RENESAS

CubeSuite+ V2.01.00

Release Note

Chapter 1.   Target Devices

RENESAS

The target devices supported by the CubeSuite+ are listed on the Website.
Please see this URL.
CubeSuite+ Product Page:

<div align="center">

http://www.renesas.com/cubesuite+

</div>

Chapter 2.   User's Manuals

Please read the following user's manuals together with this document.

Manual Name	Document Number
CubeSuite+ V2.01.00 Start	R20UT2682EJ0100
CubeSuite+ V1.03.00 78K0 Design	R20UT2138JJ0100
CubeSuite+ V1.03.00 78K0R Design	R20UT2137JJ0100
CubeSuite+ V2.01.00 RL78 Design	R20UT2684JJ0100
CubeSuite+ V1.03.00 V850 Design	R20UT2134JJ0100
CubeSuite+ V2.01.00 RX Design	R20UT2683JJ0100
CubeSuite+ V1.01.00 78K0 Debug	R20UT0731EJ0100
CubeSuite+ V1.01.00 78K0R Debug	R20UT0732EJ0100
CubeSuite+ V2.00.00 RL78 Debug	R20UT2445JJ0100
CubeSuite+ V2.00.00 V850 Debug	R20UT2446EJ0100
CubeSuite+ V2.01.00 RX Debug	R20UT2702EJ0100
CubeSuite+ V2.01.00 RH850 Debug	R20UT2685EJ0100
CubeSuite+ V2.01.00 Analysis	R20UT2686EJ0100
CubeSuite+ V2.01.00 Message	R20UT2687EJ0100

Chapter 3.   Key Points for Selecting Uninstallation Method

There are two ways to uninstall this product.
- Use the integrated uninstaller (uninstalls CubeSuite+)
- Use separate uninstaller (uninstalls this product only)

To use the separate uninstaller, select the following from the Control Panel:
- Add/Remove Programs (Windows XP)
- Programs and Features (Windows Vista / Windows 7 / Windows 8)

Then select "CubeSuite+".

Chapter 4.   Changes

This chapter describes changes from V2.00.00 to V2.01.00

4.1   Improvement of CubeSuite+ functionality

4.1.1   Addition of new microcontrollers
This version newly supports RH850-family microcontrollers. The build tool for the RH850 family is CC-RH.

4.1.2   Addition of support for code generation
This version newly supports code generation for the RX100.

4.1.3   Enhanced menus
The following items were added to the Build and Debug menus.
• Set Link Order of Project
• Rebuild & Download
• Using Debug Tool
• Startup Stack Usage Tracer

4.1.4   Enhanced toolbar
A pull-down menu for selecting the build mode and a button for controlling bookmarks were added.

4.1.5   Enhanced facility to change the target microcontroller (RL78 and 78K0R)
When the target microcontroller is changed, [Debug monitor area start address] of [Link Option] is also changed to the address of the currently selected microcontroller.

4.1.6   Addition of command-line options in CubeSuite+
The following two options related to starting CubeSuite+ from the command line were added.
• /lp: The specified plug-in is loaded before CubeSuite+ starts.
• /lpall: All plug-ins are loaded before CubeSuite+ starts.

4.1.7   Enhanced facility for creating multi-core projects (RH850)
A facility for creating projects for multi-core RH850 products was added to the Start panel.

4.1.8   Enhanced Property panel
A search facility was added to the Property panel.

4.2   Improvements to the editor

4.2.1   Addition of a [Bookmark] menu item
A [Bookmark] menu item was added to the context menu in the editor.
A bookmark toolbar was also added to CubeSuite+.
Information on bookmarks is saved as part of the project information.

4.2.2   Enhanced highlighting
Highlighting of the current line is now supported.
To enable this feature, make an appropriate setting in the text editor category of the Options dialog box.
Emphasizing brackets is also supported.

4.2.3 Enhanced zooming in/out

The display in the editor is also in accord with the zoom ratio selected from the toolbar of CubeSuite+.

4.2.4 Enhanced display of tooltip information

When the mouse cursor is placed on a variable of class, structure, union, or enumeration type, its members are displayed as tooltip information. This feature is only supported for RX-, V850E2-, and RH850-family projects. [RX] [V850E2] [RH850]

4.2.5 Enhanced mixed display of source code

[Save Source Mixed Data As...] was added to the CubeSuite+ menu.
This menu item is available when an Editor panel in mixed display mode is active.

4.2.6 Improvements to processing for saving the content of an editor to a file

The processing for saving the content of an editor to a file was improved to reduce the memory size used in the computer.

4.2.7 Raising of a caution

The following caution on an editor was raised.
• If a file opened in an editor is saved and error E2000003 occurs, the file size becomes 0 and the content of the file may be lost.

4.3 Improvements to the build tool

4.3.1 Addition of a facility for making dependency settings

A new facility to make settings for dependencies between projects was added. This makes it possible to control the build order of projects.

4.3.2 Enhanced Property panel

Locking of the Property panel of the build tool is now supported. Locking the panel prevents property settings from being modified by mishandling the mouse or keyboard.

4.3.3 Addition of the upper limit on the number of errors in building

An upper limit on the number of messages was added. When the number of output error messages exceeds the limit, CubeSuite+ will stop the build.

4.3.4 Enhanced facility for updating dependencies

Dependencies can be updated at either of the following two times.
• Between opening a project and executing the build for the first time (the same behavior as in earlier versions)
• Immediately before building

4.4 Improvements to the debugging tool

4.4.1 Enhanced Memory panel

The values of the following three items in the Memory panel can be edited even while the program is running.
• Notation
• Size Notation
• Encoding

4.4.2   Enhanced Watch panel

The states of [Notation], [Size Notation], [Encoding], and [Memo] for watched variables can be saved.
These settings are restored when the file containing data on the Watch panel is imported.

4.4.3   Enhanced Events panel (RX)

After a breakpoint, trace point, or timer event point is set, its address can be changed.

4.4.4   Addition of hot plug-in function for E1 emulator (RL78)

Hot plug-in function is added for RL78 E1 emulator.
Target device: RL78/F13, RL78/F14

4.5   Improvement of the program analysis tool

4.5.1   Enhanced Analysis Chart panel

Popup information on how to make settings in the Analysis Chart panel of the Program Analyzer was
added.

4.6   Improvement of Python console

4.6.1   Addition of Python functions

The following Python functions were added.

Function name	Overview
debugger.Timer.Delete	This function deletes a conditional timer.
debugger.Timer.Disable	This function disables a conditional timer.
debugger.Timer.Enable	This function enables a conditional timer.
debugger.Timer.Get	This function references the result measured by a conditional timer.
debugger.Timer.Information	This function displays conditional timer information.
debugger.Timer.Set	This function sets the conditional timer.
debugger.Trace.Clear	This function clears the trace memory.
debugger.Trace.Delete	This function deletes a conditional trace.
debugger.Trace.Disable	This function disables a conditional trace.
debugger.Trace.Enable	This function enables a conditional trace.
debugger.Trace.Get	This function dumps the trace data.
debugger.Trace.Information	This function displays conditional trace information.
debugger.Trace.Set	This function sets conditional trace information.

4.6.2   Addition of PythonClasses

The following Python classes were added.

Class name	Overview
TimerCondition	This class creates conditions of a conditional timer.
TimerEventInfo	This class holds conditional timer event information.
TimerInfo	This class holds conditional timer information.
TraceCondition	This class creates conditions of a conditional trace.
TraceEventInfo	This class holds conditional trace event information.

4.6.3   Addition of Python properties

The following Python properties were added.

Property name	Overview
debugger.IsMulticore	This property checks whether or not the microcontroller of the active project is multi-core.
debugger.ProcessorElement	This property sets or references the PE of the multi-core.

Chapter 5.   Cautions

This section describes cautions for CubeSuite+.

5.1   Cautions for CubeSuite+ (general)

5.1.1   Cautions for file names
The following cautions apply to folder and file names.
- Folder and file names
  Do not use folder or file names that cannot be created from Windows Explorer.
- Source file names, load module file names, and project file names
  File names consist of the characters a-z, A-Z, 0-9, period (.), underscore (_), plus (+), and minus (-).
  File names cannot end with a period (.).
  File names cannot start with a period (.).
  Filenames cannot start with a plus sign (+) or minus sign (-).
  File names are case-insensitive.
  File names may be up to 259 characters, including the path.
  Do not use source files whose file name is same. Even if they exist in a different path, CubeSuite+ can't classify them.
- File names other than the above.
  File names comply with Windows conventions.
  Note that the following characters cannot be used in file names.
  \ / : * ? " < > | ;
  File names cannot start with a period (.) or space.
  The uppercase and lowercase characters of the file name are not distinguished.
  File names may be up to 259 characters, including the path.
 - Folder names
  Comply with Windows file name conventions.
  Note that the following characters cannot be used in file names. (Excluding RL78, 78K0, 78K0R, and V850 projects) ( ) , =

5.1.2   Caution for panel displays
If your hardware environment does not meet the recommended specifications for CubeSuite+, the [Property] panel may appear small, and the contents scrambled.
If this happens, move the [Property] panel outside the split panel area.
- Enable Dockable, and make it a Docking panel
- Enable Floating, and make it a Floating panel

5.1.3   Caution for User Account Control (UAC) function (Windows Vista)
If the UAC function is disabled on Windows Vista / Windows 7, then if a user without administrator privileges creates a project, and no Device Dependence Information is installed, then the installation of the Device Dependence Information will begin, but the installation will fail. If the UAC function is disabled, create projects after logging in with administrator privileges.

