedence over other projects in stand-along debugging.

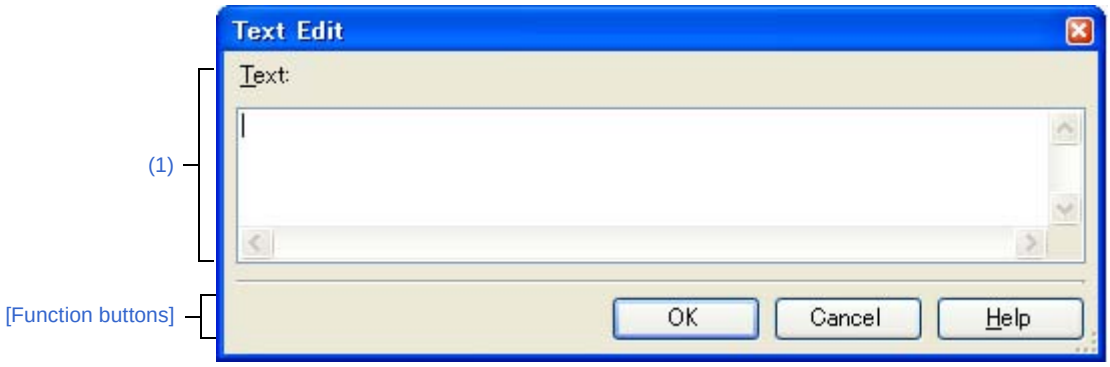
[Function buttons]

Button	Function
OK	The selected boot loader project will take precedence over other projects in stand-along debugging. The property setting ([Debug] category >> [Priority Debugging] property) for all boot loader projects is also changed accordingly.
Help	Displays the help for this dialog box.

Text Edit dialog box

This dialog box is used to input/modify character strings.

Figure A-44. Text Edit Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Hook Transaction Settings] tab of the Property panel, click the [...] button displayed by selecting one of the property in the [Hook Transaction Settings] category.

[Description of each area]

(1) [Text] area

Input/modify character strings in this area.

[Function buttons]

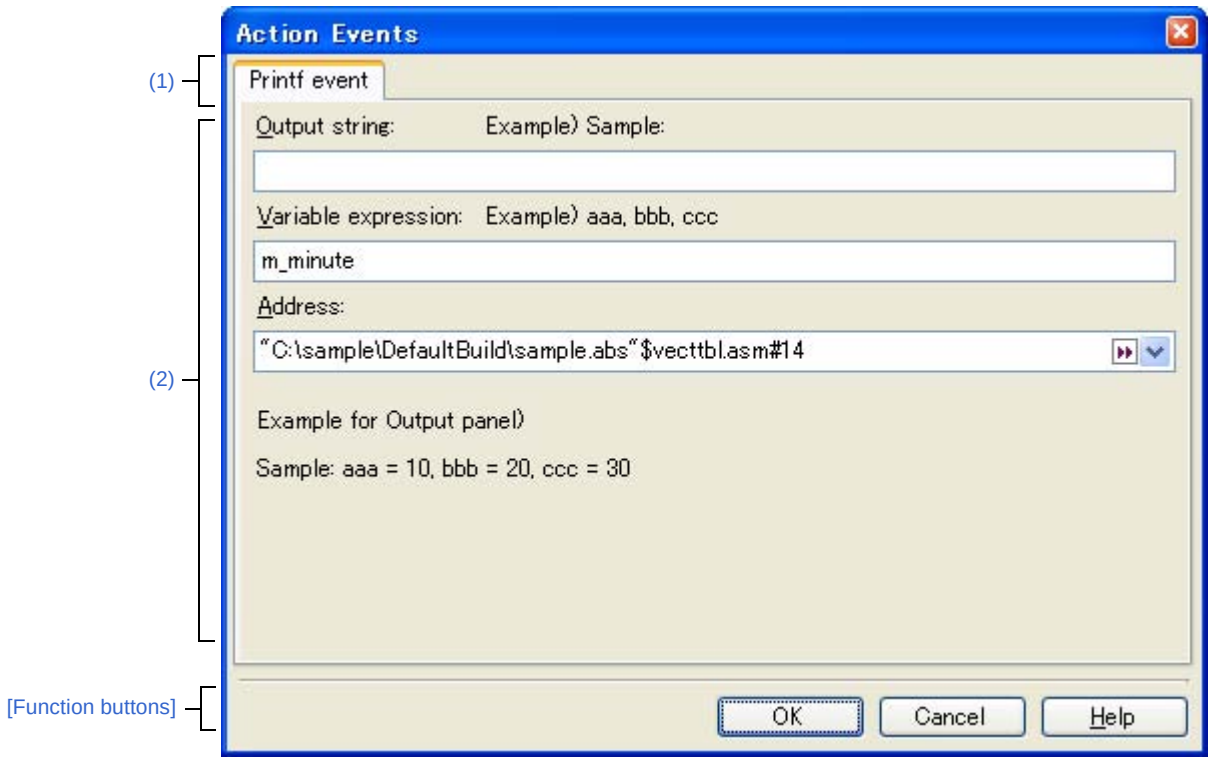
Button	Function
OK	Sets the input character strings to the caller panel/dialog box and closes this dialog box.
Cancel	Closes this dialog box.
Help	Displays the help for this dialog box.

Action Events dialog box

This dialog box is used to configure action events (see "2.15 Set an Action into Programs").
This dialog box appears only when connected to the debug tool.

Caution Also see "2.16.6 Notes for setting events" for details on Printf events (e.g. limits on the number of enabled events).

Figure A-45. Action Events Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Editor panel, move the caret to the line where you wish to set an action event, then select [Register Action Event...] from the context menu.
- On the Disassemble panel, move the caret to the address where you wish to set an action event, then select [Register Action Event...] from the context menu.
- On the Events panel, select an action event, then select [Edit Condition...] from the context menu.

[Description of each area]

(1) Tab selection area

Select a tab to switch the type of an action event to be set.
This dialog box has the following two tabs.

- [Printf event] tab

Caution If this dialog box is opened by selecting [Edit Condition...] from the context menu, this area does not appear.

(2) Event condition setting area

Use this area to configure detailed condition of an action event.

For details on how to setup an action event, see the section explaining the corresponding tab.

[Function buttons]

Button	Function
OK	Finishes configuring the action event, and sets it at the position specified in this dialog box.
Cancel	Cancels the action event settings and closes this dialog box.
Help	Displays the help for this dialog box.

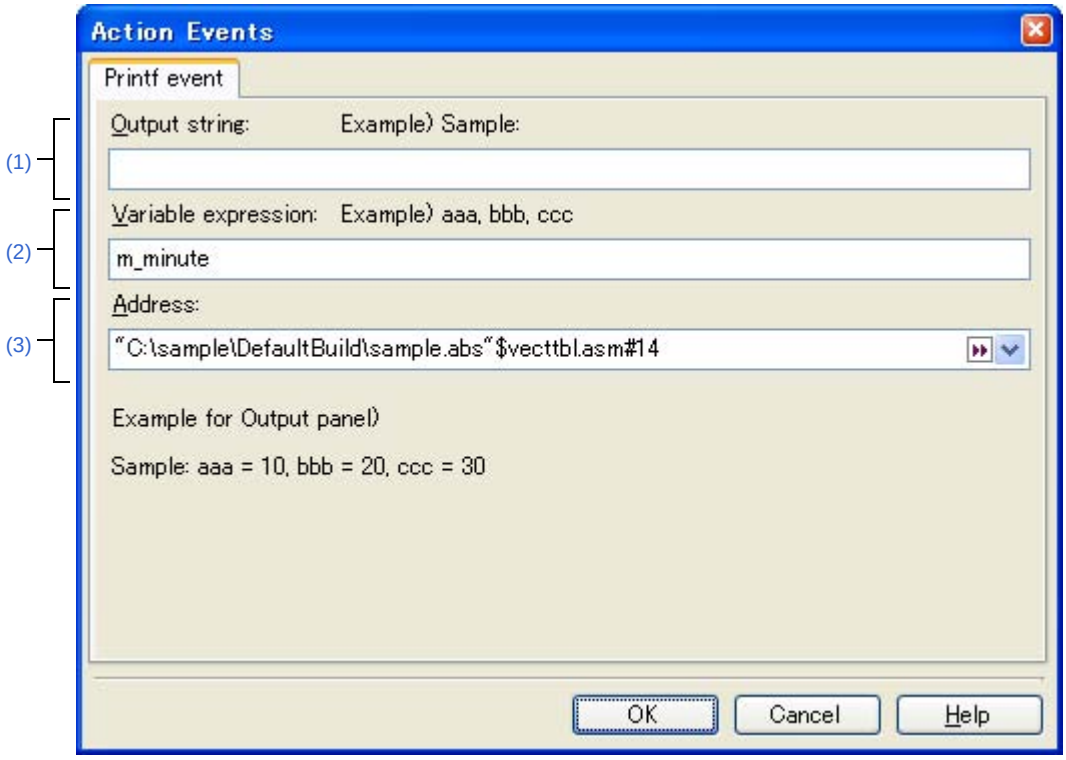
[Printf event] tab

This tab is used to configure Printf events as action events (see "2.15 Set an Action into Programs").

A Printf event momentarily stops the execution of the program at a specified location, and executes the printf command via software processing. When a Printf event is set, the program momentarily stops immediately before executing the command at the location where this event is set, and the value of the variable expression specified in this dialog box is output to the Output panel.

This dialog box appears only when connected to the debug tool.

Figure A-46. Action Events Dialog Box: [Printf event] Tab



This section describes the following.

- [How to open]
- [Description of each area]

[How to open]

- On the Editor panel, move the caret to the line where you wish to set a Printf event, then select [Register Action Event...] from the context menu.
- On the Disassemble panel, move the caret to the address where you wish to set a Printf event, then select [Register Action Event...] from the context menu.
- On the Events panel, select a Printf event, then select [Edit Condition...] from the context menu.

[Description of each area]

(1) [Output string] area

Type in the string to add to the Output panel directly via the keyboard (up to 1024 characters).

Note that the output string can only be one line (spaces allowed).

(2) [Variable expression] area

Specify the variable expression(s) for the Printf event.

Type a variable expression directly into the text box (up to 1024 characters).

You can specify up to 10 variable expressions for a single Printf event by separating them with commas (",").

If this dialog box opens with a variable expression selected in the Editor panel /Disassemble panel, the selected variable expression appears as the default.

The basic input format that can be specified as variable expressions and the values output by Printf event are as follows:

Table A-11. Relationship between Variable Expressions and Output Value (Printf Event)

Variable Expression	Output Value
Variable name of C language	Value of C language variable
Variable expression [*Variable expression*]	Element of array
Variable expression.Member name	Member of structure/union
Variable expression -> Member name	Member of structure/union that pointer designates
**Variable expression*	Value of pointer variable
&*Variable expression*	Location address
CPU register name	Value of the CPU register
I/O register name	I/O register value
Label name[Note], EQU symbol name[Note] and [immediate address]	Values of label, EQU symbol and immediate address

Note If the label name or EQU symbol name includes a "$," be sure to enclose the name in "{ }".
Example: {$Label}
Any imaginary number must be multiplied by an uppercase "I" (e.g. 1.0 + 2.0*I). When you specify the CPU register name "I", add ":REG" (e.g. I:REG) to distinguish it from the keyword "I".

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

(3) [Address] area

Specify the address at which to set the Printf event.

You can either type an address expression directly into the text box (up to 1024 characters), or select them from the input history via the drop-down list (up to 10 items). The address of the location currently being specified is displayed by default.

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

Note that the output result format by the Printf event in the Output panel are as follows:

Figure A-47. Output Result Format of Printf Event

```
Specified-characters  Variable-expression-1 = Value-1, Variable-expression-2 = Value-2, ...
```

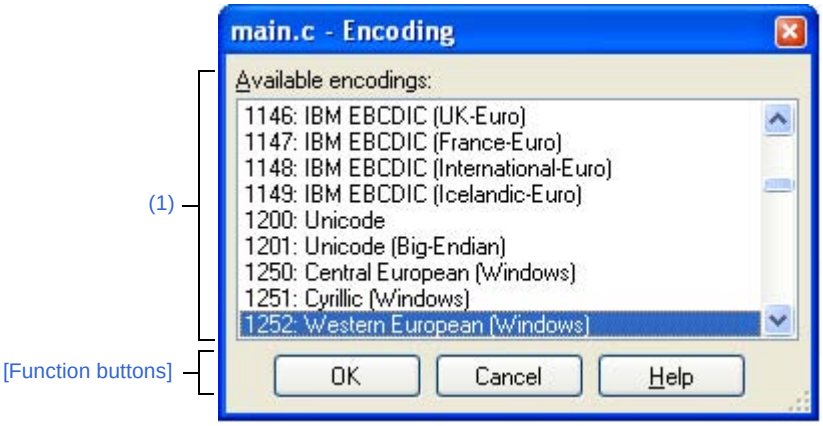
Specified characters	Characters specified with [Output string]

Variable expression 1 - 10	Characters specified with [Variable expression]
Value 1 - 10	Value of variable corresponds to " *Variable expression 1 - 10* ". The value is displayed in the default notation (see "Table A-7. Display Format of Watch-Expressions (Default)") according to the type of the variable (note, however, that "?" will be displayed if the specified variable expression cannot be obtained). Moreover, the value in hexadecimal number enclosing with "()" is also displayed (note, however, that "-" will be displayed if the value cannot be displayed in that notation).

Encoding dialog box

This dialog box is used to select a file-encoding.

Figure A-48. Encoding Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [File] menu, open the Open File dialog box by selecting [Open with Encoding...], and then click the [Open] button in the dialog box.

[Description of each area]

(1) [Available encodings]

Select the encoding to be set.

The encoding of the selected file is selected by default.

[Function buttons]

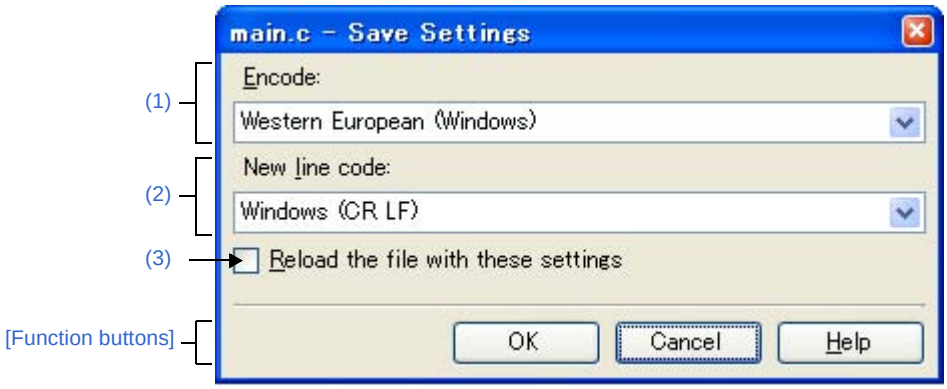
Button	Function
OK	Opens the selected file in the Open File dialog box using a selected file encoding.
Cancel	Not open the selected file in the Open File dialog box and closes this dialog box.
Help	Displays the help for this dialog box.

Save Settings dialog box

This dialog box is used to specify the encoding and the new line code of the file being edited in the Editor panel.

Remark The target file name is displayed on the title bar.

Figure A-49. Save Settings Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- With the Editor panel in focus, select [*File name* Save Settings…] from the [File] menu.

[Description of each area]

(1) [Encode] area

Select the encoding to be set from the drop-down list.

The items of the drop-down list are displayed according to the following sequence.

Note that the same encoding and encoding which are not supported by the current OS will not be displayed.

- *Current encoding of the file* (default)
- *Default encoding of the current OS*
- *Most recently used encodings* (maximum 4)
- *Popular encodings for current locale*

  (e.g. for United States locale it will be:
  - Western European (Windows)
  - Unicode (UTF-8)
- *All other encodings supported by the OS (in alphabetical order)*

(2) [New line code] area

Select the new line code to be set from the drop-down list.

Either of the following can be selected.

- Windows (CR LF)
- Macintosh (CR)
- Unix (LF)

An active newline entry is selected by default.

(3) [Reload the file with these settings]

☑	Reloads the file with the specified encoding and new line code when the [OK] button is clicked.
☐	Does not reload the file when the [OK] button is clicked (default).

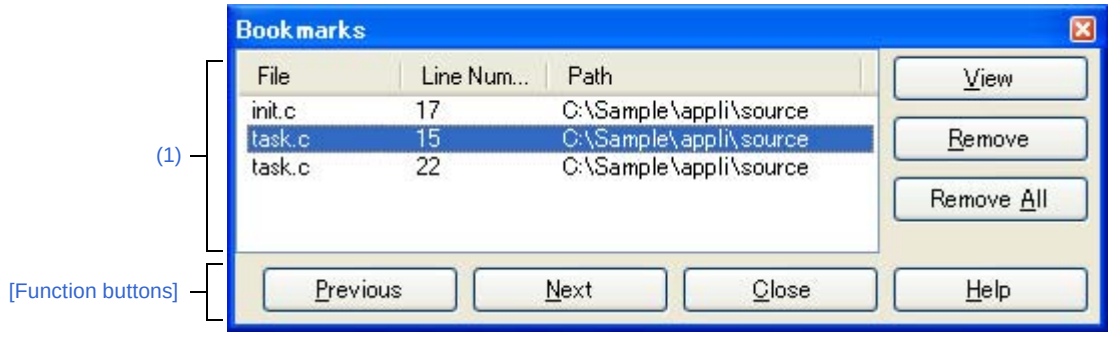
[Function buttons]

Button	Function
OK	Sets the selected encoding and newline code to the target file and closes this dialog box. If [Reload the file with these settings] is selected, sets the selected encoding and newline code to the target file and reloads the file. And then closes this dialog box.
Cancel	Cancels the setting and closes this dialog box.
Help	Displays the help for this dialog box.

Bookmarks dialog box

This dialog box is used to display the position where a bookmark is to be set or to delete a bookmark.

Figure A-50. Bookmarks Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- Click the button on the toolbar.
- From the [Edit] menu, select [Bookmark] >> [List Bookmarks...].
- On the Editor panel, select [Bookmark] >> [List Bookmarks...] from the context menu.

[Description of each area]

(1) Bookmark list area

This area displays a list of bookmarks that have been registered.

The bookmarks are listed alphabetically by file name for [Bookmark]. Bookmarks in the same file are listed in line number order.

When a bookmark is added to the Editor panel, a bookmark function is added.

In the bookmark list area, double-clicking on a line moves a caret to the corresponding position for the bookmark.

(a) [File]

Displays the name of the file (without any path) in which a bookmark is registered.

(b) [Line Number]

Displays the number of the line to which the bookmark is registered.

(c) [Path]

Displays the path of the file in which the bookmark is registered.

(d) Buttons

View	Moves a caret to the selected position for the bookmark. However, this button is disabled when no bookmark is selected, two or more bookmarks are selected, or no bookmark is registered.

Remove	Removes a selected bookmark. When two or more bookmarks are selected, all of those selected are removed. However, this button is disabled when no bookmark is selected or no bookmark is registered.
Remove All	Removes all the registered bookmarks. This button is disabled when no bookmark is registered.

Caution Registered bookmarks are not deleted even if the Editor panel is closed.
Note, however, that if the Editor panel in which a file that has never been saved is being displayed is closed, then registered bookmarks will be deleted.

[Function buttons]

Button	Function
Previous	Moves a caret to the position of the bookmark previous to the selected bookmark. This button is disabled in the following cases. - A bookmark shown in the first line has been selected. - No bookmark is selected. - Two or more bookmarks are selected. - No bookmark is registered. - Only one bookmark is registered.
Next	Moves a caret to the position of the bookmark next to the selected bookmark. This button is disabled in the following cases. - A bookmark shown in the last line has been selected. - No bookmark is selected. - Two or more bookmarks are selected. - No bookmark is registered. - Only one bookmark is registered.
Close	Closes this dialog box.
Help	Displays the help for this dialog box.

Column Number Settings dialog box

This dialog box is used to set the number of view columns of memory values on the Memory panel.

Figure A-51. Column Number Settings Dialog Box

This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Memory panel, select [View] >> [Column Number Settings…] from the context menu.

[Description of each area]

(1) [Column Number] area

Directly enter a decimal value as the number of columns you want to display.
The settable range depends on [Size Notation] currently being set on the Memory panel, as follows:

Size Notation	Settable Range
4 Bits	2 - 512[Note]
1 Byte	1 - 256
2 Bytes	1 - 128
4 Bytes	1 - 64
8 Bytes	1 - 32

Note Only an even number is specifiable (if an odd number is specified, then it will be changed to a value one greater than such odd number).

[Function buttons]

Button	Function
OK	Displays memory values in the specified number of columns.
Cancel	Cancels the settings and closes this dialog box.
Help	Displays the help for this dialog box.

Address Offset Settings dialog box

This dialog box is used to set an offset value of the start address in the address area on the Memory panel.

Figure A-52. Address Offset Settings Dialog Box

This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Memory panel, select [View] >> [Address Offset Value Settings...] from the context menu.

[Description of each area]

(1) [Address Offset Value] area

Directly enter a hexadecimal value as an offset value for the address display.

The settable range depends on the number of bytes of the memory currently being displayed in a line on the Memory panel, as follows:

Settable range: 0x0 - ("*Set value of [Size Notation]*" x "*The number of view columns*") -1

Example When "*Set value of [Size Notation]*" is 1 byte and "*The number of view columns*" is 16 columns:

Offset Value	Displayed Content of Address Area
0x0 (default)	0000 0010 0020
0x1	0001 0011 0021
0x2	0002 0012 0022

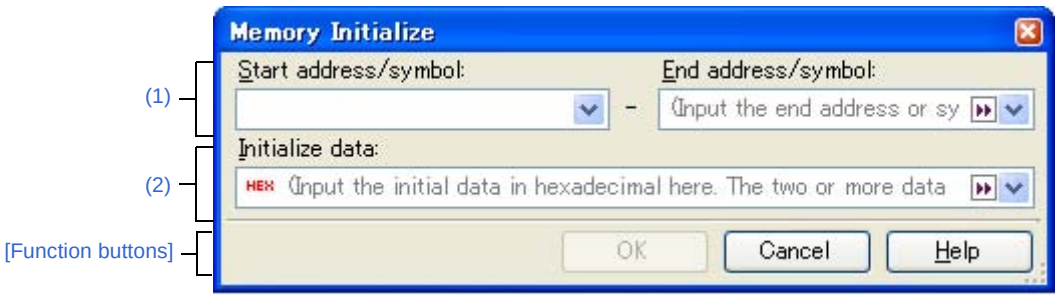
[Function buttons]

Button	Function
OK	Displays memory addresses with the specified offset value.
Cancel	Cancels the settings and closes this dialog box.
Help	Displays the help for this dialog box.

Memory Initialize dialog box

This dialog box is used to initialize memory (see "(6) Modify the memory contents in batch (initialize)").
The memory area in the specified address range is repeatedly overwritten with the specified initial data pattern.

Figure A-53. Memory Initialize Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Memory panel, select [Fill…] from the context menu.

[Description of each area]

(1) Range specification area

Specify the range of memory address to initialize via the [Start address/symbol] and [End address/symbol]. You can either type address expressions directly into the text boxes (up to 1024 characters), or select them from the input history via the drop-down list (up to 10 items).
The results of calculating the address expressions you have entered are treated as start and end addresses, respectively.
Note that address values greater than the microcontroller address space cannot be specified.

Caution **You cannot specify the range of address aligned across the different endian area.**

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").

(2) [Initialize data] area

Specify the initial value(s) with which to overwrite the memory.
You can either type the initial value into the text box directly in hexadecimal number (the value need not start with "0x"), or select one from the input history via the drop-down list (up to 10 items).
You can specify more than one initial value. Specify up to 16 values of up to 4 bytes (8 characters) each, separated by spaces.
Each initial value is parsed from the end of the string, with each two characters interpreted as a byte.
If the string has an odd number of characters, then the first character is interpreted as one byte.
Note that if a initial value consists of more than one byte, then the target memory is overwritten with the value converted into an array of bytes of the specified address range's endian, as follows:

Input Character Strings (Initial Value)	How Data is Overwritten (in Bytes)	
	Little Endian	Big Endian
1	01	01
0 12	00 12	00 12
00 012 345	00 12 00 45 03	00 00 12 03 45
000 12 000345	00 00 12 45 03 00	00 00 12 00 03 45

[Function buttons]

Button	Function
OK	The memory area in the specified address range is repeatedly overwritten with the specified initial data pattern. If the end address is reached in the middle of the pattern, then writing ends at that point.
Cancel	Cancels the memory initialization and closes this dialog box.
Help	Displays the help for this dialog box.

Memory Search dialog box

This dialog box is used to search memory (see "(5) Search the memory contents").

Search in either the Memory value area or Character strings area where the caret was located in the Memory panel immediately before this dialog box opened.

Figure A-54. Memory Search Dialog Box

This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Memory panel, select [Find…] from the context menu.

[Description of each area]

(1) [Search Data] area

Specify data to search.

You can either type a value directly into the text box (up to 256 bytes), or select one from the input history via the drop-down list (up to 10 items).

If the search is performed in the Memory value area of the Memory panel, the value must be entered in the same display format (notation and size) as that area.

If the search is performed in the Character strings area, then the target of the search must be a string. The specified string is converted into the encoding format displayed in that area, and searched for.

If a memory value was selected immediately prior to opening this dialog box, then that value will appear as default.

(2) [Search Range] area

Select the range to search from the following drop-down list.

Specify address range	Searches in the address range specified in the [Address] area.
Memory mapping	Searches within the selected memory mapping range. This list item displays the memory mappings set in the Memory Mapping dialog box. Display format: <*memory type*> <*address range*> <*size*>

(3) [Address] area

This item is only enabled if [Specify address range] is selected in the [Search Range] area.

Specify the range of memory address to search via the start and end addresses. You can either type address expressions directly into the text boxes (up to 1024 characters), or select them from the input history via the drop-down list (up to 10 items).

The results of calculating the address expressions you have entered are treated as start and end addresses, respectively.

Note, however, that the largest address that can be searched is the maximum address of the program space (0x03FFFFFF) (the mirror area cannot be searched).

In addition, an address value greater than the value expressed within 32 bits cannot be specified.

Remarks 1. A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").

2. If the start address field is left blank, it is treated as if "0x0" were specified.

3. If the end address field is left blank, then it is treated as if the maximum value in the microcontroller's address space were specified.