5.1.4   Caution for command accelerators included in split panels/categories
Although accelerators are displayed in the command menus of split panels and categories, pressing the keyboard shortcuts will have no effect. Use the mouse to select menu items.

5.1.5   Caution for Windows update program
Your computer may suffer "blue screen" errors if you apply the KB2393802 patch published by Microsoft Corporation. If this error occurs, please apply the patch provided by your computer's manufacturer or other source.

RENESAS

5.1.6   Caution for Renesas Electronics real-time OS

If you use the real-time operation system for the RX family provided by Renesas Electronics, install CubeSuite+ to a folder path that does not contain parentheses. If you install it to the 64-bit version of Windows, it will be installed in the "Program Files (x86)" folder by default, and if the folder path includes parenthesis characters, it will result in an error.

5.1.7   Caution with regard to changing the microcontroller

Note the following points for caution when changing the microcontroller.
- The microcontroller can only be changed to another within the same family, since this will correspond to the same build tool (RH850, V850, RX, RL78, 78K0R, and 78K0).
- When changing the microcontroller, do so while the debugger is not connected.
- Save the project before changing the microcontroller.
- Information on pin layout (design tool), code generation (design tool), and debugging (except for watch registration information) are not carried over after the microcontroller has been changed.
When you are working with a project for a multi-core RH850 product, select the [Microcontroller] node of all projects and change the target microcontroller at the same time. If the microcontroller for a project is changed individually, the projects that belong to [Constituent application projects] will be deleted.

5.1.8   Caution with the Plug-in Manager Function

We recommend that the checkbox for the plug-in for the microcontroller that is the target for development is not deselected on the Basic Function tabbed page of the Plug-in Manager dialog box. Deselect the checkboxes for the build tool and debugging tool plug-ins that are for microcontrollers that are not the target for development.For example, if only the plug-in for the build tool is deselected, the file to be downloaded by the debugging tool will not be found and an error will occur.

5.1.9   Cautions for Editor panel

- When you change the active file using the File tab, the "Forward to Next Cursor Position" and "Back to Last Cursor Position" features may not work.
- The Page Setup dialog box is not available.
- Although there is a Copy button on the Print Preview toolbar, it does not work..
- When a variable or label is selected and the Jump to Function feature is used from the context menu, execution does not jump to the variable or label.
- The Jump to Function feature will not jump to a static function defined in another file.
- The following cautions apply to editor, when the source file of the same name from which a folder is different was registered with a main project and a sub project, and downloading a load module of a main project and a sub project both.
  - The address of the main project is displayed on the file.
  - At jumping to a source file from disassembly code, the file registered with a main project is opened.
  - If the file is opened from whichever project s, only 1 file is opened.
- The smart edit function does not run correctly for a structure that does not have a name.
- If the arguments of a function include a function call, incorrect information will be displayed in the tooltip.
- Member variables or functions are not correctly complemented in arrays of classes and of pointers to classes.
- Complementation does not produce appropriate strings even if a part of a member name is entered and 'ctrl + (space)' is also entered.
- While a debug tool is connected to CubeSuite+, breakpoints and timer/trace events cannot be set on the last line of each source file. If you wish to set breakpoints or timer/trace events, insert an empty line at the end of the file.
- The total number of pages indicated on the footer becomes 0 when the data in the panel are printed out (e.g. Page 3 of 0).
To correctly print out the total number of pages, print via a print preview where the correct number is displayed.
- Outlining (collapse/expand) only applies to files that have been registered with the project. Any files that have not been registered with the project will not be outlined in the Editor panel of CubeSuite+.
- On Windows 8, the display may become unclear due to anti-aliasing.
- In mixed display mode, if a line number is specified for jump, a disassembly code is inserted and the specified line may not be displayed on the screen.
- If a structure is nested, the smart edit function is not available in the third and the subsequent stages. In addition, the information on toolchip is not displayed.
- The code immediately after '#ifdef - #endif' does not show candidates of members in the smart edit. In addition, no toolchip is displayed.

- For the first variable in '#ifdef - #endif', the codes after '#endif' do not show candidates of members in the smart edit. In addition, no toolchip is displayed.

5.1.10 Caution for conversion from PM+ to CubeSuite+ project.

CubeSuite+ can't read the CA850 project, if the project is made by PM+ V6.00/V6.10/V6.11 and is added a new Build Mode. It'll be as follows.

1)When Debug Build or Release Build is specified, information on the added Build Mode can't be read.

2)When the added Build Mode is specified, it'll be an error.

[Workaround]
Please read the project by more than PM+ V6.20 and save it.

5.1.11 Cautions for Debugging Tool Settings during Project Composition

When you create a project by composite the settings of another project, only the settings for the default debugging tool will be imported. In the RX family, however, internal processing is common to the emulator and simulator, so the settings are imported regardless of which debugging tool is selected.

5.1.12 Cautions for Online Help

If you close the online help while the Search tab is displayed, and you then display the online help again and display the Contents tab, the Coding and Build editions may disappear.
If this happens, close the online help with the Contents tab displayed, and then display the online help again.

5.1.13 Cautions for project conversion

When changed conversion device of a project in [project convert setting] dialog when opened High-performance Embedded Workshop/PM+/CubeSuite, return a value chosen in [king of project] to the value that an initial value appears.

For example:
A kind of a project is replaced by the top (for example, [Application]) when chooses a device again in the back.

RENESAS

5.1.14 Note on Conversion of High-performance Embedded Workshop Projects

If you attempt to load a High-performance Embedded Workshop project into the CubeSuite+ under certain conditions, an error may occur during conversion or building of the project.

(1) Converting a High-performance Embedded Workshop project to make it compatible with the CubeSuite+ fails when any of the following conditions is satisfied.
  ➢ No toolchain from Renesas Electronics Corp. is selected for the project.
  ➢ The project contains no tps file that is used to set up the High-performance Embedded Workshop environment. (the tps file is automatically created when the project is opened through the High-performance Embedded Workshop). To avoid this problem, you should open the project through the High-performance Embedded Workshop once before starting conversion.
  ➢ The project contains multiple CFG files, each of which is used to set up the realtime OS from Renesas Electronics Corp.

(2) Converting a High-performance Embedded Workshop project to make it compatible with the CubeSuite+ succeeds but building of the project leads to an error when any of the following conditions is satisfied.

  ➢ Placeholder $(TCINSTALL) is used in the project.
      $(TCINSTALL) remains in the project even after conversion but the CubeSuite+ does not recognize $(TCINSTALL). Placeholder $(TCINSTALL) that has been used as a parameter for [Options] in the High-performance Embedded Workshop is simply passed to the CubeSuite+ and may cause an unintended result (e.g. an error) upon building of the project. For this reason, you should manually change $(TCINSTALL) after converting the project.

  ➢ Placeholder $(WORKSPDIR) is used in the project.
      If you select a HEW project file (with extension hwp) in the CubeSuite+, it is automatically converted to "%ProjectDir%\.." (the directory above the project directory). An error may occur during building of the project if the workspace does not exist in the directory indicated by "%ProjectDir%\..".
      For this reason, you should manually change "%ProjectDir%\.." after converting the project.

  ➢ A custom build phase is used in the project.
      Since all custom phases are deleted upon conversion, an error may occur during building of the project that involved a file output created for a custom build phase in the High-performance Embedded Workshop.
      After converting the project, register the custom build-phase command with the CubeSuite+ as a command to be executed before or after each phase as required.

  ➢ A custom placeholder is used in the project.
      Custom placeholders are not converted because the CubeSuite+ does not recognize them. Any custom placeholder that has been used as a parameter for [Options] in the High-performance Embedded Workshop is simply passed to the CubeSuite+ and may cause an unintended result (e.g. an error) upon building of the project. For this reason, you should manually change the custom placeholder after converting the project.

(3) Other
  (a) $(FILEDIR) is converted to %FileDir%.
      Leaving %FileDir% as it is when the pathname is edited in the [Path Edit] dialog box will lead to the following error: The specified path contains a non-existent folder. (W0205012)
      When you edit the pathname, replace %FileDir% with another placeholder or directory name.
  (b) $(WINDIR) is converted to %WinDir%.
  (c) The order in which folders are displayed in CubeSuite+ may differ from that in the High-performance Embedded Workshop.
  (d) If a High-performance Embedded Workshop project for which downloaded files have been specified is loaded into CubeSuite+, CubeSuite+ will show these files as the second and subsequent items in the list of downloaded files for each debug tool.
  (e) The compiler option -output=src is converted to -output=obj (default).
  (f) If you load a library project that has been linked to the standard library into CubeSuite+, the linkage setting will be discarded (this is indicated in the log information that is output as a result of loading the project).
  (g) If [Use an existing library file] has been selected for the library generator in the High-performance Embedded Workshop, the setting is changed to [Do not add a library file] in CubeSuite+. For this reason, linking with the specified library will not proceed (this is indicated in the log information that is output as a result of loading the project).
  (h) Option settings that were made on the [Toolchain Option] tabbed page of the High-performance Embedded Workshop are not converted but discarded (i.e. they are not moved across to CubeSuite+).