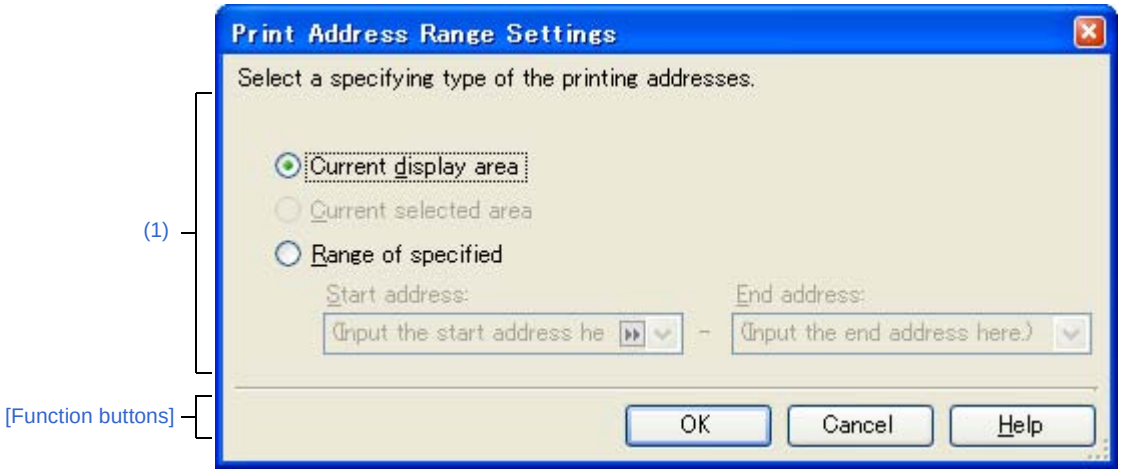
[Function buttons]

Button	Function
Search Backward	Searches upward within the range specified in the [Address] area or the [Search Range] area. The location found by the search is selected in the Memory panel. Note that if an illegal value is specified or while the program is being executed, a message will appear, and the memory search will not be performed. If focus moves to this dialog box while the memory panel is hidden or another panel has focus, then this button will be disabled.
Search Forward	Searches downward within the range specified in the [Address] area or the [Search Range] area. The location found by the search is selected in the Memory panel. Note that if an illegal value is specified or while the program is being executed, a message will appear, and the memory search will not be performed. If focus moves to this dialog box while the memory panel is hidden or another panel has focus, then this button will be disabled.
Cancel	Cancels the memory search and closes this dialog box.
Help	Displays the help for this dialog box.

Print Address Range Settings dialog box

This dialog box is used to specify the address range to print the contents of the Disassemble panel.

Figure A-55. Print Address Range Settings Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Disassemble panel, select [Print...] from the [File] menu.

[Description of each area]

(1) Range specification area

Select a range to print from the following option buttons.

(a) [Current display area] (default)

Prints only the contents of the Disassemble panel currently being displayed.

(b) [Current selected area]

Prints only the range currently being selected in the Disassemble panel.
Note, however, that this option button will be disabled when nothing is selected in the Disassemble panel.

(c) [Range of specified]

Specify the range of address to print via [Start address] and [End address]. You can either type address expressions directly into the text boxes (up to 1024 characters), or select them from the input history via the drop-down list (up to 10 items).

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").

[Function buttons]

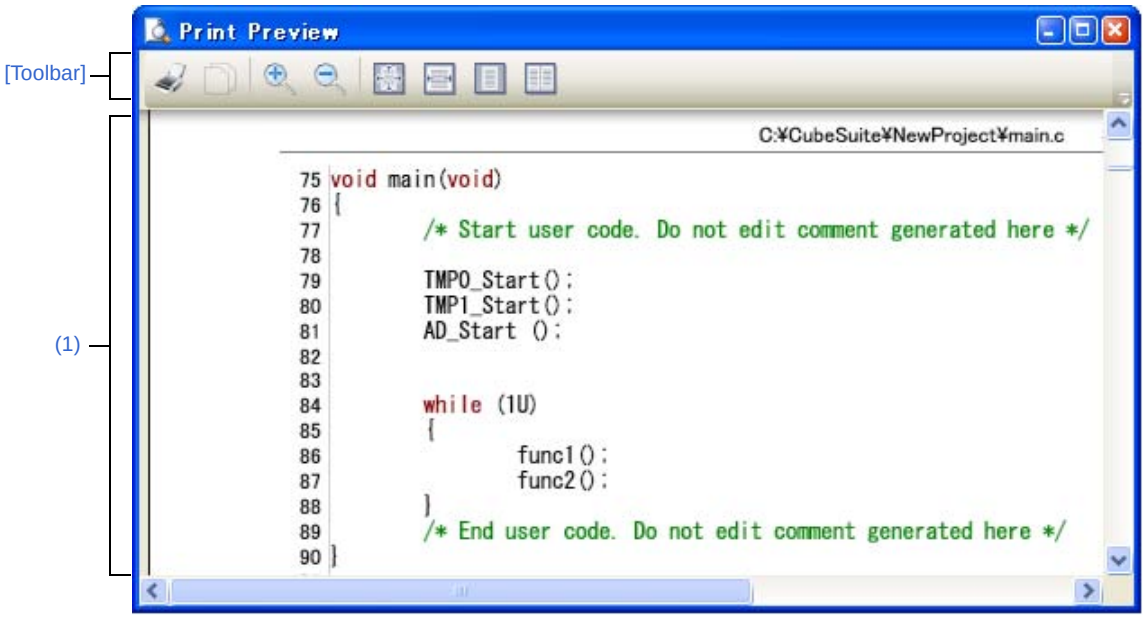
Button	Function
OK	Closes this dialog box and opens the Windows dialog box to print the contents of the specified range of the Disassemble panel.
Cancel	Cancels the range specification and closes this dialog box.
Help	Displays the help for this dialog box.

Print Preview window

This window is used to preview the file currently being displayed in the Editor panel before printing.

Remark This panel can be zoomed in and out by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

Figure A-56. Print Preview Window



The following items are explained here.

- [How to open]
- [Description of each area]
- [Toolbar]
- [Context menu]

[How to open]

- Focus the Editor panel, and then select [Print Preview] from the [File] menu.

[Description of each area]

(1) Preview area

This window displays a form showing a preview of how and what is printed.
The file name (fully qualified path) and the page number are displayed at the page header and page footer.
The display differs according to whether the debug tool is or is not connected, and when it is connected, to whether the display is in normal display mode or mixed display mode. Note, however, that columns that are hidden on the Editor panel are not displayed (these columns are not printed).

[Toolbar]

	Opens the Print dialog box provided by Windows to print the current Editor panel as shown by the print preview form.

	Copies the selection into the clipboard.
	Increases the size of the content.
	Decreases the size of the content.
	Displays the preview at 100-percent zoom (default).
	Fits the preview to the width of this window.
	Displays the whole page.
	Displays facing pages.

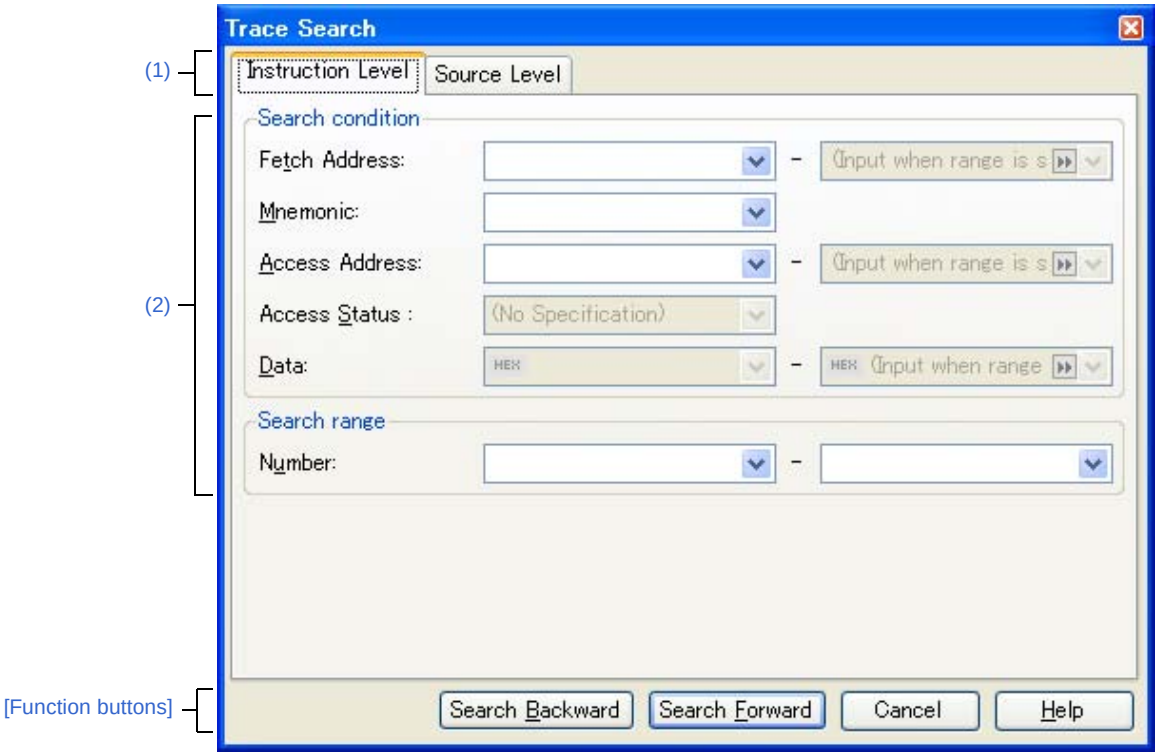
[Context menu]

Increase Zoom	Increases the size of the content.
Decrease Zoom	Decreases the size of the content.

Trace Search dialog box

This dialog box is used to search for trace data (see "2.12.8 Search the trace data").
The search can be performed at the instruction or source level.

Figure A-57. Trace Search Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Trace panel, select button on the toolbar.
- On the Trace panel, select [Find…] from the context menu.

[Description of each area]

(1) Tab selection area

Select a tab to switch the level of the search.
This dialog box has the following two tabs.

- [Instruction Level] tab
- [Source Level] tab

(2) Search parameter setup area

Use this area to configure detailed search parameters.

For details on the window elements and how to configure the parameters for a particular tab, see the section for the tab in question.

[Function buttons]

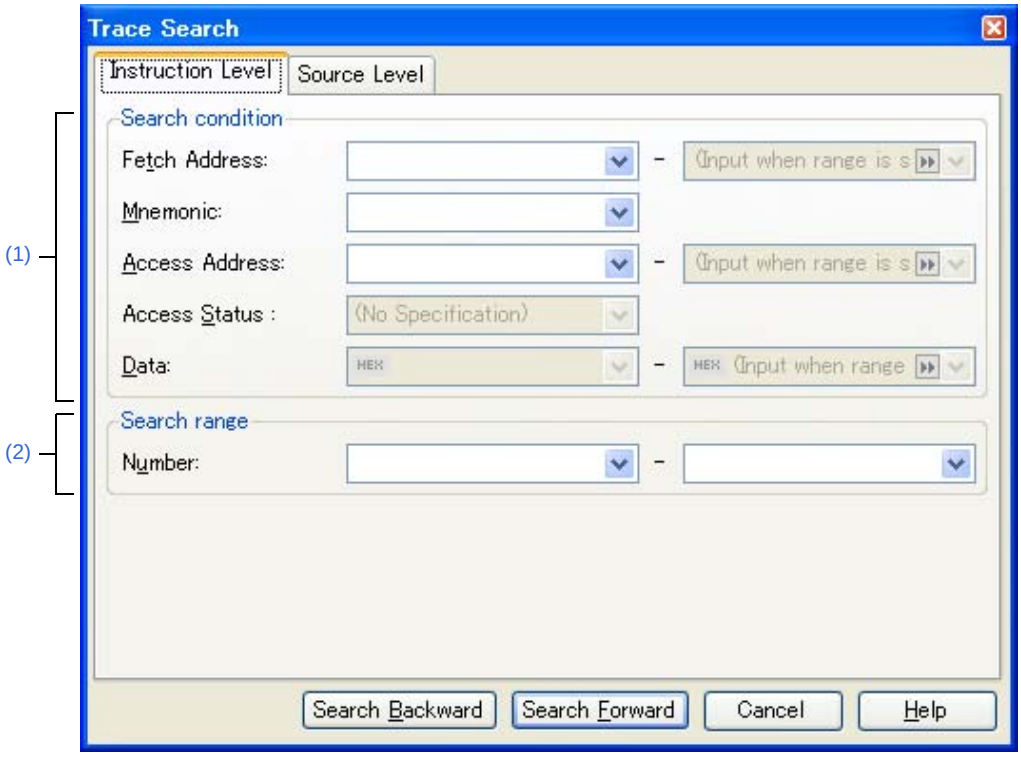
Button	Function
Search Backward	Searches upward (in the direction of larger to smaller numbers) within the specified range. Search matches are selected in the Trace panel. Note that if an illegal value is specified or while the program is being executed, a message will appear, and the trace data search will not be performed. If focus moves to this dialog box while the Trace panel is hidden or another panel has focus, then this button will be disabled.
Search Forward	Searches forward (in the direction of smaller to larger numbers) within the specified range. Search matches are selected in the Trace panel. Note that if an illegal value is specified or while the program is being executed, a message will appear, and the trace data search will not be performed. If focus moves to this dialog box while the Trace panel is hidden or another panel has focus, then this button will be disabled.
Cancel	Cancels the trace data search and closes this dialog box.
Help	Displays the help for this dialog box.

[Instruction Level] tab

This tab is used to search for the acquired trace data at the instruction level.

Caution If the Trace panel is set to Source display mode, then performing an instruction level search via this tab will not perform the target search correctly. In order to perform an instruction level search, set the mode to Mixed display mode or Disassemble display mode.

Figure A-58. Trace Search Dialog Box: [Instruction Level] Tab



This section describes the following.

- [How to open]
- [Description of each area]

[How to open]

- On the Trace panel, select button on the toolbar.
- On the Trace panel, select [Find…] from the context menu.

[Description of each area]

(1) [Search condition] area

(a) [Fetch Address]

Specify the fetch address if it is a required search parameter.

You can either type address expressions directly into the text boxes, or select them from the input history via the drop-down lists (up to 10 items).

The fetch address can also be specified as a range. In this case, specify a range by specifying address expressions in both the left and right text boxes.
If the right-hand text box is blank or contains the text [(Input value when range is specified)], then the fixed address specified in the left-hand text box will be searched.
Note that if an address value greater than the microcontroller address space is specified, the upper address value is masked.
In addition, an address value greater than the value expressed within 32 bits cannot be specified.

(b) [Mnemonic]

Specify the mnemonic if it is a required search parameter.
The specified character strings in this area are searched within the [Source/Disassemble] area of the Trace panel.
You can either type a mnemonic directly into the text boxes, or select one from the input history via the drop-down list (up to 10 items).
Searches are case-insensitive, and partial matches are also allowed.

(c) [Access Address]

Specify the access address if it is a required search parameter.
You can either type address expressions directly into the text boxes, or select them from the input history via the drop-down lists (up to 10 items).
The access address can also be specified as a range. In this case, specify a range by specifying address expressions in both the left and right text boxes.
If the right-hand text box is blank or contains the text [(Input value when range is specified)], then the fixed address specified in the left-hand text box will be searched.
Note that if an address value greater than the microcontroller address space is specified, the upper address value is masked.
In addition, an address value greater than the value expressed within 32 bits cannot be specified.

(d) [Access Status]

This item is only enabled if a value for [Access Address] is specified.
Select the access type from the following drop-down list.

(No Specification)	Does not limit access types.
Read/Write	Searches the location where a read or write access occurred.
Read	Searches the location where a read access occurred.
Write	Searches the location where a write access occurred.
Vector Read	Searches the location where a vector read access occurred.
DMA	This item is invalid.

(e) [Data]

This item is only enabled if a value for [Access Address] is specified.
Specify the access data.
You can either type the data directly into the text boxes (in hexadecimal number), or select it from the input history via the drop-down list (up to 10 items).
The data can also be specified as a range. In this case, specify a range by specifying data in both the left and right text boxes.
If the right-hand text box is blank or contains the text [(Input value when range is specified)], then the fixed data specified in the left-hand text box will be searched.

(2) [Search range] area

(a) [Number]

Specify the range within the trace data to search via the number displayed in the [Number] area of the Trace panel.

Specify the starting number in the left text box, and the ending number in the right text box ("0" to "last number" are specified by default).

You can either type the numbers directly into the text boxes (in base-10 format), or select them from the input history via the drop-down lists (up to 10 items).

If the left-hand text box is left blank, it is treated as if "0" were specified.

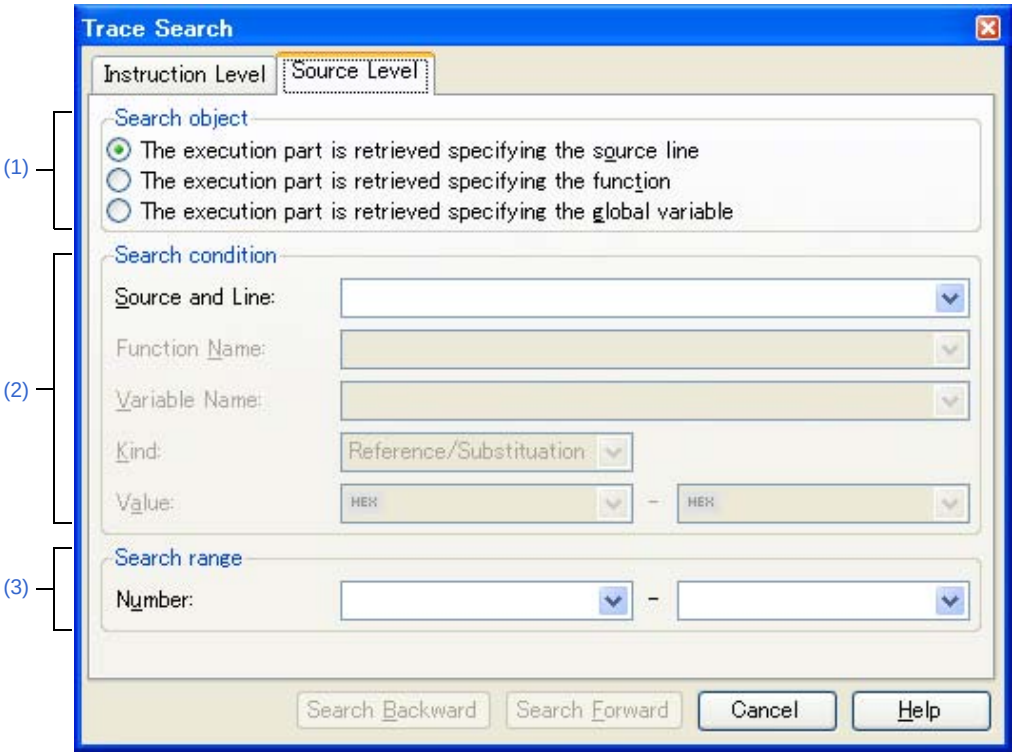
If the right-hand text box is left blank, it is treated as if the last number were specified.

[Source Level] tab

This tab is used to search for the acquired trace data at the source level.

Caution If the Trace panel is set to Disassemble display mode, then performing an source level search via this tab will not perform the target search correctly. In order to perform an source level search, set the mode to Mixed display mode or Source display mode.

Figure A-59. Trace Search Dialog Box: [Source Level] Tab



This section describes the following.

- [How to open]
- [Description of each area]

[How to open]

- On the Trace panel, select button on the toolbar.
- On the Trace panel, select [Find…] from the context menu.

[Description of each area]

(1) [Search object] area

Select the search object from the following option buttons.

The execution part is retrieved specifying the source line	Finds the execution location in the specified source line (default). Only [Source and Line] will be enabled as a search parameter.

The execution part is retrieved specifying the function	Finds the execution location in the specified function. Only [Function Name] will be enabled as a search parameter.
The execution part is retrieved specifying the global variable	Finds the location at which the specified global variable was accessed. Only [Variable Name], [Kind] and [Value] will be enabled as a search parameters.

(2) [Search condition] area

(a) [Source and Line]

This item is only enabled if [The execution part is retrieved specifying the source line] is selected.
The specified character strings in this area are searched within the [Line Number/Address] area of the Trace panel.
You can either type the character strings of the source line to be find directly into the text box, or select them from the input history via the drop-down list (up to 10 items).
Searches are case-insensitive, and only complete matches are retrieved.

Examples 1. main.c#40
2. main.c
3. main

(b) [Function Name]

This item is only enabled if [The execution part is retrieved specifying the function] is selected.
You can either type the function name to be find directly into the text box, or select it from the input history via the drop-down list (up to 10 items).
Searches are case-insensitive, and only complete matches are retrieved.

(c) [Variable Name]

This item is only enabled if [The execution part is retrieved specifying the global variable] is selected.
You can either type the variable name to be find directly into the text box, or select it from the input history via the drop-down list (up to 10 items).
Searches are case-insensitive, and only complete matches are retrieved.

(d) [Kind]

This item is only enabled if [The execution part is retrieved specifying the global variable] is selected.
Select the access type ([Reference/Substitution], [Reference], or [Substitution]) from the drop-down list.

(e) [Value]

This item is only enabled if [The execution part is retrieved specifying the global variable] is selected.
Specify the accessed variable value in hexadecimal number.
You can either type a variable value directly into the text box, or select one from the input history via the drop-down list (up to 10 items).
The variable value can also be specified as a range. In this case, specify a range by specifying variable values in both the left and right text boxes.
If the right-hand text box is blank, then access locations with the fixed variable values specified in the left-hand text box will be searched for.

(3) [Search range] area

(a) [Number]

Specify the range within the trace data to search via the number displayed in the [Number] area of the Trace panel.

Specify the starting number in the left text box, and the ending number in the right text box ("0" to "last number" are specified by default).

You can either type the numbers directly into the text boxes (in base-10 format), or select them from the input history via the drop-down lists (up to 10 items).

If the left-hand text box is left blank, it is treated as if "0" were specified.

If the right-hand text box is left blank, it is treated as if the last number were specified.

Scroll Range Settings dialog box

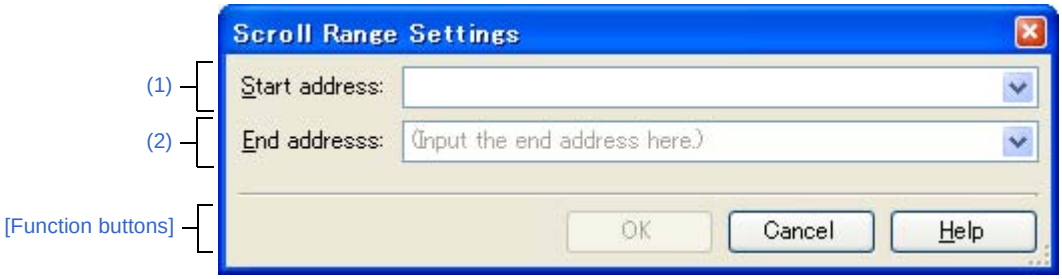
This dialog box is used to set the scroll range of the vertical scroll bar on the Memory panel/Disassemble panel.

By setting the appropriate range, it is possible to improve the operability of a mouse (e.g. dragging) because the size of the vertical scroll bar on the panel is changed suitably.

Caution **After setting a scroll range via this dialog box, the scroll range is not updated automatically even if the address evaluated by the address expression is changed because of such as a line assembly.**

Remark It is possible to move outside the scroll range by using the [Page Up]/[Page Down]/[Up]/[Down] key, a button at either end of the scroll bar or a menu item related to the jump function.

Figure A-60. Scroll Range Setting Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Memory panel, click the button from [View] on the toolbar.
- On the Memory panel, select the [View] menu >> [Settings Scroll Range...] from the context menu.
- On the Disassemble panel, click the button from [View] on the toolbar.
- On the Disassemble panel, select the [View] menu >> [Settings Scroll Range...] from the context menu.

[Description of each area]

(1) [Start address] area

Specify the start address of the range of scrolling.

You can either type an address expression directly into the text box (up to 1024 characters), or select it from the input history via the drop-down list (up to 10 items).

Note that the setting of the scroll range is not performed if "All" is selected in the drop-down list at this time (the scroll range is not limited).

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

(2) [End address] area

Specify the end address of the range of scrolling.

You can either type an address expression directly into the text box (up to 1024 characters), or select it from the input history via the drop-down list (up to 10 items).
Note that this area becomes invalid if [Start address] is specified with [All].

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

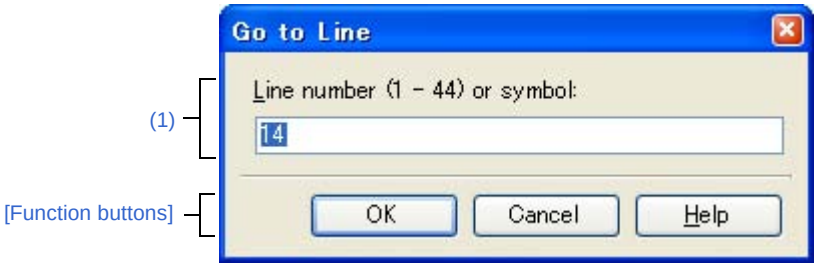
[Function buttons]

Button	Function
OK	Sets the specified scroll range for the target panel. Moves the caret to the start address, from the beginning of the area displayed in the target panel.
Cancel	Cancels the jump and closes this dialog box.
Help	Displays the help for this dialog box.

Go to Line dialog box

This dialog box is used to move the caret to a specified line number, symbol, or address.

Figure A-61. Go to Line Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- Focus the Editor panel, and then select [Go to…] from the [Edit] menu.
- On the Editor panel, select [Go to…] from the context menu.

[Description of each area]

(1) [Line number (*valid line range*) or symbol] area

Directly enter the line number (decimal number), symbol name[Note 1] or address[Note 2] to which you want to move the caret.

"(*valid line range*)" shows the range of valid lines in the current file.

By default, the number of the line where the caret is currently located in the Editor panel is displayed.

Notes 1. Note the following, when specifying a symbol name:
- Either a function name or a variable name can be specified as a symbol name.
- Set the build tool's Property panel so that the cross reference information is output ([Common Options] tab] >> [Output File Type and Path] category >> [Output cross reference information] property >> [Yes(-Xcref)]), and then run and complete a build.

2. Note the following, when specifying an address:
- Enter a hexadecimal number with prefix "0x" or "0X" added (a decimal number is handled as a line number).
- Run and complete a build.

[Function buttons]

Button	Function
OK	Places the caret at the start of the specified source line.
Cancel	Cancels the jump and closes this dialog box.
Help	Displays the help for this dialog box.

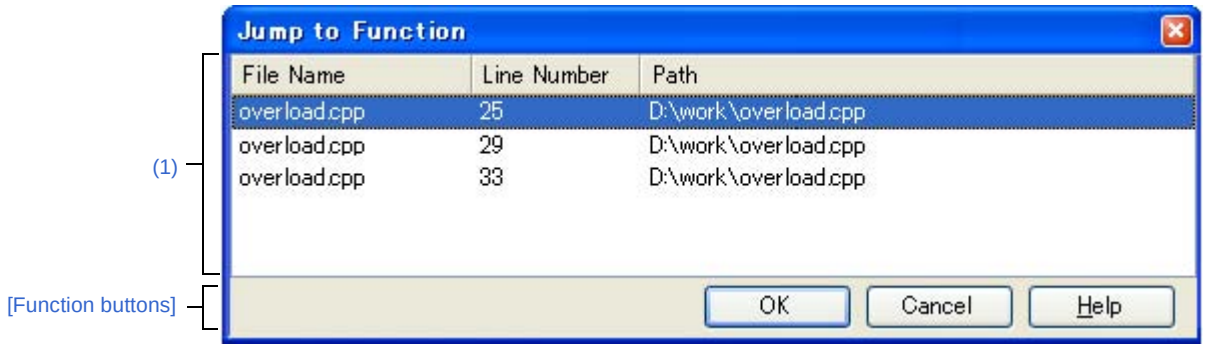
Jump to Function dialog box

This dialog box is used to select a function to be jumped if there are some functions with the same names when a program jumps to the function specified on the Editor panel.