RENESAS

(i) If a sub-command file has been selected for the linkage editor in the High-performance Embedded Workshop, the [Use external subcommand file] setting is discarded when the project is loaded into CubeSuite+. The linkage editor options will have their default settings.

(j) Any files specified with the -library, -input, or -binary option will not be listed in the [Link Order] dialog box. The result is that the order of linkage for these files will not be selectable.

(k) RTOS configuration files will not be displayed under the [Configuration file] category node after the project is loaded into CubeSuite+.

(l) RTOS option settings that were made in the High-performance Embedded Workshop are discarded. RTOS options will have their default settings in CubeSuite+.

(m) The build mode for RTOS projects will be "DefaultBuild" after the project is loaded into CubeSuite+. You will need to change the build mode as required.

(n) The order of linkage of the assembly output file (ritbl.obj) in an RTOS project will differ from that in the High-performance Embedded Workshop.

5.1.15 Caution for creating new projects

Applies to: RX

If a new project is created by selecting [Empty Application[CC-RX]] under the environment for the RX, building the project may lead to the following errors.

      ** L2132 (E) Cannot find "D" specified in option "rom"
      ** L2132 (E) Cannot find "D_1" specified in option "rom"
      ** L2132 (E) Cannot find "D_2" specified in option "rom"

If you encounter these errors, change the setting of [ROM to RAM mapped section] on the [Link Options] sheet in CubeSuite+.

5.1.16 Caution on Microsoft IME

If you are using Microsoft Office IME 2010, which is included in Office 2010 from Microsoft Corporation, CubeSute+ may output error E2000006.

Since Microsoft Office IME 2010 may have caused this problem, replace it with Microsoft's standard IME or install the module provided by Microsoft Corporation to fix KB2687611 of Microsoft Office IME 2010.

5.1.17 Caution on Tutorial

Code Generator Plug-in, Pin Configurator Plug-in and Program Analyzer Plug-in are used in tutorial. Enable them through the Plug-in Manager dialog box.

5.2 Cautions for Design Tool

5.2.1 Caution for changing packages

If you change the package name in the pin layout properties, the data input in the device top view and device pin list will be cleared.

5.2.2 Caution for saving projects

If you save a project that has sub-projects while the Device Top View or Device Pin List panel is open, then the device top view and device pin list of the last sub-project in the Project Tree will always appear.

5.2.3 Caution for saving projects

Applies to: 78K0 / 78K0R / RL78

A value of "Use on-chip debug" or "Set user option byte" on [Link Option] Tab may be different between on saving a project file and on reading a project file.

[Condition]

1) The log-in user's own .mtud file doesn't exist when reading a project file
   Example 1
   Another log-in user B read the project file that a log-in user A saved
   Example 2
   A log-in user A read a project after the user saved the project file and deleted .mtud file by intent.
2) The log-in user's own .mtud file exists and the cord generation panel is displayed on the forefront.

[Procedure]

After "read project file" or before "build", verify that the values are correct.

5.3　Cautions for debugging tool

The below abbreviated names are used in this section.

OCD(Serial)　　　　　: MINICUBE2, E1 Emulator(Serial), E20 Emulator(Serial)
OCD(JTAG)　　　　　: MINICUBE, E1 Emulator (JTAG), E20 Emulator (JTAG)

5.3.1　Caution for adding sub-projects

Applies to: All debugging tools,Common to all devices

Disconnect the debugging tool before adding a sub-project that handles a different device than the main project.

5.3.2　Caution for executing a boot swap

Applies to: Simulator/ OCD(JTAG)/ OCD(Serial), V850 / 78K0 / 78K0R / RL78

If a software break is set in a boot-swap area, then a break instruction will be written to the Flash ROM. For this reason, a break instruction will remain after the boot swap.

- OCD(JTAG)/ OCD(Serial) : Use a hardware break if you wish to set a breakpoint.
- Simulator　　　: Don't use break point in this area.

5.3.3　Caution for standby mode

Applies to: All debugging tools, V850 / 78K0 / 78K0R / RL78

If a forced break is performed while in standby mode (e.g. STOP mode or HALT mode), or an instruction to move to standby mode is made while in step execution, then behavior will differ between the simulator and the emulator (IECUBE, OCD(JTAG), and OCD(Serial)).

- Emulator: The forced break will release standby mode. In step execution, it will not go into standby mode.
- Simulator: The forced break will not release standby mode. In step execution, it will go into standby mode.

In either case, the program counter (PC) row upon forced break will break at the next instruction after the standby mode instruction (e.g. HALT). Thus in the case of the simulator, it will appear that standby mode has been released. Check the status bar to see if standby mode has been released. If the simulator is in standby mode, "Halt" or "Standby" will appear in the status bar.

5.3.4　Caution on low-power consumption modes

Applies to: All debugging tools, RX

When a forced break occurs in a low-power consumption mode (e.g. sleep, stop, or standby) or an instruction that makes the CPU enter a low-power consumption mode is executed during stepped execution, the behavior of the simulator and the emulator will differ as follows.

- Emulator:　A forced break releases the CPU from the low-power consumption mode. On the other hand, the CPU can enter a low-power consumption mode during stepped execution.
- Simulator:　Transition to a low-power consumption mode (e.g. by a register setting) is not supported. Executing a WAIT instruction causes a break, with the PC placed at the address of the next instruction. During stepped execution, the CPU also does not enter a low-power consumption mode and the PC is placed at the address of the next instruction.

5.3.5　Caution for multipliers/dividers

Applies to: Simulator, 78K0

When simulating 78K0 instructions, the multiplier and divider are not supported. For this reason, to perform multiplication or division within the program, from the build tool, open the Property panel, and on the [Compiler Options] tab, from the [Use multiplier/divider] drop-down list, select [No].

5.3.6　Caution for Memory bank function

Applies to: Simulator, 78K0

When simulating 78K0 by instruction mode, the memory bank function is not supported.

5.3.7　Caution for CPU operation clock

Applies to: Simulator, 78K0R / RL78

- When simulating 78K0R by instruction mode, the frequency of on chip oscillator is 8MHz.
- When simulating RL78 by instruction mode, the CPU operation clock operates by the specification of RL78/G13

RENESAS

5.3.8   Caution for Multiplier and Divider/Multiply-Accumulator
Applies to: Simulator, 78K0R / RL78
When simulating 78K0R or RL78 by instruction mode, cautions of Multiplier and Divider/Multiply-Accumulator are following.
   (1)   When using it by division mode, the division processing will be finished in by 1 clock.
   (2)   When using it by division mode, the interrupt "INTMD" (the end of division operation) is not occurred. But DIVST bit of Multiplication/Division Control Register "MDUC" is changed. (DIVST bit displays division operation status.)

5.3.9   Caution for traces in arbitrary intervals
Applies to: Simulator, all devices
If you perform a trace from a trace start event until a trace end event, the simulator will not display the trace end event as the results of the trace. For this reason, if you are using a simulator, set the trace end event to one line below the range that you wish to display as the trace data.

5.3.10   Caution for runtime measurement over arbitrary intervals
Applies to: Simulator, V850 / 78K0 / 78K0R / RL78
If you measure run time from a timer start event until a timer end event, the simulator will not include the time for the timer end event in the measurement results. For this reason, if you are using a simulator, set the timer end event to one line below the range for which you wish to measure the run time.

5.3.11   Caution for displaying maximum address space in memory display panel
Applies to: OCD(Serial) / IECUBE, 78K0
To access the device maximum internal ROM, internal fast RAM, or internal extended RAM sizes from the memory panel or the like, set a hook process in the memory size switch register (IMS) and internal extended RAM size switch register (IXS).

5.3.12   Caution for retuning execution and displaying the call stack
Applies to: All debug tools, 78K0R / RL78
If step execution is performed from the editor panel (in source mode), the debugging tool determines whether an interrupt is being processed via the NP, EP, and ID flags in the POWER switch register. For this reason, if the above flags or register are changed (e.g. when using multiple interrupts), then the return execution and displaying the call stack may be incorrect.

5.3.13   Adding sub-projects
[Applies to] Common to all debug tools and devices
If you add a sub-project while a debugging tool is connected, downloading and the like may fail. Add sub-projects while the debugging tool is disconnected.