Cautions 1. This dialog box is displayed only when there are some functions with the same names and also the build tool's Property panel is set so that the cross reference information is output ([Common Options] tab] >> [Output File Type and Path] category >> [Output cross reference information] property >> [Yes(-Xcref)]).

2. This dialog box targets only files that have been registered in the project.

Figure A-62. Jump to Function Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Editor panel, select [Jump to Function] from the context menu.

[Description of each area]

(1) Candidates in the jump destination display area

This area displays a list of candidates in the jump destination.
Candidates are displayed in the alphabetical order of the names of [File]. If candidates are included in the same file, they are displayed in the order of line numbers.

(a) [File Name]

Displays the name of the file (without any path) in which the function is defined.

(b) [Line Number]

Displays the number of the line to which the function is defined.

(c) [Path]

Displays the path of the file in which the function is defined.

[Function buttons]

Button	Function
OK	Jumps to the line that defines the target function after selecting the line in Candidates in the jump destination display area and clicking this button.
Cancel	Cancels the jump and closes this dialog box.
Help	Displays the help for this dialog box.

Go to the Location dialog box

This dialog box is used to move the caret to a specified position.

Figure A-63. Go to the Location Dialog Box

This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- Focus the Disassemble panel, and then select [Go to…] from the [Edit] menu.
- Focus the IOR panel, and then select [Go to…] from the [Edit] menu.
- On the Disassemble panel, select [Go to…] from the context menu.
- On the IOR panel, select [Go to…] from the context menu.

[Description of each area]

(1) [Address/Symbol], or [IOR] area

Specify the location to which the caret jumps.

You can either type a location directly into the text box (up to 1024 characters), or select one from the input history via the drop-down list (up to 10 items).

The data to specify various depending on the target panel, as follows:

Target Panel	Data Specified
Disassemble panel	Address expression
IOR panel	I/O register name

Remark If this dialog box is opened from the Disassemble panel, a symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in this text box (see "2.18.2 Symbol name completion function").

[Function buttons]

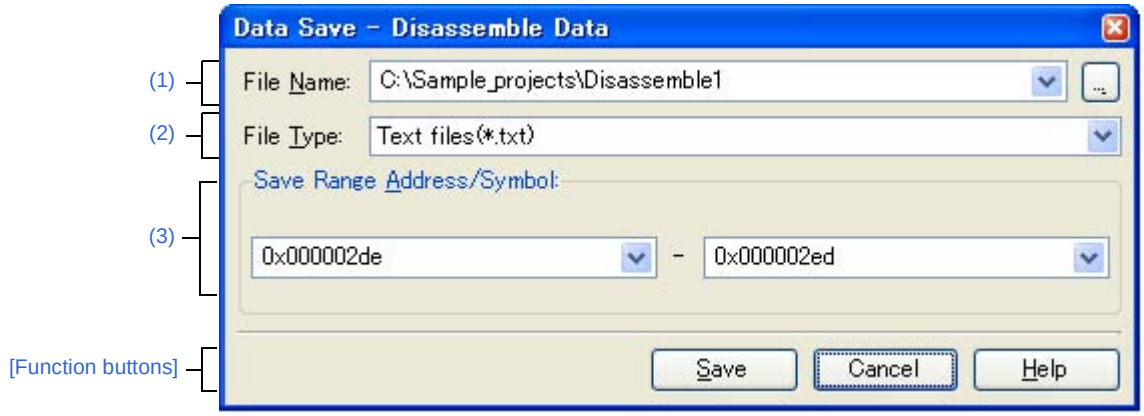
Button	Function
OK	Moves the caret to the specified location, from the beginning of the area displayed in the target panel.
Cancel	Cancels the jump and closes this dialog box.
Help	Displays the help for this dialog box.

Data Save dialog box

This dialog box is used to save data displayed in the Disassemble panel, Memory panel, or Trace panel, and save uploaded data (see "2.5.3 Execute uploading").

This dialog box appears only when connected to the debug tool.

Figure A-64. Data Save Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- With the Disassemble panel in focus, select [Save Disassemble Data As…] form the [File] menu.
- With the Memory panel in focus, select [Save Memory Data As…] form the [File] menu.
- With the Trace panel in focus, select [Save Trace Data As…] form the [File] menu.
- From the [Debug] menu, select [Upload…].

[Description of each area]

(1) [File Name] area

Specify the name of the file to save.

You can either type a filename directly into the text box (up to 259 characters), or select one from the input history via the drop-down list (up to 10 items).

You can also specify the file by clicking the […] button, and selecting a file via the Select Data Save File dialog box.

(2) [File Type] area

Select the format in which to save the file from the following drop-down list.

The available file formats will differ as follows depending on the type of data being saved.

(a) When saving the displayed content of a panel

Text files (*.txt)	Text format (default)
CSV (Comma-Separated Variables) (*.csv)	CSV format[Note]

Note The data is saved with entries separated by commas (,).
If the data contains commas, each entry is surrounded by double quotes ("") in order to avoid illegal formatting.

(b) When saving upload data

Intel HEX format (*.hex)	Intel HEX file format
Motorola S-record (*.mot)	Motorola S-record file format
Binary data (*.bin)	Binary file format

Remark See "2.5.3 Execute uploading" for details on uploading.

(3) [Save Range *xxx*] area

Specify the range of data to save.
You can either type ranges directly into the text boxes, or select them from the input history via the drop-down lists (up to 10 items).
The method of specifying the ranges will differ as follows depending on the type of data to be saved.

Type of Data	Description
Disassemble panel	Specify the range of addresses to save via the start and end addresses. Ranges can be entered as base-16 numbers or as address expressions. When a range is selected in the panel, that range is specified by default. When there is no selection, then the range currently visible in the panel is specified.
Memory panel	Specify the range of memory to save via the start and end addresses. Ranges can be entered as base-16 numbers or as address expressions. When a range is selected in the panel, that range is specified by default. When there is no selection, then the range currently visible in the panel is specified.
Trace panel	- Specifying a range to save Specify the trace range to save via the start and end trace numbers[Note]. Ranges can only be entered as base-10 numbers. - Saving all trace data From the drop-down list to the left, select [All Trace Data]. The text box to the right is disabled. All currently acquired trace data will be saved. The range currently visible in the panel is specified by default.
Upload data	Specify the range of memory to save via the start and end addresses. Ranges can be entered as base-16 numbers or as address expressions.

Note These are the numbers shown in the [Number] area of the Trace panel.

Remark A symbol name at the current caret position can be complemented by pressing the [Ctrl] + [Space] key in each text box (see "2.18.2 Symbol name completion function").

[Function buttons]

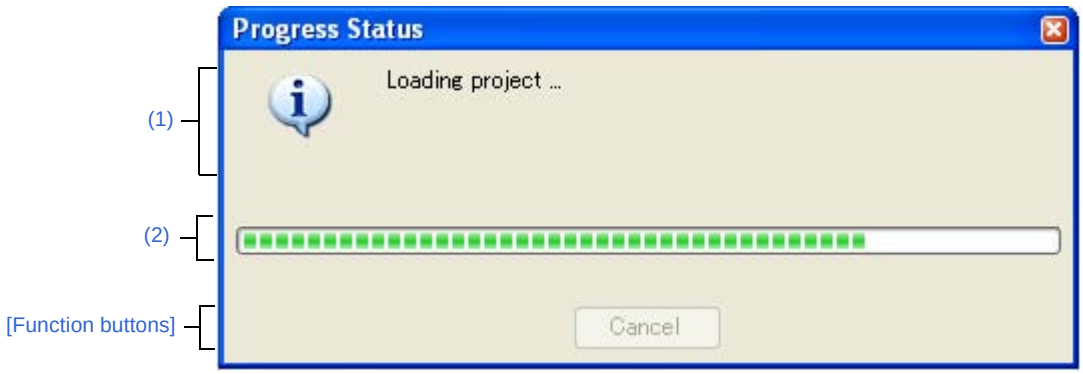
Button	Function
Save	Saves the data to a file with the specified filename, in the specified format.
Cancel	Cancels the save and closes this dialog box.

Button	Function
Help	Displays the help for this dialog box.

Progress Status dialog box

This dialog box is used to display the progress of long processes.
This dialog box closes automatically when the currently executing process completes.

Figure A-65. Progress Status Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- This dialog box appears automatically when a message is displayed during a long process.

[Description of each area]

(1) Message display area

Displays messages during processing (cannot be edited).

(2) Progress bar

The amount of progress made toward completing the current progress is indicated by the length of the bar.
The dialog box will automatically close when the progress reaches 100% (the length of the bar reaches the right end).

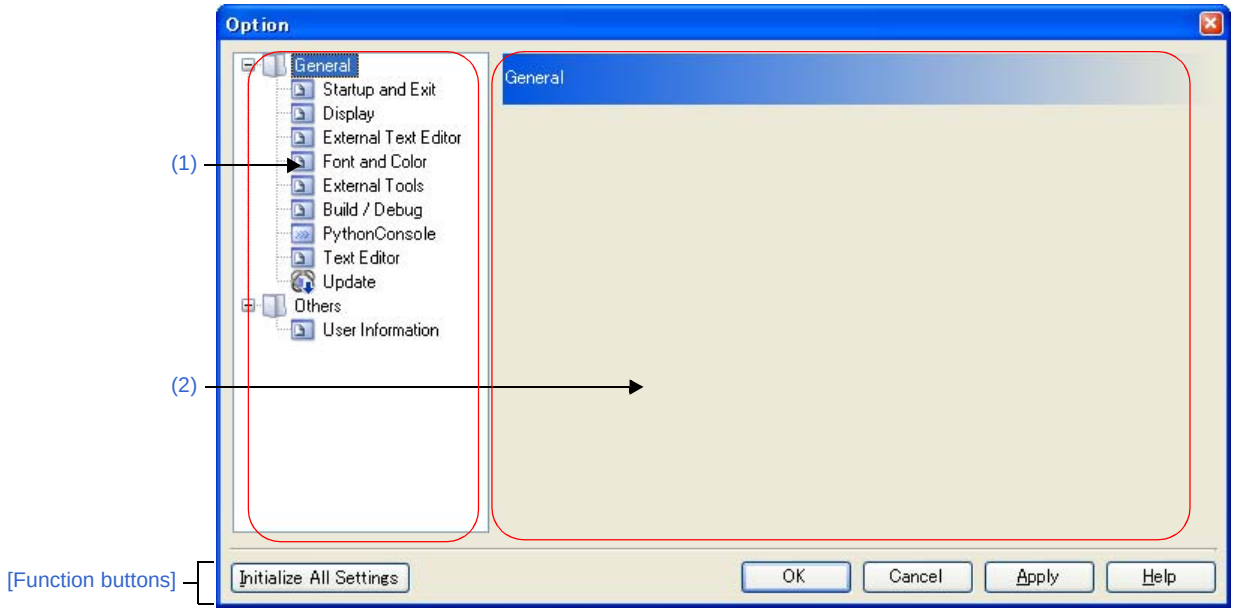
[Function buttons]

Button	Function
Cancel	Cancels the currently executing process, and closes this dialog box. Note that this button will be disabled if the currently executing process cannot be interrupted.

Option dialog box

This dialog box is used to configure the CubeSuite+ environment.
All settings made via this dialog box are saved as preferences for the current user.

Figure A-66. Option Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Options...].

[Description of each area]

(1) Category selection area

Select the items to configure from the following categories.

Category	Description
[General - Startup and Exit] category	Configure startup and shutdown.
[General - Display] category	Configure messages from the application.
[General - External Text Editor] category	Configure the external text editor.
[General - Font and Color] category	Configure the fonts and colors shown on each panel.
[General - External Tools] category	Configure the startup of external tools.
[General - Build/Debug] category	Configure building and debugging.
[General - PythonConsole] category	Configure the phthon console.
[General - Text Editor] category	Configure the text editor.

Category	Description
[General - Update] category	Configure updating.
[Others - User Information] category	Configure user information.

Remark See "CubeSuite+ Integrated Development Environment User's Manual: Start" for details on the categories other than [General - Font and color]/[General - Build/Debug].

(2) Setting area

This area is used to configure the various options for the selected category.
For details about configuration for a particular category, see the section for the category in question.

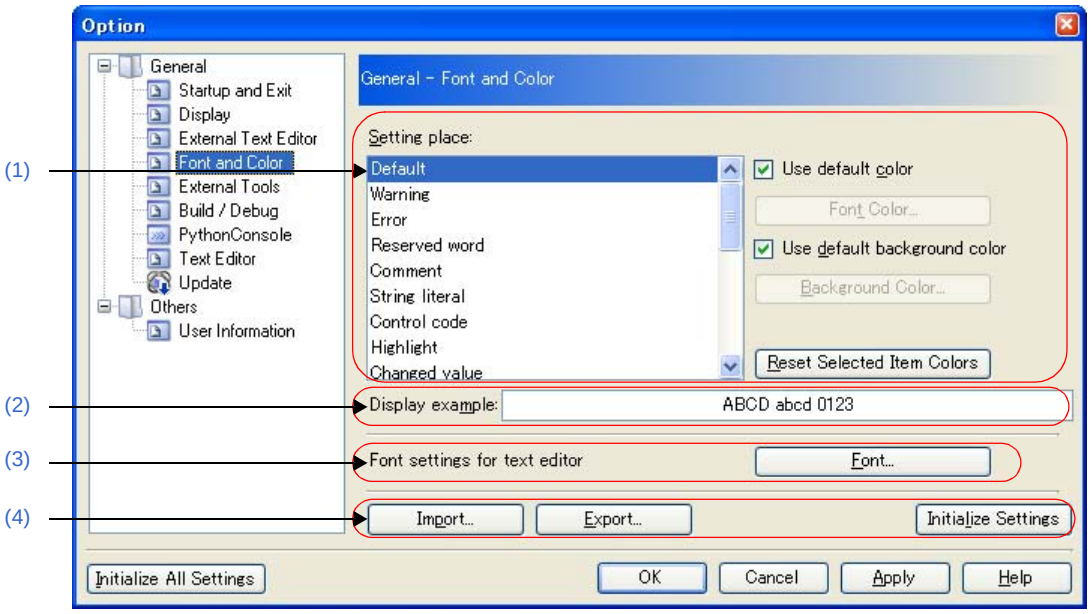
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help for this dialog box.

[General - Font and Color] category

This tab is used to configure general settings relating to fonts and colors on each panel.

Figure A-67. Option Dialog Box ([General - Font and Color] Category)



The following items are explained here.

- [How to open]
- [Description of each area]

[How to open]

- From the [Tool] menu, select [Options...].

[Description of each area]

(1) Color options area

Use this area to configure the colors.

(a) [Setting place] area

Select a location from the list for which the color will be specified.

The relationships between the list items and default color settings are as follows:

Item	Example			Description
Default[Note]	AaBbCc	Font color	Black	Standard display colors on all windows and panels
		Background color	White	
Warning	AaBbCc	Font color	Blue	Display colors of warning messages on the Output panel, as well as display colors for file names with "warnings included" on the Project Tree panel
		Background color	Default color	

Item	Example			Description
Error	AaBbCc	Font color	Red	Display colors of warning messages on the Output panel, as well as display colors for file names with "errors included" on the Project Tree panel
		Background color	Whitesmoke	
Reserved word	AaBbCc	Font color	Maroon	Display colors of reserved words on the Editor panel for compilers/assemblers used
		Background color	Default color	
Comment	AaBbCc	Font color	Green	Display colors of comment parts (for C source files, "/* to */ ") on the Editor panel
		Background color	Default color	
String literal	AaBbCc	Font color	Gray	Display colors of string literals on the Editor panel
		Background color	Default color	
Control code	AaBbCc	Font color	Darkcyan	Display colors of control characters on the Output panel
		Background color	Default color	
Highlight	AaBbCc	Font color	White	Display colors of highlighted spots in plug-in products, etc.
		Background color	Mediumvioletred	
Changed value	AaBbCc	Font color	Sienna	Display colors on the Memory panel, CPU Register panel, Local Variables panel, IOR panel and Watch panel of spots whose values have been changed by program execution
		Background color	Lightyellow	
Edit value	AaBbCc	Font color	Blue	Display colors on the Memory panel, CPU Register panel, Local Variables panel, IOR panel and Watch panel of spots whose values have been forcibly changed by user
		Background color	Default color	
Current PC	AaBbCc	Font color	Black	Display colors on the Editor panel and Disassemble panel of a line where the current PC position exists
		Background color	Gold	
Breakpoint	AaBbCc	Font color	Black	Display colors on the Editor panel and Disassemble panel of a line where breakpoints are set
		Background color	Lightsalmon	
Update periodic	AaBbCc	Font color	Deeppink	Display colors on the Memory panel and Watch panel of areas whose display is set to be updated in real time
		Background color	Default color	
Read or fetch	AaBbCc	Font color	Default color	Display colors on the Memory panel and Trace panel of spots that have been read or fetched
		Background color	Palegreen	
Write	AaBbCc	Font color	Default color	Display colors on the Memory panel and Trace panel of spots that have been written
		Background color	Orange	

Item	Example			Description
Read and write	AaBbCc	Font color	Default color	Display colors on the Memory panel and Trace panel of spots that have been read and written
		Background color	Paletur-quoise	
Lost	AaBbCc	Font color	White	Display colors on the Analysis Chart panel of the analyze tool (Program Analyzer) of sections where acquisition of graph data has failed
		Background color	Lightgray	
Coverage 100%	AaBbCc	Font color	Default color	Display colors on the Editor panel and Disassemble panel of lines whose code coverage rates are 100 %
		Background color	Lightgreen	
Coverage 1 - 99%	AaBbCc	Font color	Default color	Display colors on the Editor panel and Disassemble panel of lines whose code coverage rates are 1 to 99 %
		Background color	Lightpink	
Coverage 0%	AaBbCc	Font color	Default color	Display colors on the Editor panel and Disassemble panel of lines whose code coverage rates are 0 % (unexecuted)
		Background color	Lightgray	
Invalid	AaBbCc	Font color	Gray	Display colors on the Memory panel of areas that are not memory-mapped, and of file names that are not actually present on the Project Tree panel
		Background color	Default color	

Note The [Default] text and background colors depends on the Windows settings of the host computer. Here, we use the Windows defaults, which are black text and white background.

(b) [Use default color]

☑	Display items selected via the [Setting place] area using the standard text color.
☐	Display items selected via the [Setting place] area with a user-defined text color. The [Font color...] button is enabled.

(c) [Use default background color]

☑	Display items selected via the [Setting place] area using the standard background color.
☐	Display items selected via the [Setting place] area with a user-defined background color. The [Background Color...] button is enabled.

(d) Buttons

Font Color...	The Edit Colors Dialog Box opens. Specify the text color of the item selected via the [Setting place] area. Note, however, that this button will be disabled if the [Use default color] check box is selected.

Background Color...	The Edit Colors Dialog Box opens. Specify the background color of the item selected via the [Setting place] area. Note, however, that this button will be disabled if the [Use default background color] check box is selected.
Reset Selected Item Colors	Reset the color information for the item selected via the [Setting place] area to the defaults.

Figure A-68. Edit Colors Dialog Box

(2) [Display example] area

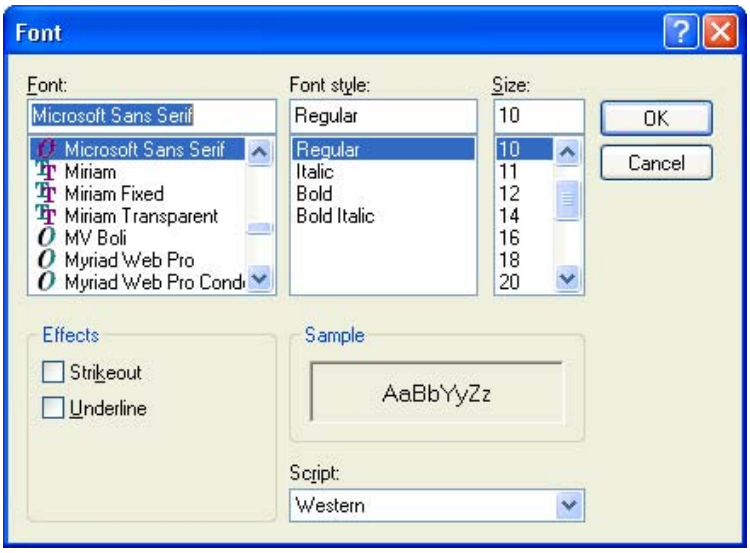
Display sample text using the color and font settings from the Color options area.

By default the string "AaBbCc" is shown, but you can type an arbitrary string directly into the text box.

(3) [Font settings for text editor] area

Click the [Font...] button to open the Font Dialog Box and configure the fonts for your text editor.

Figure A-69. Font Dialog Box



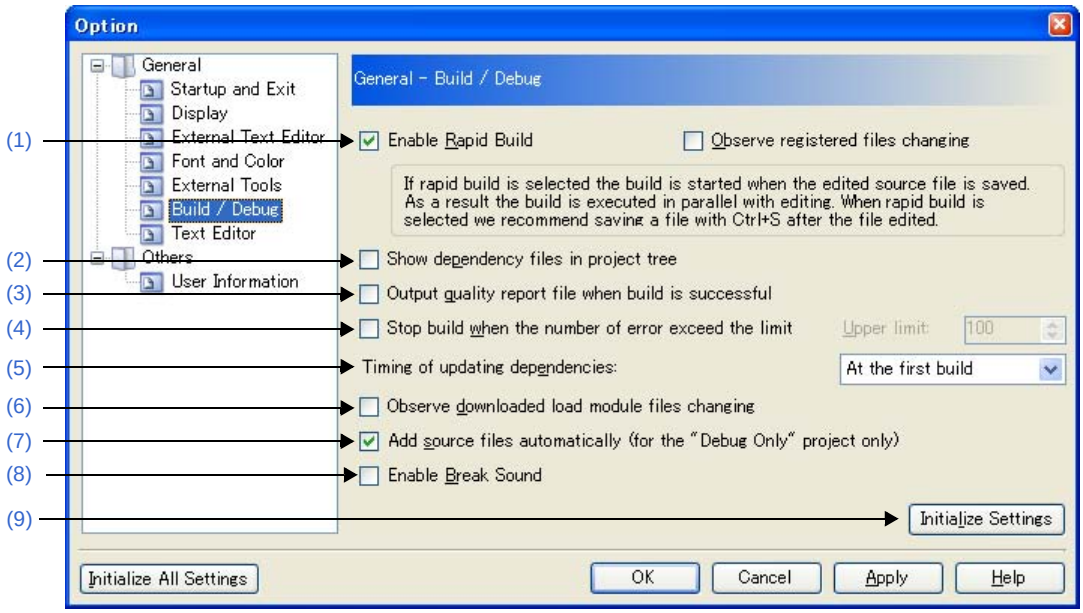
(4) Buttons

Import...	Opens the Open Option Setting File dialog box to reflect the set contents that were saved in a file to this category.
Export...	Opens the Save Option Setting File dialog box to save the set contents of this category to a file.
Initialize Settings	Returns all currently displayed setting to their default values.

[General - Build/Debug] category

This tab is used to configure general setting relating to building and debugging.

Figure A-70. Option Dialog Box ([General - Build/Debug] Category)



The following items are explained here.

- [How to open]
- [Description of each area]

[How to open]

- From the [Tool] menu, select [Options...].

[Description of each area]

(1) [Enable Rapid Build]

☑	Enable the rapid build Note feature (default).
☐	Do not use the rapid build feature.

Note This feature automatically begins a build when the source file being edited is saved.
Enabling this feature makes it possible to perform builds while editing source files.
If this feature is used, we recommend saving frequently after editing source files.

(a) [Observe registered files changing]

☑	Start a rapid build when a source file registered in the project is edited or saved by an external text editor or the like.
☐	Do not start a rapid build when a source file registered in the project is edited or saved by an external text editor or the like (default).

Remark This item is only enabled if the [Enable Rapid Build] check box is selected.

Cautions 1. The rapid build will not finish if this item is selected, and the files to be built have been registered for automatic editing or overwriting (e.g. by commands executed before or after the build). If the rapid build does not finish, unselect this item, and stop the rapid build.

2. If this item is selected, a file that is registered in the project but does not exist (a file grayed out) will not be observed even if it is registered again by the Explorer, etc. To observe the file, reload the project file, or select this item again after unselecting this item and closing this dialog box.

(2) [Show dependency files in project tree]

☑	Displays the group of files on which the source file depends as a project tree.
☐	Does not display the group of files on which the source file depends as a project tree (default).

(3) [Output quality report file when build is successful]

☑	Outputs a quality report file if the build is successful.
☐	Does not output a quality report file if the build is successful (default).

Remarks 1. The quality report file is not output when a rapid build is executed, a debug-dedicated project is built, and compiling or assembling is executed in file units.

2. The following information item is output to the quality report file.
- Time and date on which the file is created
- Log of the build results
- Information on the command file which is used during building
- Information on the detailed version of this product or the current project

3. The quality report file is output with the file name "QuarityReport(*project-name.build-mode-name*).text" to the project folder of each project.
If a file having the same name exists, it will be overwritten.
It is also shown on the project tree, under the Build tool generated files node.

(4) [Stop build when the number of error exceed the limit]

☑	Stops the build if the total number of errors at the build reaches the number specified in [Upper limit].
☐	Does not stop the build even if the total number of errors at the build reaches the number specified in [Upper limit] (default).

(a) [Upper limit]

Specify the upper limit of the number of errors.
Either enter a number between 1 and 10000 directly via the keyboard, or specify a number via the buttons.
The default is 100.

Remark This item is only enabled if the [Stop build when the number of error exceed the limit] check box is selected.

(5) [Timing of updating dependencies]

At the first build	Updates dependencies immediately before executing the first build after opening the project (default).
At every build	Updates dependencies immediately before executing the build.

(6) [Observe downloaded load module files changing]

☑	Watches changes made to the load module files downloaded to the debug tool, so that when changes are made, a message dialog box is displayed for confirmation of whether or not to execute a download.
☐	Does not watch changes made to the load module files downloaded to the debug tool (default).

(7) [Add source files automatically (for the "Debug Only" project only)]

☑	Automatically adds the source files to the project tree when the load-module files are downloaded to the debug tool in the debug-dedicated project (default).
☐	Does not automatically add the source files to the project tree when the load-module files are downloaded to the debug tool in the debug-dedicated project

Caution This function is valid only when the load module files have been added to the Download files node of the project tree. If the load module files have been added via the [Download File Settings] tab in the Property panel of the debug tool, then the source files will not be added to the project tree.