5.3.14   Configuring Flash options
[Applies to] OCD(JTAG), V850E2M
The bit shown below indicating the following Flash option has been locked to 1. Use a Flash programmer if you wish to write 0 to it.
 - Bit 95 of on-chip debugging security ID (security lock signal release)
 - Bit 31 of option byte 0 (debug interface connection disabled bit)

5.3.15   Caution for Stack-trace display
[Applies to] All debugging tools, 78K0
The stack-frame display function may fail to correctly display up to the **main** function if a function is used that does not push the frame pointer (HL) onto the stack (e.g. **noauto** or **norec** function), or if the memory bank is used.
Additionally, a free-run state may occur if a return is executed from a function that does not push the frame pointer (HL) onto the stack (e.g. **noauto** or **norec** function), or if a memory bank function is used.

5.3.16   Caution for stepping into main bank
[Applies to] All debugging tools, 78K0
If you step into a user-defined library function or function without debugging information in the memory bank at the source level, execution will break in the bank-switching library.

5.3.17 Caution for local-variable display
[Applies to] All debugging tools, 78K0
Local variables outside the scope of the current PC are not displayed correctly in the stack trace panel.

5.3.18 Caution for disassemble window
[Applies to] All debugging tools, 78K0
When displaying instructions in the common area in the disassemble window, if the displayed instruction uses a symbol in the memory bank area, a symbol from a different bank may be displayed.

5.3.19 Breakpoint and other settings become invalid
[Applies to] Common to all debugging tools and devices
If you differentiate function or variable names by leading underscores, then the debugger may misrecognize them, and convert symbols or make breakpoint settings invalid.
This applies for cases like when you have two functions, one named _reset and the other named __reset.

5.3.20   Caution on two or more variables with the same name
Applies to: All debug tools, RX
When two or more variables with the same name are defined in unnamed name spaces written in different source files, the Watch panel only shows the information on the variable that was found first.

5.3.21   Caution on member-variable pointers
Applies to: All debug tools, RX
After the member-variable pointer "mp1" defined in the program below is registered with the Watch and Local Variables panels, the type of the pointer is indicated as "int *", not "int Foo::*".
```
class Foo {
    int   m1;
};
int Foo::*mp1 = &Foo::m1;
```

5.3.22   Caution on unions assigned to registers
Applies to: All debug tools, RX
When a union is assigned to a register, it is assumed that the members of the union are assigned to the lower-order bytes of the register. For this reason, the values of the members in big endian being displayed are incorrect.

5.3.23   Caution on functions with the same name and char-type parameters
Applies to: All debug tools, RX
When three functions with char-type parameters are defined as shown below, the address of "Func(signed char)" is not displayed (i.e. the address of "Func(char)" is displayed instead).
```
void Func(char);
void Func(signed char);
void Func(unsigned char);
```

5.3.24   Caution on char-type one-dimensional arrays
Applies to: All debug tools, RX
When a char-type one-dimensional array is assigned to multiple locations in registers or memory as shown below, no character string will be displayed in the value column of the Watch or Local Variables panel even after the array "array" has been registered with the panel. ("" is shown in the column.)
```
char array[5] = "ABCD";
```

5.3.25   Caution on changing the priority section of overlay sections
Applies to: All debug tools, RX
Changing the priority section of overlay sections does not immediately reflect the debugger operation. To update the display of addresses in the editor, for example, you need to close the file and open it again. To update the display of variables in the Watch panel, single step in the program.

5.3.26   Caution on variables assigned to registers

Applies to: All debug tools, RX

When the selection for [Scope] in the Local Variables panel is not "Current", the values of variables assigned to registers are not displayed correctly. Editing these values is also not possible.

5.3.27   Caution on the locations where variables are assigned

Applies to: All debug tools, RX

When a defined variable satisfies both of the two conditions given below, the location of its member variables indicated in the Watch and Local Variables panels is actually the location of the entire variable.
Conditions:
(1) The variable is assigned to two or more addresses or registers (i.e. two or more addresses or registers are displayed in the address column).
(2) A structure-, class-, array-, or union-type member is defined in the variable.
Example:
```
struct Mem {
    long m_base;
};
struct Sample {
    long m_a;
    struct Mem m_b;   <- Condition (2)
};

main () {
    struct Sample obj;
}
```

Display in the Watch and Local Variables panels:
```
"obj"                         -        { R1:REG, R2:REG }      (struct Sample)
   L m_a          0x00000000   { R1:REG }              (long)
   L m_b                -       { R1:REG, R2:REG }      (struct Base)
      L m_base 0x00000000  { R2:REG }              (long)
```

5.3.28   Caution on casting variables

Applies to: All debug tools, RX

When the variable was cast to other type in watch panel,the variable is cast to C-style.
For this reason, the result that Virtual inheritance class was cast to base class is not same the cast result described in the program
```
class AAA [
    int m_aaa;
} objA;
class BBB : public AAA {   // BBB inherits AAA.
    int m_bbb;
} objB;
class CCC {   // CCC does not inherit AAA.
    int m_ccc;
} objC

class AAA* pa = objA;
class BBB* pb = objB;
class CCC* pc = objC;
```

```
"(AAA*)pa"               Available
"(BBB*)pb"               Available
"(AAA*)pb"               Available
"(CCC*)pc"               Available
"(AAA*)pc"               Not available because it considers that the address which pc points out is a
top address of AAA.
  cast image :(AAA*)((void*)pc)
```

5.3.29   Caution on sleep state of PC

Applies to: OCD(JTAG), OCD(Serial),RX

When PC shifts to a sleep state using Windows Vista or Windows 7, Cubeuite+   cannot be continued to debug after a return from sleep state.

 Please set up PC ,in order not to   shift to a sleep state.

5.3.30   Caution on the trace stop and restart during program execution

Applies to: All debug tool,RX

When the trace start event or the end event set up, the trace can not stop and restart during the program execution.

5.3.31   Caution on the timestamp of trace

Applies to: OCD(JTAG), OCD(Serial),RX

When the time between frames exceeded a trace counter (20 bits), and when trace type was lost,the time stamp of trace does not display the right time.

5.3.32   Caution for link option of CC-RX

Applies to: All debug tool,RX

CC-RX does not support link option '-sdebug'.

Please set [Outputs debugging information] in [output] category of [Link Options] tab to '-dubug'.

5.3.33   Caution for Clock Generating Circuit

 [Applies to] Simulator, 78K0/Kx2

The Clock Generating Circuit is simulated by the specification of 78K0Kx2.

5.3.34   Caution for Data Flash

 [Applies to] Simulator, RL78 / 78K0R

The data flash is not supported.

The data flash area is treated like the usual RAM.

5.3.35   Caution for Code Flash

[Applies to] Simulator, RL78 / 78K0R

 4 bytes of the final address of the code flash area and the RAM area which can be fetched cannot be fetched.

 ex) When a code flash area is 0x0-0x1ffff, 0x1fffc-0x1ffff corresponds to restriction.

5.3.36   Caution for Pipeline

[Applies to] Simulator, RL78 / 78K0R

 The Pipeline is not supported.

5.3.37   Target system voltage during a break

 Applies to: OCD(JTAG)/ OCD(Serial), 78K/RL78/V850/V850E2M

 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 the debugger or communication errors.

 A break during emulation of power supply off also causes communication errors.

5.3.38   Caution of Debugger Property Panel

 Applies to: OCD(Serial), V850

 Open-Break function is not alowed to enable/disable on [Debug Tool Settings] Tab of Property Panel.

 Please set Open-Break function on Python Console.

5.3.39   Caution of generated load module by GHS compiler

Applies to: IECUBE/OCD(JTAG)/ OCD(Serial), V850

When using defined valuable of undefined function as Scope Specification, errors may not be generated.

Don't use Scope Specification with undefined function.

5.3.40   Caution of Return Out execution

Applies to: All debug tools, RX

When Return Out execute in the function of a recursive call,it may stop at the address which executed the return instruction in the call place function instead of the calling function.

5.3.41   Caution of Start up program protection function

Applies to: OCD(Serial), RX100

If "CPU reset" is executed after one of the following operations is performed during the execution of the user program, the contents of ROM which a debugger displays and the contents of ROM of MCU are not in agreement.

In this case, please execute the user program again and stop the user program.

  Calling the R_FCL_ChangeSwapFlag function

  Control the flash initial setting register (FISR) to immediately swap blocks

5.4   Cautions for analysis tool

5.4.1   Cautions for Analysis Chart panel(E1/E20,RX)

- Execution Time(Percentage) chart tab of Analysis Chart panel cannot be displayed.(E1/E20, RX/RH850)
- Variable Value Changing chart tab of Analysis Chart panel cannot support "Analysys of trace data" mode. (E1/E20,RX/RH850)
- When a simulator is specified as a debugging tool in a graph of transition in values, realtime sampling methods for IOR/SFR are not supported.

5.4.2   Cautions for Function List / Variable List panel (E1/E20,RH850)

- Function List panel cannot support "Execution Time", "Execution Time(Percentage)" and "Code Coverage".
- Variable List panel cannot support "Data Coverage".

5.5   Cautions for Python Console

5.5.1   Caution for Japanese input

The Japanese input feature cannot be activated from the Python Console. To enter Japanese text, write it in an external editor or the like, and copy and paste it into the console.