(8) [Enable Break Sound]

☑	Beeps when the execution of a program is halted due to a break event (Hardware or Software break).
☐	Does not beep when the execution of a program is halted due to a break event (Hardware or Software break) (default).

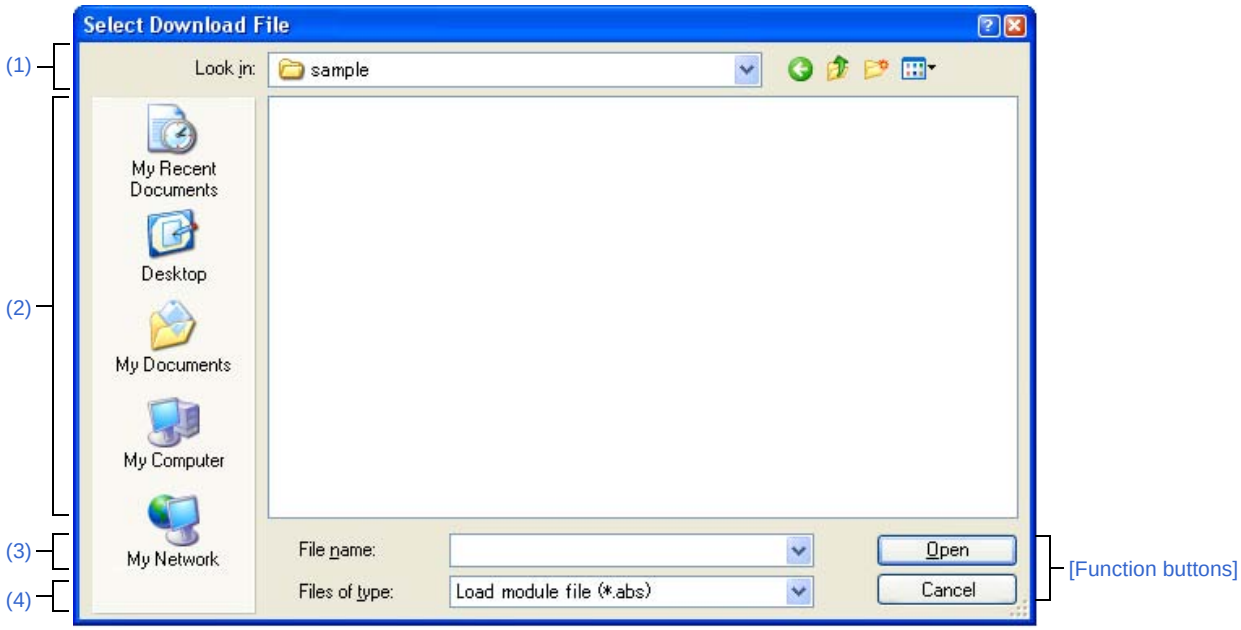
(9) Buttons

Initialize Settings	Returns all currently displayed setting to their default values.

Select Download File dialog box

This dialog box is used to select a downloaded file.

Figure A-71. Select Download File Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Download file property] area in the Download Files dialog box, click the [...] button on the [File] property.

[Description of each area]

(1) [Look in] area

Select the folder which contains the file you want to download.

(2) List of files area

This area displays a list of files matching the conditions selected in the [Look in] and [Files of type] areas.

(3) [File name] area

Specify the name of a file you want to download.

(4) [Files of type] area

Select the type of a file to download (file type).

Load module file (*.abs)	Load module file format (default)
Hex file (*.hex)	Intel HEX file format
S record file (*.mot)	Motorola S-record file format
Binary file (*.bin)	Binary file format
All files (*.*)	All file formats

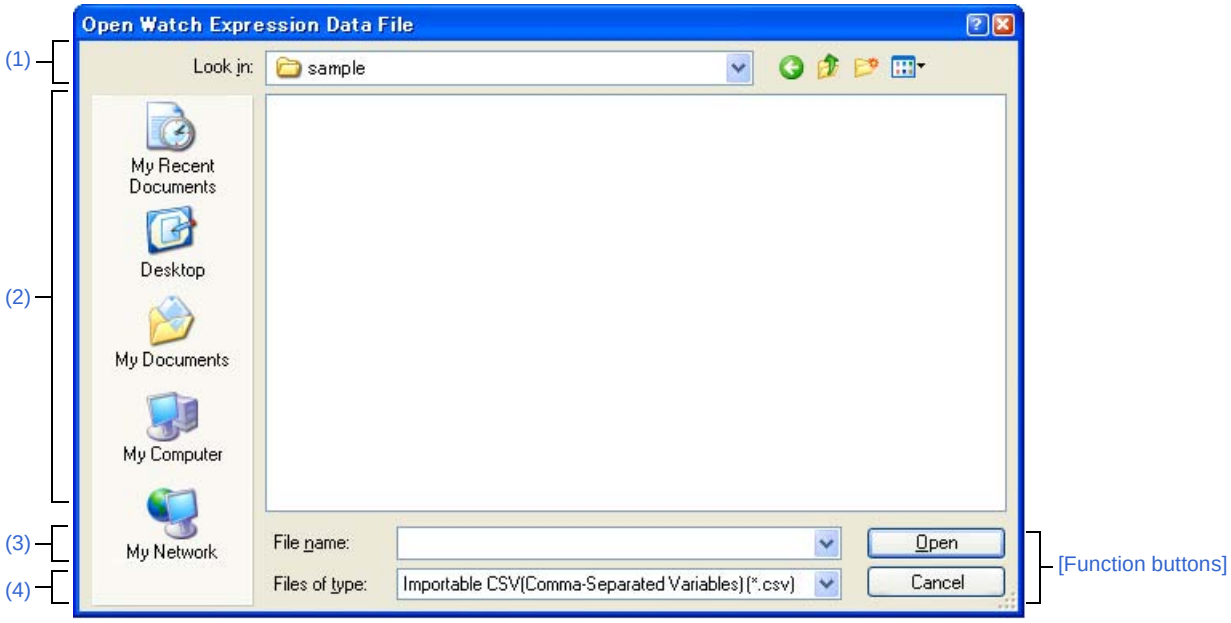
[Function buttons]

Button	Function
Open	Adds the specified file to the Download Files dialog box.
Cancel	Closes the dialog box.

Open Watch Expression Data File dialog box

This dialog box is used to select a file that imports watch-expressions to the Watch panel.

Figure A-72. Open Watch Expression Data File Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the Watch panel, select [Import Watch Expression...] from the context menu.

[Description of each area]

(1) [Look in] area

Select the folder which contains the file you want to import.

(2) List of files area

This area displays a list of files matching the conditions selected in the [Look in] and [Files of type] areas.

(3) [File name] area

Specify the name of a file you want to import.

(4) [Files of type] area

The following type of the file (file type) is shown.

Importable CSV(Comma-Separated Variables) (*.csv)	CSV format to enable import

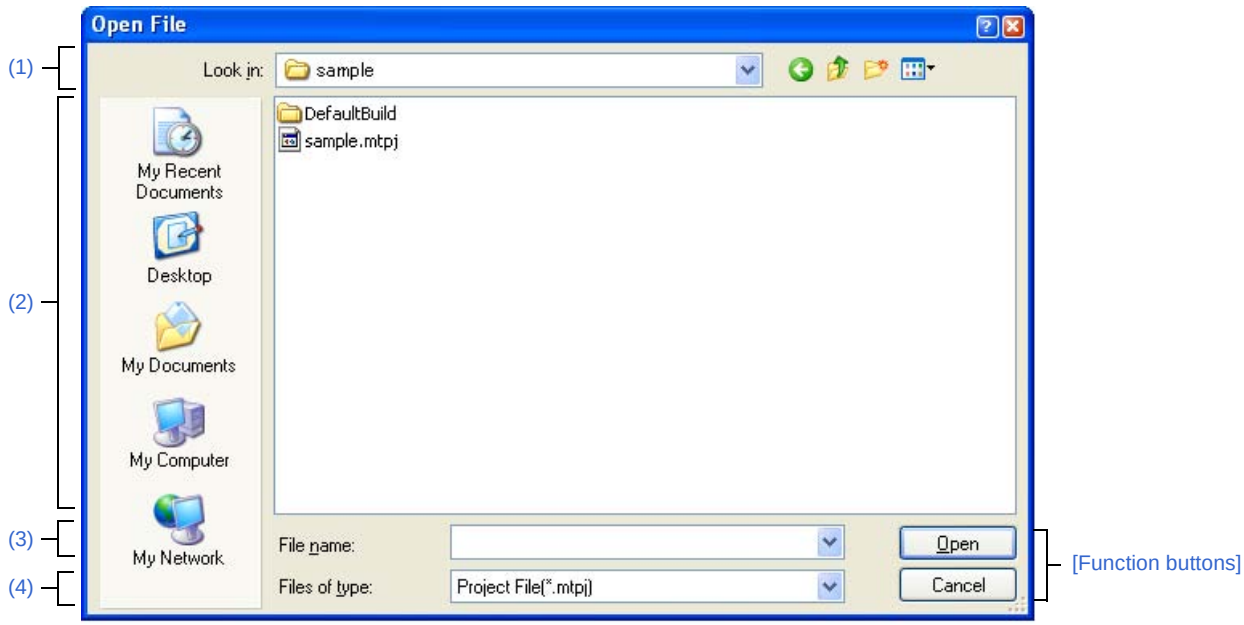
[Function buttons]

Button	Function
Open	Imports the specified file to the Watch panel.
Cancel	Closes the dialog box.

Open File dialog box

This dialog box is used to open a file.

Figure A-73. Open File Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [File] menu, select [Open File...] or [Open with Encoding...].

[Description of each area]

(1) [Look in] area

Select the folder which contains the file you want to open.

When you first open this dialog box, the folder is set to "C:\Documents and Settings *user-name*\My Documents".

The second and subsequent times, this defaults to the last folder that was selected.

(2) List of files area

This area displays a list of files matching the conditions selected in the [Look in] and [Files of type] areas.

(3) [File name] area

Specify the name of a file you want to open.

(4) [Files of type] area

Select the type of the file you want to open (file type).

All files (*.*)	All formats

Project File(*.mtpj)	Project file
Project File for e2 studio (*.rcpc)	Project file for e^2 studio
Project File for CubeSuite (*.cspj)	Project file for CubeSuite
Workspace File for HEW (*.hws)	Workspace file for HEW
Project File for HEW (*.hwp)	Project file for HEW
Workspace File for PM+ (*.prw)	Workspace file for PM+
Project File for PM+ (*.prj)	Project file for PM+
C source file (*.c)	C language source file
Header file (*.h; *.inc)	Header file
Assembly source file (*.asm; *.s; *fsy)	Assembly source file
Link map file (*.map; *.lbp)	Link map file
Stack information file (*.sni)	Stack information file
Intel HEX file (.hex)	Intel HEX file
Motorola S-record file (*.mot)	Motorola S-record file
Text file (*.txt)	Text format

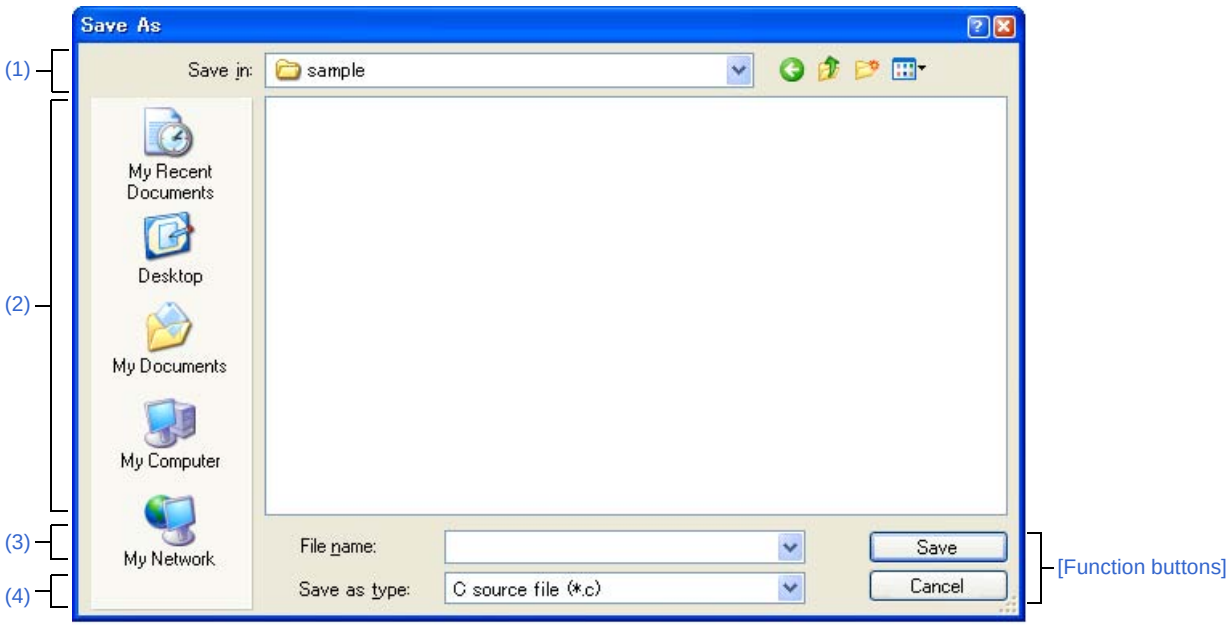
[Function buttons]

Button	Function
Open	- When this dialog box is opened by [Open File...] from the [File] menu Opens the specified file. - When this dialog box is opened by [Open with Encoding...] from the [File] menu Opens the Encoding dialog box.
Cancel	Closes this dialog box.

Save As dialog box

This dialog box is used to save the contents of the panel into a specified file.