5.5.2   Caution for prompt displays

The Python Console prompt of ">>>" may be displayed multiply, as ">>>>>>", or results may be displayed after the ">>>", and there may be no ">>>" prompt before the caret. If this happens, it is still possible to continue to enter functions.

5.5.3   Cautions for paths to folders and files

IronPython recognizes the backslash character (\) as a control character. For example, if a folder or file name starts with a "t", then the sequence "\t" will be recognized as a tab character. Please use [r + "path name"] to avoid this.

  Example: r"c:\test\test.py"

A forward slash (/) can be used instead of a backslash (\).

5.5.4   Caution for executing scripts for projects without load modules

If a script is specified in the startup options that uses a project without a load module file, or if project_file.py is placed in the same folder as the project file, then although the script will be executed automatically after normal project loading, it will not be executed if there is no load module file.

5.5.5   Cautions for forced termination

If the following operations are performed while a script like an infinite loop is running, then the results of function execution may be an error, because the function execution will be terminated forcibly.
1. Forcible termination by selecting "Forcibly terminate" from the context menu or pressing Ctrl+D in the Python Console
2. Changing the active project in a project with multiple projects

5.5.6   Caution for forced stops

Executing "Abort" from the context menu will forcibly stop an executing script or function, but hook and callback functions that had not started at the time the "Abort" was executed will execute in order afterward.

Chapter 6.   Restrictions

This section describes restrictions for CubeSuite+.

6.1   Restrictions for debugging tool

The below abbreviated names are used in this section.
OCD(Serial)   : MINICUBE2, E1 Emulator(Serial), E20 Emulator(Serial)
OCD(JTAG)   : MINICUBE, E1 Emulator (JTAG), E20 Emulator (JTAG)

6.1.1   List of restrictions for debugging tool

No.	Target tool	Target device	Description
1	OCD(JTAG)	V850E2M	Restriction for Flash options
2	OCD(serial)	RL78	Restriction for stopping a high-speed on-chip oscillator during subsystem clock operation by using RL78/G14.

6.1.2   Details of restrictions for debugging tool

No.1 Restriction for flash options
Applies to: OCD(JTAG), OCD(Serial), V850E2M
Description: All security settings and boot-block cluster settings of the Flash Option Settings property are invalid.
Workaround: There is no workaround

No.2 Restriction for stopping a high-speed on-chip oscillator during subsystem clock operation by using RL78/G14
Applies to: OCD(Serial), RL78
Description: When meeting all of the following conditions by using RL78/G14, an error message "Executing program failed" is displayed.
[Conditions]
1. "Monitor clock" of a "Connect Settings" tab in a property panel is set as "System".
2. High-speed on-chip oscillator is set as 64MHz or 48MHz by the user option byte.
3. High-speed on-chip oscillator is stopped.
4. System clock is subsystem clock.
5. No.2-4 is set during user program execution. After a break, a program is executed again.
Workaround: Please set "Monitor clock" of a "Connect Settings" tab in a property panel as "User". But, the response of debugging operation by the CubeSuite+ sometimes becomes slow during a break by this setting change.

Chapter 7.   Changes in User's Manual

This section describes Changes in User's Manual in CubeSuite+.

7.1   Modifications of Editor

This section describes additional functions of the editor. The editor is described in the Coding and Debugging editions.

7.1.1   Additional description of toolbar

[Before addition]
None
[After addition]

	This toggles between normal and mixed display modes when the debugging tool is connected. In normal display mode, the source program is displayed. In mixed display mode, the source program and assembly code are displayed.
	This toggles between step execution at the source and assembly level when the debugging tool is connected and in mixed display mode. At the source level, the program counter indicates the line in the source program. At the assembly level, the program counter indicates the line in the assembly code.
	This displays the current position of the program counter (PC) when the debugging tool is connected and in mixed display mode.
	This jumps to the position before executing the **Back to Last Cursor Position** command.
	This returns to the position before executing the **Jump to Funciton** command.
	This toggles between showing and hiding the column. Clicking this displays the areas that can be toggled.

7.1.2   Additional description of column headers
[Addition]

The following information items are displayed in the respective columns.

Display on the column header	Description of the items
Line	Line numbers for the files being displayed
(No display)	The display is colored to reflect the state in terms of saving of the state of editing. Yellow: The line has been newly created or modified but the file has not yet been saved. Green: The line has been newly created or modified and the file has been saved. This column is displayed except when the mixed display mode is chosen.
(No display)	The display is colored to reflect cases where a source file has been updated more recently than the corresponding load module file. The display in this area is cleared when the load module file is downloaded again after executing a build. This column is displayed if the normal display mode is chosen when a debug tool is connected.
▦	When the coverage function is enabled, the display is colored reflecting the results of code coverage measurement. This column is displayed when a debug tool is connected.
Address	Addresses in the microcontroller This column is displayed when a debug tool is connected.
Op code	Instruction codes This column is displayed if the mixed display mode is chosen while a debug tool is connected.
Label	Labels This column is displayed if the mixed display mode is chosen while the debug tool is connected.
▣	Items in this column indicate the setting of events such as timer and trace events by the context menu on lines having an address display. When an event has been set, an event mark indicating its type is displayed here. 🔔 Trace event enabled 🔔 Trace event disabled 🔔 Action event enabled 🔔 Action event disabled This column is displayed when a debug tool is connected.

	This column displays the breakpoint enabled/disabled state, current PC location, bookmarks, and address marks. On the line where an address mark is shown, this column is used to set a breakpoint. This column is manipulated from the context menu in most cases, but to set or clear a breakpoint, click this column with the left mouse button. If a breakpoint has been set, an event mark is displayed here. A mark indicating the current PC location is also displayed if the current PC location (PC register value) is included in the active panel (only in the state where the program is halted). Breakpoint enabled Breakpoint disabled Current PC location Bookmark Address mark

7.1.3 Additional description of dragging and dropping

[Before addition]
None
[After addition]
You can add symbols to Watch panel or Analysis Chart panel by dragging and dropping them.

7.1.4 Additional description of highlighting

[Addition]

• Highlighting the current line

The current line (where the caret is located) can be displayed within a rectangle.

This feature can be enabled on the [Text editor] page of the [Option] dialog box.

• Emphasizing brackets

The editor is capable of emphasizing the bracket that corresponds to a bracket at the caret position. Supported types of brackets vary with the file type.

File Type	Types of Brackets
C or C++	(and), { and }, [and]
Python	(and), [and]
HTML or XML	< and >

When CubeSuite+ searches for 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.

7.1.5 Additional description of zooming in and out

[Addition]

The display in the editor is in accord with the zoom ratio selected from the toolbar of CubeSuite+. When you select a zoom ratio, it is applied to the panel currently in focus. If you close the panel and open the same file again, the zoom ratio will be 100%. Separate zoom ratios can be selected for the normal mode and the mixed display mode (with the debug tool connected).

7.1.6 Additional description of smart edit functions

[Addition]

The smart edit function is used to complement the names of functions, variables and the parameters of functions during input and offer them as candidates.

This function is only supported if the project is for the RH850, V850E2 or RX family.
[RH850][V850E2] [RX]

The function operates with the items listed below.
- Global functions in the C language and C++ language
- Global variables in the C language and C++ language
- Class member functions in the C++ language
- Class member variables in the C++ language
- Multiple overloaded functions in the C++ language

To display candidates for functions and variables for the smart edit function, operate the program as described below.
 (1) In cases where candidates are automatically to be shown
- In the C language and C++ language, if there is a relevant member for the left side when '.' is input, candidates are automatically shown at the location of the text cursor.
- In the C language and C++ language, if there is a relevant member for the left side when '->' is input, candidates are automatically shown at the location of the text cursor.
- In the C++ language, if there is a relevant member for the left side when '::' is input, candidates are automatically shown at the location of the text cursor.
- If there is a relevant method (function) on the left side of '(' when '(' is input, candidate parameter names are automatically shown.

 (2) In cases where candidates are to be shown in response to keyboard and mouse operations
- When Ctrl + Space is pressed on the keyboard, all candidates are displayed at the location of the text cursor. If there is only one candidate, the relevant character string is pasted without displaying the candidate.
- When Ctrl + Shift + Space is pressed on the keyboard while the text cursor is at the location of a parameter for a method (function), a list of candidate parameter names will be shown.
- Information to complement the name of a method (function) and variable is displayed in a tooltip when a mouse cursor is placed on the partial method (function) or variable name (only when a debug tool is not connected).

Any of the actions listed below stops the display of candidates by the smart edit function.
- Pressing the Esc key
  Pressing the Esc key while the smart edit function is displaying candidates for functions and variables halts the display of candidates.
- Input of a character other than an alphanumeric character while nothing is selected from the candidate list
  When no candidate for a function or variable is selected, the input of a character other than an alphanumeric character halts the display of candidates.
- Input of a character other than an alphanumeric character while an item in the list of candidates is selected
  When no candidate for a function or variable is selected, the input of a character other than an alphanumeric character leads to pasting of the selected candidate and thus stops the display of candidates.