Figure A-74. Save As Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- With the Editor panel in focus, select [Save *file name* As...] from the [File] menu.
- With the CPU Register panel in focus, select [Save CPU Register Data As...] from the [File] menu.
- With the Watch panel in focus, select [Save Watch Data As...] from the [File] menu.
- With the IOR panel in focus, select [Save IOR Data As...]. from the [File] menu.
- With the Call Stack panel in focus, select [Save Call Stack Data As...] from the [File] menu.
- With the Local Variables panel in focus, select [Save Local Variable Data As...] from the [File] menu.
- With the Output panel in focus, select [Save *tab name* As...] from the [File] menu.

[Description of each area]

(1) [Save in] area

Select the folder in which you want to save the file, from the drop-down list.

(2) List of files area

This area displays a list of files matching the conditions selected in the [Save in] and [Save as type] areas.

(3) [File name] area

Specify a file name under which you want to save.

(4) [Save as type] area

(a) In the Editor panel

The following file types are displayed depending on the file being edited.

Remark The following strings are displayed only for the files registered in the project tree.

C source file (*.c)	C language source file
Header file (*.h;*.inc)	Header file
Assembly source file (*.asm; *.s; *fsy)	Assembly source file
Link order specification file (*.mtls)	Link order specification file
Link map file (*.map; *.lbp)	Link map file
Intel HEX file (*.hex)	Intel HEX file
Assemble list file (*.lst)	Assemble list file
Stack information file (*.sni)	Stack information file
Motorola S-record file (*.mot)	Motorola S-record file
Text file (*.txt)	Text format
CSV (Comma-Separated Variables)(*.csv)Note 1	CSV formatNote 2

Notes 1. This item appears only when this dialog box was opened from the Editor panel in the Mixed display mode.

2. The data is saved with entries separated by commas (,).
If the data contains commas, each entry is surrounded by double quotes (" ") in order to avoid illegal formatting. Moreover, "0x" is added to the address and code information.

(b) In the CPU Register panel/Watch panel/IOR panel/Call Stack panel/Local Variables panel

The following file types are displayed.
Select the format in which to save the file from the drop-down list.

Text file (*.txt)	Text format (default)
CSV (Comma-Separated Variables)(*.csv)	CSV formatNote 1
Importable CSV (Comma-Separated Variables)(*.csv)Note 2	CSV formatNote 1 to enable import

Notes 1. The data is saved with entries separated by commas (,).
If the data contains commas, each entry is surrounded by double quotes (" ") in order to avoid illegal formatting.

2. This item appears only when this dialog box was opened from the Watch panel.

(c) In the Output panel

The following file types are displayed.
The contents can be saved only in text format.

Text file (*.txt)	Text format (default)

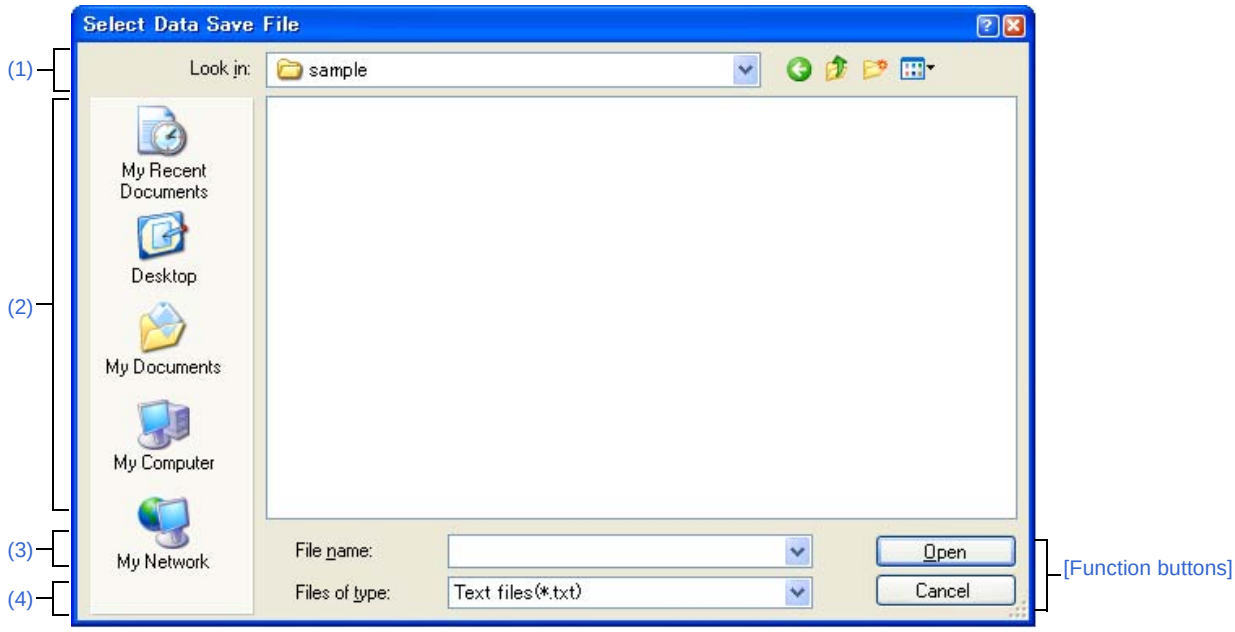
[Function buttons]

Button	Function
Save	Saves the file with the specified name.
Cancel	Closes the dialog box.

Select Data Save File dialog box

This dialog box is used to select a file in which to save the data.

Figure A-75. Select Data Save File Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [File Name] area in the Data Save dialog box, click the [...] button.

[Description of each area]

(1) [Look in] area

Select the folder which contains the file you want to save.

(2) List of files area

This area displays a list of files matching the conditions selected in the [Look in] and [Files of type] areas.

(3) [File name] area

Specify the name of a file you want to save.

(4) [Files of type] area

Select the type of the file (file type).

The available file formats will differ as follows depending on the type of data being saved.

(a) When saving the displayed content of a panel

Text files (*.txt)	Text format (default)
CSV (Comma-Separated Variables)(*.csv)	CSV format[Note]

Note The data is saved with entries separated by commas (,).
If the data contains commas, each entry is surrounded by double quotes (" ") in order to avoid illegal formatting.

(b) When saving upload data

Intel Hex format (*.hex)	Intel HEX file format
Motorola S-record (*.mot)	Motorola S-record file format
Binary data (*.bin)	Binary file format

Remark See "2.5.3 Execute uploading" for details on uploading.

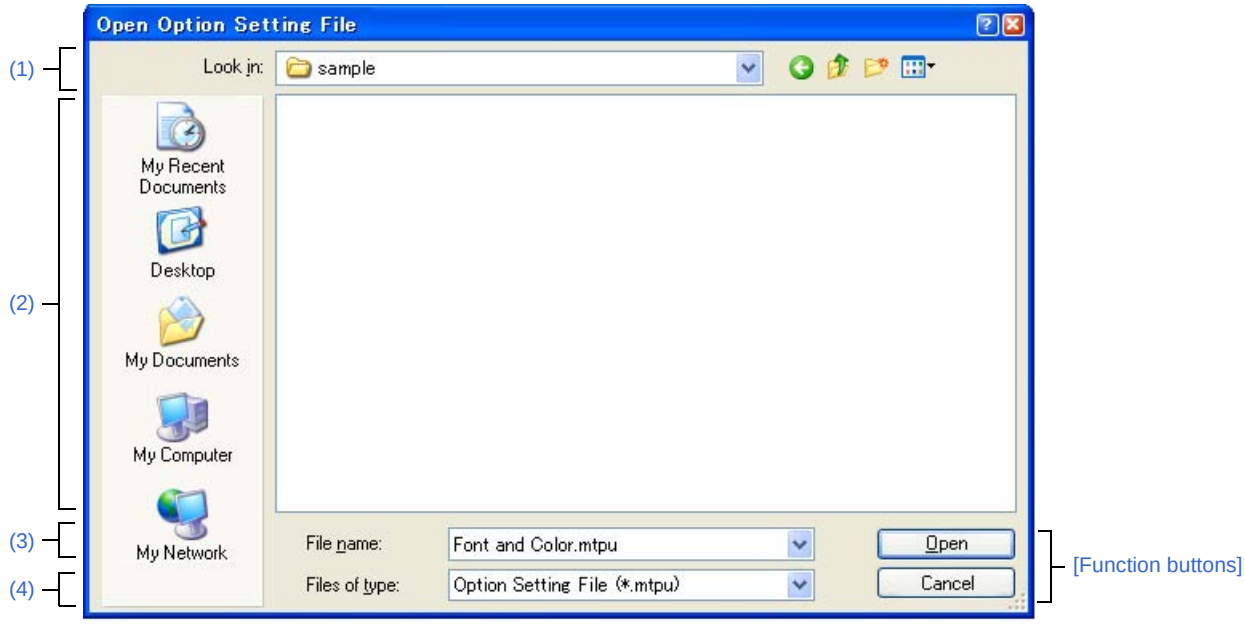
[Function buttons]

Button	Function
Open	Specifies the specified file in the Data Save dialog box.
Cancel	Closes the dialog box.

Open Option Setting File dialog box

This dialog box is used to select a option setting file to import to the [General - Font and Color] category of the Option dialog box.

Figure A-76. Open Option Setting File Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [General - Font and Color] category of the Option dialog box, click the [Import...] button.

[Description of each area]

(1) [Look in] area

Select the folder which contains the option setting file you want to open.
When you first open this dialog box, the folder is set to "C:\Documents and Settings \user-name\My Documents".
The second and subsequent times, this defaults to the last folder that was selected.

(2) List of files area

This area displays a list of files matching the conditions selected in the [Look in] and [Files of type] areas.

(3) [File name] area

Specify the name of a option setting file you want to open.

(4) [Files of type] area

The following type of the file (file type) is shown.

Option Setting File (*.mtpu)	Option setting file

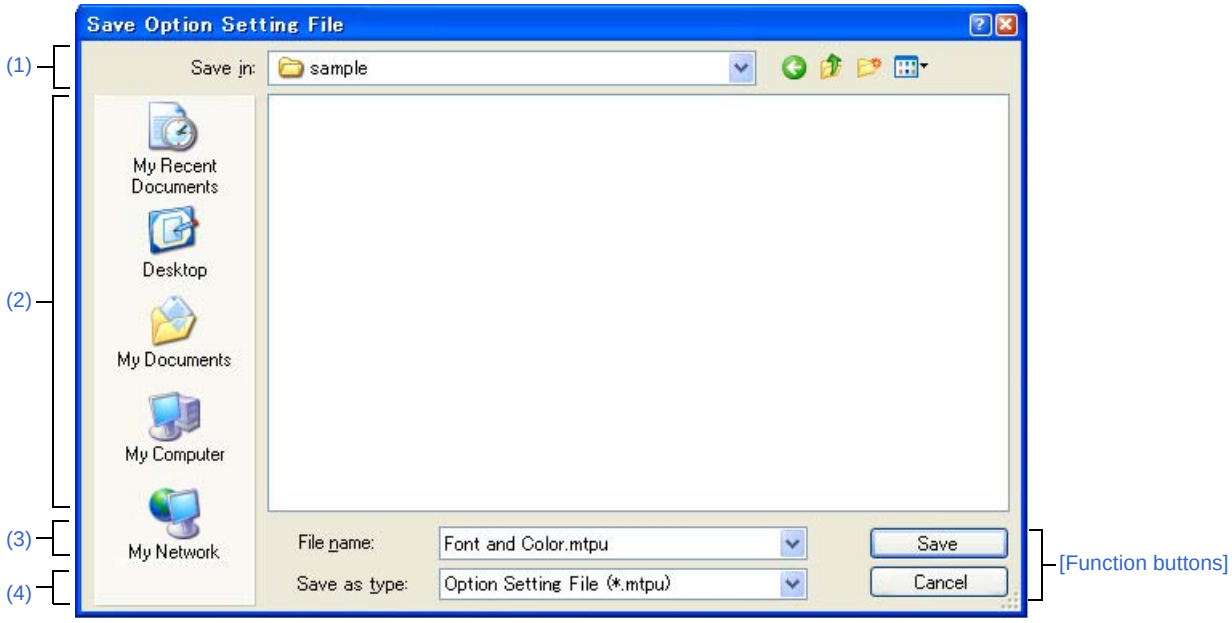
[Function buttons]

Button	Function
Open	Imports the specified file to the [General - Font and Color] category of the Option dialog box.
Cancel	Closes this dialog box.

Save Option Setting File dialog box

This dialog box is used to save the set contents of the [General - Font and Color] category of the Option dialog box to a option setting file.

Figure A-77. Save Option Setting File Dialog Box



This section describes the following.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [General - Font and Color] category of the Option dialog box, click the [Export...] button.

[Description of each area]

(1) [Save in] area

Select the folder in which you want to save the option setting file.

When you first open this dialog box, the folder is set to "C:\Documents and Settings *user-name*\My Documents".

The second and subsequent times, this defaults to the last folder that was selected.

(2) List of files area

This area displays a list of files matching the conditions selected in the [Save in] and [Save as type] areas.

(3) [File name] area

Specify the name of a option setting file under which you want to save.

(4) [Save as type] area

The following type of the file (file type) is shown.

Option Setting File (*.mtpu)	Option setting file

[Function buttons]

Button	Function
Save	Saves the option setting file with the specified name.
Cancel	Closes the dialog box.

APPENDIX B INDEX

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Sep 01, 2013	-	First Edition issued

CubeSuite+ V2.01.00 User's Manual:
RH850 Debug

Publication Date: Rev.1.00 Sep 01, 2013

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.3

CubeSuite+ V2.01.00

Renesas Electronics Corporation

R20UT2685EJ0100