Follow the procedure below to enable the smart edit function.
- In the [Text editor] page on the [Option] dialog box, select [Smart edit].
- Set the build tool property so that cross-reference information is output.
- Candidates are displayed by using cross-reference information. Execute and complete a build. If an error in building occurs, cross-reference information before the error occurred is used if any exists (when the output of cross-reference information by the build tool is disabled, the smart edit function cannot be used since the output will be empty of cross-reference information).

Note the following items regarding the smart edit function.
- Cross-reference information is necessary.
- The smart edit function cannot be used while the mixed display mode is in use and a debugger is connected.
- The function does not support macro definitions.
- The function does not support local variables.
- The function does not support the display of candidate members of types defined by using typedef. [V850E2]
- The function does not support namespaces and subclasses (classes declared within classes) of C++.
- When a structure, union or class is declared within a function in C++, candidates are not displayed within the function after its own declaration.
- The attributes const and mutable are not displayed.
- Candidates for functions and variables are not displayed in C++ in response to the input of "(*class)" or "(*this)".
- Candidates for functions and variables are not displayed in an array declared in a class in C++.
- In C++, when candidates for functions and variables are displayed by using the Ctrl + Space key while a class name is specified to the left and input of a method name is incomplete, candidates for global functions and variables are displayed instead of candidates for functions and variables in the class.
- 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. For example: when "Yes" is set for "Replace the int type with the short type" in CC-RX, a variable declared as int is displayed as short.
- Tooltip information is not displayed while the debug tool is connected.
- The tooltip display does not support the friend attribute of C++.
- The tooltip display does not indicate the const, static, volatile and virtual attributes.
- The tooltip display does not show structures, unions, and member functions declared in C++ header files.
- When the build tool is CX and a union is declared, a tooltip is not displayed even if a mouse cursor is placed over the tag name of the unions.

7.1.7   Additional description of the smart editing display
[Addition]

Display of candidates for functions and variables

The smart edit function is used to display candidates for functions and variables.
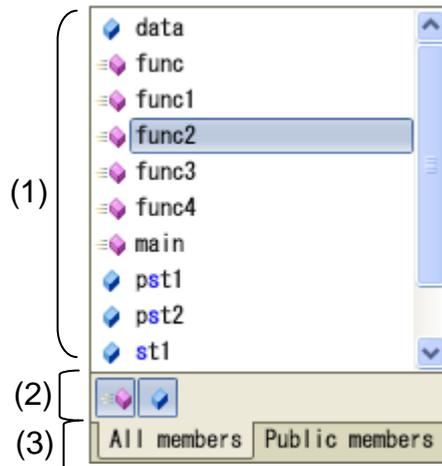
The following items are described.
- [How to open the list of candidates]
- [Descriptions of the individual areas]

[How to open]

(1) In cases where candidates are automatically to be shown
- In the C language and C++ language, if there is a relevant member for the left side when '.' is input, candidates are automatically shown at the location of the text cursor.
- In the C language and C++ language, if there is a relevant member for the left side when '->' is input, candidates are automatically shown at the location of the text cursor.
- In the C++ language, if there is a relevant member for the left side when '::' is input, candidates are automatically shown at the location of the text cursor.
 (2) In a case where candidates are to be shown in response to keyboard operations
- When Ctrl + Space is pressed on the keyboard, all candidates are displayed at the location of the text cursor. If there is only one candidate, the relevant character string is pasted without displaying the candidate.

[Descriptions of each area]

(1) Area for displaying candidates
   Candidates for functions and variables are displayed in alphabetical order.
   If a key is pressed while the smart editing display is active, the corresponding character strings are highlighted.
   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 or an union.

	Shows that the candidate is for a namespace.
	Shows that the candidate is for a protected member.
	Shows that the candidate is for a private member.

A candidate for a string is inserted at the location of the text cursor by selecting a candidate from the list of candidates and entering the Enter key or the Tab key.

(2) Toolbar area

Switches whether candidates for functions and variables are displayed or not.

Icon	Description
	Selected: Candidates for methods (functions) are displayed.
	Not selected: Candidates for methods (functions) are not displayed.
	Selected: Candidates for variables are displayed.
	Not selected: Candidates for variables are not displayed.

(3) Tab area

Switches the members to be displayed.

Tab name	Description
All members	All candidates are displayed.
Public members	Only the candidates with the public attribute are displayed.

Detailed display of candidates

The smart edit function is used to display details of candidates for functions and variables.

(1)　　　　(2)　　　(3)

```
(function) int func2()
int func2(int a, int b)
```

(4)　(5)　(6)　(7)
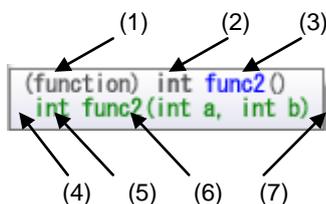
The following items are described.
- [How to open the list of candidates]
- [Descriptions of the individual areas]

[How to open]

A candidate for a method (function) or variable is selected from the display of candidates for functions and variables.

[Descriptions of each area]

No.	Description	
(1)	Kind	Shows whether the selected item is a method (function) or a variable.
		(function): method (function)
		(variable): variable
(2)	Type	Shows the type of the function or variable.
(3)	Name	Shows the name of the function or variable.
(4)	Attribute	Shows the attribute (public, protected, or private).
		If an attribute is not defined, this item is not shown.
(5)	Type	Shows the type of the function or a variable.
(6)	Name and parameter	Shows the name of the function or a variable. When the name is a function, a parameter is also shown.
(7)	Overload information	Shows the amount of information that has been overloaded.
		(Example): (+1 overloads)

Display of candidates for parameters

The smart edit function is used to display details of candidates for parameters.

(1)
(2)

```
function int func2(int a, int b)
 int func2(int a, int b).
```

The following items are described.
- [How to open the list of candidates]
- [Descriptions of the individual areas]

[How to open]

(1) In a case where candidates are automatically to be shown
・ If there is a relevant method (function) on the left side of '(' when '(' is input, candidate parameter names are automatically shown.
(2) In a case where candidates are to be shown in response to keyboard operations
・ When Ctrl + Shift + Space is pressed on the keyboard while the text cursor is at the location of a parameter for a method (function), a list of candidate parameter names will be shown.

[Descriptions of each area]

(1)
(2)

```
function int func2(int a,  int b)
 int func2(int a, int b).
```

No.	Description	
(1)	Candidate	Shows the name of the candidate and the parameter.
(2)	Attribute of candidate	Shows the attribute of a candidate.

When a candidate for a parameter is shown, a parameter for the location of the text cursor is highlighted (in the figure above, the first parameter is highlighted).

Display of tooltip information

The smart edit function is used to display information on functions and variables.

The following items are described.
- [How to open the list of candidates]
- [Descriptions of the individual areas]

[How to open]

Place the mouse cursor on a function or variable.

[Descriptions of each area]

The following icons are displayed according to the type of the target on which the mouse cursor is placed.

Icon	Description
≡	Indicates that the target is a typedef.
≡◆	Indicates that the target is a function.
◆	Indicates that the target is a variable.
◆	Indicates that the target is a structure or union.
{ }	Indicates that the target is a namespace.
≡	Indicates that the target is of enumeration type.

When the mouse cursor is placed on a variable of class, structure, union, or enumeration type, its members are displayed as tooltip information.
- 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.
- Up to 20 members can be displayed, and each is placed on a single line. If the number of members exceeds 20, the 21st and subsequent members are indicated by "…".
- Members are displayed in the same order as they are defined in the source file.

7.1.8 Additional description of mixed display

[Before addition]
None
[After addition]
The mixed display feature displays the source program and assembly code together; in normal display mode, only the source program is displayed.
You can set mixed display from the toolbar.
When the same file is opened again after an Editor panel that has been opened in the mixed display mode is closed, it will be displayed in the normal display mode rather than the mixed display mode. If the debugging tool is disconnected and reconnected, the file will also be displayed in the normal display mode.

Note the following when using mixed display mode.
- Editing is not available.

- Features that change a files content, such as cut and paste, delete, redo, replace, outline, and indent, are not available.
- The select all feature is not available.
- The display on the panel cannot be split.

You can save mixed code into a text file or CSV file in either of the following ways.
- Activate an Editor panel in mixed display mode and select [File] – [Save Source Mixed Data As...] from the CubeSuite+ menu.
- Select [Save Source Mixed Data As...] from the context menu on an Editor panel in mixed display mode

7.1.9   Additional description of recycle mode

[Addition]

The recycle mode is used to display the sequence of multiple source files in the editor panel during debugging while the program counter (PC) enters code generated by multiple source files during stepped execution, with the multiple files of source code displayed in the same editor panel.
The recycle mode is set from [Tools] – [Options] – [Text Editor].

If an editor panel for the corresponding source file has been already opened, it will not be displayed in the panel running in recycle mode; instead, it is only displayed in the editor panel where it was already open.
Editing in the editor panel is still possible in recycle mode. When the program counter (PC) is in a given file so that file is open in the editor panel in recycle mode while a program is being executed and the contents of the panel are edited, that panel leaves recycle mode and a new editor panel is opened in recycle mode.
When an editor panel operating in the recycle mode is closed, the next editor panel to be opened will initially operate in recycle mode.

7.1.10   Change to description of split bar

[Before addition]
You can split the Editor panel display by using horizontal and vertical split bars within the view. This panel can be split up to **four times**.
[After addition]
You can split the Editor panel display by using horizontal and vertical split bars within the view. This panel can be split up to **two times vertically, and two types horizontally**.

7.1.11   Additional description of jump to functions

[Addition]

The Jump to Functions feature depends on the build tool in use and is only available when the following conditions are satisfied.

(a) When the build tool is CA78K0R, CA850, or CA78K0
   · The target is a function in the active project.
       · The type of project specified as the active project is "application".
   · A file with the symbol information is specified in [Download files].
    However, if the file is disconnected from the debug tool, it is specified as the first file in the [Download files] property.
        Note: When a file that includes symbol information is disconnected from the debug tool, jumping to static functions is not possible.

(b) When the build tool is CC-RH, CC-RX or CX
   · When the debug tool is disconnected
    - The type of project specified as the active project is "Application".
    - A file with the symbol information is specified as the first file in the [Download files] property.
    - The above file includes information on the target function.
    - The target function is a global function.
   · When the debug tool is connected to download the load module
    - 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.
· Furthermore, in the case of jumping to functions in C++ programs, the following point for caution also applies to specifying the functions.
- When a function a given character string is specified as the chosen function name, there is a possibility that the jump will actually be to a different function with the same name.
   (1) Member functions of classes
      It's necessary to include the name of the class of which the target function is a member.
      When other functions have the same name as the target function but the arguments are different, please also include the argument type.
      Example: "memfunc": Jumping is not possible.
              "Class::memfunc(short)": Jumping is possible.
   (2) Functions defined in Namespaces
      It's necessary to include the full name of the Namespace to which the target function target belongs.
      When other functions have the same name as the target function but the arguments are different, please also include the argument type.
      Example: "func": Jumping is not possible.
              "Namespace1::Namespace2::func(int)": Jumping is possible.
   (3) Template functions
      It's necessary to include the type of the arguments in the function generated by the compiler.
      Example: "template": Jumping is not possible.
              "template(int, short)": Jumping is possible.
(c) External build tools
   · The target is a function in the active project.
   · A file with the symbol information is specified in [Download files].
   However, if the file is disconnected from the debug tool, it is specified as the first file in the [Download files] property.
      Note: When a file that includes symbol information is disconnected from the debug tool, jumping to static functions is not possible.

7.1.12   Additional description of the context menu
[Addition]

ProgramAnalyzerAddVariableToChart	Registers a variable in the [Variable Value Changing Chart] tab of the Analysis Chart.

7.1.13   Additional description of the Print Preview dialog box
[Before addition]
   (1) Preview area
      The image to be printed is displayed as a preview.
[After addition]
   (1) Preview area
      The image to be printed is displayed as a preview. 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 mode or mixed mode.
      Even collapsed lines are displayed in an expanded state in a print preview.
   (a) When the debug tool is not connected
      The display is as follows.
         Left of page: line number
            However, this is not displayed when line numbers are hidden on the editor panel.
         Page header: file name (fully qualified path)
         Page footer: page number
   (b) When the debug tool is connected (in normal display mode)
      The display is as follows.

Left of page: line number and address

However, these are not displayed if line numbers and addresses are hidden on the
editor panel.

Page header: file name (fully qualified path)

Page footer: page number

(c) When the debug tool is connected (in mixed display mode)

The display is as follows.

Left of page: line number, address, instruction code

However, any of these items that are hidden on the editor panel are not displayed.

Page header: file name (fully qualified path)

Page footer: page number

7.2   Modifications in Build

This section describes modifications in CubeSuite+ V1.01.00 Build User's Manual
(document # R20UT0783EJ0100, R20UT2143EJ0100).

7.2.1   Modified description of Link Options tab (only in R20UT2143EJ0100)

[Location]   Page 210

[Before change]   (5) [Device]
                      Use on-chip debug

[After change]   (5) [Device]
                      Set enable/disable on-chip debug by link option

7.2.2   Additional description of the stack usage tracer

[Location]   Page 315 → Functions analyzed

[After addition]   Consequently, functions in assembly files written by the user and library files
created by the user are not analyzed. For this reason, the information for
these files must be set using the Adjust Stack Size dialog box.

**And, since an interruption function also becomes the outside of analysis
management, it is necessary to set up applicable information using a
adjust stack size dialog box.**

[Location]   Page 324 → Functions analyzed

[After addition]   Consequently, functions in assembly files written by the user and library files
created by the user are not analyzed. For this reason, the information for
these files must be set using the Adjust Stack Size dialog box.

**And, since an interruption function also becomes the outside of analysis
management, it is necessary to set up applicable information using a
adjust stack size dialog box.**

7.2.3   Additional description of the Link Directive File Generation dialog box

[Addition]

Link Directive File Generation dialog box

This dialog box is used to generate a link directive file based on the specified memory and
segment allocation information.
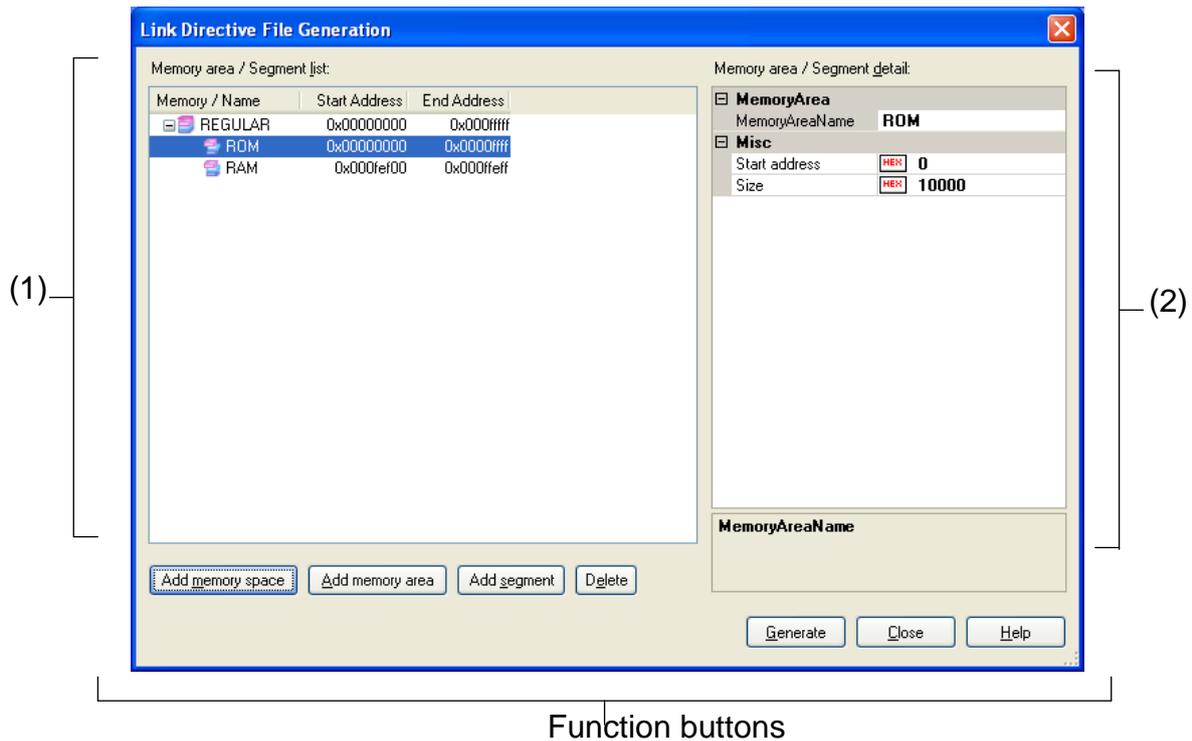
Function buttons

The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]
- On the Project Tree panel, select the Build tool node, and then select [Create Link
  Directive File...] from the context menu.
- On the Property panel of the build tool, select the Link Options tab, and then click on the
  [...] button for the [Input File] ― [Generate link directive file] property.

[Description of each area]
(1) [Memory area/Segment list:] area
   This area displays a list of the memory spaces, memory areas, and segments to be
   generated in the link directive file.

   (a) [Memory/Name]
    This column displays the names of memory spaces, memory areas, and segments.
    For a memory space, the corresponding name from among the following memory
    space names is displayed.
    - REGULAR
    - EX1, EX2, EX3, EX4, EX5, EX6, EX7, EX8, EX9, EX10
    - EX11, EX12, EX13, EX14, EX15
    For memory areas and segments, the contents of this column can be directly edited. If
    a memory area name or a segment name is changed, the [Name] in the [Memory
    area/Segment detail:] area is also changed.

   (b) [Start Address]
    This column displays the start addresses of memory spaces, memory areas, and
    segments.

For memory areas and segments, the contents of this column can be directly edited. If a start address is changed, the [Start Address] in the [Memory area/Segment detail:] area is also changed.

(c) [End Address]
This column displays the end addresses of memory spaces and memory areas.
In a segment row, "—" is shown in this column.

(d) Buttons

Add memory space	Adds a new memory space below the last row of the list. The new memory space is named EX*n* with the smallest value that can be registered when the space is created (n: a decimal value from 0 to 15). Note that this button is disabled when memory spaces EX1 to EX15 (15 spaces in total) have been registered.
Add memory area	Adds a new memory area below the row selected in the list. The default name for the new memory area is "NewMemoryArea_XXX" (XXX: a decimal value from 0 to 255). Make detailed settings of the memory area in the [Memory area/Segment detail:] area. Note that this button is disabled when a segment row is selected or when 256 memory areas have been registered in the list.
Add segment	Adds a new segment below the row selected in the list. The default name for the new segment is "Seg_XXX" (XXX: a decimal value from 0 to 255). Make detailed settings of the segment in the [Memory area/Segment detail:] area. Note that this button is disabled when a memory space row is selected or when 256 segments have been registered in the list.
Delete	Deletes the memory space (except for REGULAR), memory area, or segment selected in the list. When a memory area is deleted, the segments included in the memory area are also deleted. Memory space REGULAR cannot be deleted.

This area has the following functions.

- Expand or collapse the display of a row
  Double-click a row or click the + or − mark at the beginning of a row to expand or collapse the display of the row.

- Move a memory area or segment row
  You can move a memory area row or a segment row by dragging and dropping it.
Remark If a memory area row is moved, the segments included in the memory area are also moved.

- Copy a memory area or a segment
  Select a memory area or a segment and press the [Ctrl] + [C] keys to copy it.
  To paste it, press the [Ctrl] + [V] keys.
  The memory area or segment is pasted under the row selected when the [Ctrl] + [V] keys are pressed.
  "C_" is added at the beginning of the name of the pasted memory area or segment, and if the number of characters in the name exceeds the maximum allowed number, the end characters of the name are truncated.

Remarks 1. When a memory area is copied, the segments in the memory
area are not copied.

2. The start address column for the pasted memory area or segment
is blank.

3. If a memory area or a segment cannot be pasted due to an
attribute of the destination memory area, an error will occur.

(2) [Memory area/Segment detail:] area

This area displays detailed information regarding the memory area or segment
selected in the [Memory area/Segment list:] area. You can also edit the information in
this area.

(a) Detailed information on memory area

MemoryAreaName	Specify the memory area name. Specifiable characters are numeric characters (0 to 9), uppercase alphabet letters (A to Z), lowercase alphabet letters (a to z), underscore (_), question mark (?), and at sign (@).	
	Default	NewMemoryArea_*XXX* (*XXX*: a decimal value from 0 to 255)
	How to change	Directly enter a value in the text box, or click on the [...] button and edit the contents in the opened Character String Input dialog box.
	Restriction	Up to 31 characters

Start address	Specify the start address of the memory area. This field can be left blank or only a hexadecimal value can be specified.	
	Default	Blank
	How to change	Directly enter a value in the text box.
	Restriction	0x0 to 0xFFFF (hexadecimal) [78K0 (except for products with a memory bank)] 0x0 to 0xFFFFF (hexadecimal) [RL78, 78K0R, and 78K0 (products with a memory bank)] The specifiable values depend on the device.

Size	Specify the size of the memory area. This field can be left blank or only a hexadecimal value can be specified. Note that an error will occur at linkage if this field is blank.	
	Default	Blank
	How to change	Directly enter a value in the text box.
	Restriction	0x1 to 0xFFFF (hexadecimal) [78K0 (except for products with a memory bank)] 0x1 to 0xFFFFF (hexadecimal) [RL78, 78K0R, and 78K0 (products with a memory bank)] The specifiable values depend on the device.

(b) Detailed information on segment

Name	Specify the segment name. Specifiable characters are numeric characters (0 to 9), uppercase alphabet letters (A to Z), lowercase alphabet letters (a to z), underscore (_), question mark (?), and at sign (@).	
	Default	NewMemoryArea_*XXX* (*XXX*: a decimal value from 0 to 255)
	How to change	Directly enter a value in the text box, or click on the [...] button and edit the contents in the opened Character String Input dialog box.
	Restriction	Up to eight characters

Specify the start address	Select whether to specify the segment start address.		
	Default	No	
	How to change	Select from the drop-down list.	
	Restriction	Yes	Specifies the segment start address.
		No	Does not specify the segment start address.

Start address	Specify the start address of the segment. This field can be left blank or only a hexadecimal value can be specified. This property is displayed only when [Yes] is selected in the [Specify the start address] property.
	Default	Blank
	How to change	Directly enter a value in the text box.
	Restriction	0x0 to 0xFFFF (hexadecimal) [78K0 (except for products with a memory bank)] 0x0 to 0xFFFFF (hexadecimal) [RL78, 78K0R, and 78K0 (products with a memory bank)] The specifiable values depend on the device.

Merge attribute	Specify the merge attribute of the segment. This property specifies how to merge multiple segments with the same name.		
	Default	Auto (None)	
	How to change	Select from the drop-down list.	
	Restriction	Auto(None)	Executes the same processing as sequential merge (SEQUENT). However, if relocation attribute AT is specified at assembly for any of the input segments having the same name, an error (COMPLETE) processing will be done.
		Merge sequentially (SEQUENT)	Merges multiple segments having the same name sequentially in the order of appearance so that there is no vacant space between them. BSEG-attribute segments are merged in bit units in the order of appearance.
		Error (COMPLETE)	When multiple segments have the same name, an error will occur.

[Function buttons]

Button	Function
Generate	Generates a link directive file (named project-name.dir) based on the specified memory and segments allocation information, and then adds to the project. The link directive file is generated in the project folder. The link directive file that has been generated is also shown on the project tree, under the File node. The generated link directive file will be a build target. If a link directive file has already been registered to the project, then the file will be removed from the build target.
Close	Closes the dialog box.
Help	Displays the help of this dialog box. This is not supported by this version.

7.3   Modification in 78K0 debug
This section describes modification in CubeSuite+ V1.01.00 78K0 Debug User's Manual (document# R20UT0731EJ0100).

7.3.1   Addition of point trace description Special Function Registers (SFRs) /variables of 2 byte size.
[Location] page 118 Addition before 「2.11.5 Display the collected execution history」
[Addition]
2.11.4(2) When an access to SFRs/variables of 2byte size occurs.   [IECUBE]
Register each address of the higher 8 bit and the lower 8 bit in watch panel directly and make a trace event to do point trace of SFRs/variables of 2 byte size. (See the figure below)

[Setting]
Example: The address higher 8 bit is 0xff08 and lower 8bit is 0xff09, right click and select [Trace output] and [Record Value] to make a trace event.

Watch1

Watch	Value	Type(Byte Size)	Address	Memo
0xff08	0 ?(1)		0xff08	
0xff09	0 ?(1)		0xff09	

Local Variables CPU Register Trace Events Watch1

A
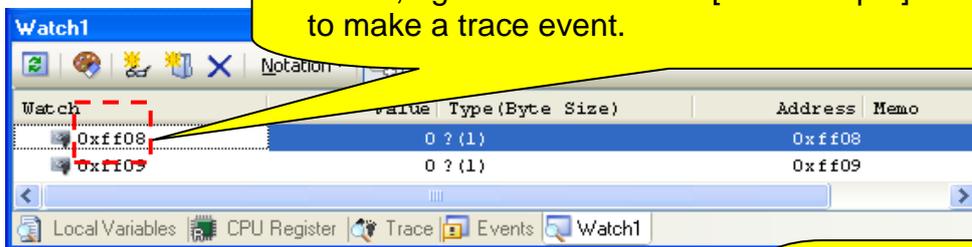
trace result of this setting will be as follows.

[Trace result]
An access address of trace shows a defined SFR name or address.

Trace

Number	Time (h:min:s,ms,µs,ns)	Line Number/Address	Source/Disassemble	Address	Data
0			<<<ADCR=0x0000>>>		
0				ADCRL	R 0x00
1				0xff09	R 0

Stopped by Software Break.

Local Variables CPU Register Trace Even
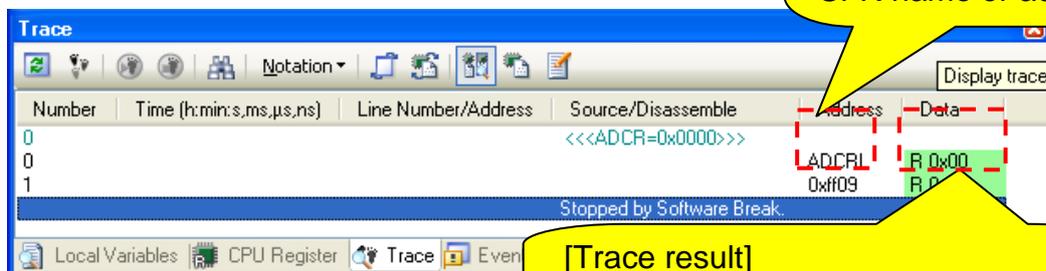
[Trace result]
The data is displayed each 1 byte.

All trademarks and registered trademarks are the property of their respective owners.